Heuristic Ensembles of Filters for Accurate and Reliable Feature Selection

Ghadah Nasser Aldehim

Submitted for the degree of Doctor of Philosophy

School of Computing Sciences
University of East Anglia
December 2015

Abstract

Feature selection has become increasingly important in data mining in recent years. However, the accuracy and stability of feature selection methods vary considerably when used individually, and yet no rule exists to indicate which one should be used for a particular dataset. Thus, an ensemble method that combines the outputs of several individual feature selection methods appears to be a promising approach to address the issue and hence is investigated in this research.

This research aims to develop an effective ensemble that can improve the accuracy and stability of the feature selection. We proposed a novel heuristic ensemble of filters (HEF). It combines two types of filters: subset filters and ranking filters with a heuristic consensus algorithm in order to utilise the strength of each type. The ensemble is tested on ten benchmark datasets and its performance is evaluated by two stability measures and three classifiers. The experimental results demonstrate that HEF improves the stability and accuracy of the selected features and in most cases outperforms the other ensemble algorithms, individual filters and the full feature set.

The research on the HEF algorithm is extended in several dimensions; including more filter members, three novel schemes of mean rank aggregation with partial lists, and three novel schemes for a weighted heuristic ensemble of filters. However, the experimental results demonstrate that adding weight to filters in HEF does not achieve the expected improvement in accuracy, but increases time and space complexity, and clearly decreases stability. Therefore, the core ensemble algorithm (HEF) is demonstrated to be not just simpler but also more reliable and consistent than the later more complicated and weighted ensembles.

In addition, we investigated how to use data in feature selection, using ALL or PART of it. Systematic experiments with thirty five synthetic and benchmark real-world datasets were carried out.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

D	Dataset
F	Filter
RF	Ranking filter
SF	Subset filter
FS	Feature selection
EFS	Ensemble of feature selection
CV	Cross-validation
PART	Feature selection with CV
ALL	Feature selection without CV
iid	Independently and identically distributed
HEF	Heuristic Ensemble of Filters
WHEF	Weighted Heuristic Ensemble of Filters
HHEF	Hybrid Heuristic Ensemble of Filters
x	Feature
y	Class variable
x_R	Relevant features
x_I	Irrelevant features
$x_{\acute{R}}$	Redundant features
x_{Id}	Independent feature
N	Total number of features
N_R	Number of relevant features
N_I	Number of irrelevant features
$N_{\acute{R}}$	Number of redundant features
S	Number of instances
n	Number of runs
k	Number of folds
K	Number of top features selected
A	Number of Algorithms
l	Number of filter members in HEF
l_s	Number of subset filter members in HEF

l_r	Number of rank filter members in HEF
Q	Heuristic consensus rule
N_P	Total number of predictions (number of test samples)
P	Probability
X	Set of all features of size N
X_i, X_j, X_m	$X_i \in X$, $X_j \in X$, $X_m \in X$
N_{X_i}	Number of features in X_i
f_i	Feature ranking with respect to X
X_s	Set of k feature subsets obtained from k folds
N_s	Total number of occurrences of any features in X_s
q_x	Frequency of feature x in X_s
G	Number of features in the blocks
Δx	Number of features (x) in ALL - number of features (x) in PART
Δacc	Accuracy in ALL - accuracy in PART
ɣ, λ, β	Coefficients
Θ	Decision threshold
M	Mean
A	0.05 (significant)
Σ	Standard deviation
E	Amount of noise injected
P	The number of sample injected by noise
W	Weight
r_i	Ranking position of x_i in ranking j (R_j),
m_j	subset of ranking j
$\overline{r_p}$	Mean rank aggregation of partial ranking

Publications

As a result of the research undertaken for this thesis, the following publications have been produced.

Journal Papers

1. ALDEHIM, G. & WANG, W. 2015. Determining Appropriate Approaches for Using Data in Feature Selection. *International Journal of Machine Learning and Cybernetics*, Springer, DOI: 10.1007/s13042-015-0469-8, pp: 1-14

Conference Papers

1. ALDEHIM, G., DE LA IGLESIA, B. & WANG, W. 2014. Heuristic Ensemble of Filters for Reliable Feature Selection. *International Conference on Pattern Recognition Applications and Methods (ICPRAM 2014),* France.
2. ALDEHIM, G. & WANG, W. 2014. Reliability and Effectiveness of Cross-Validation in Feature Selection. *Research and Development in Intelligent Systems XXXI.* Springer, pp: 179-184.
3. ALDEHIM, G. & WANG, W. 2015. Hybrid Ensemble for Identifying the Most Relevant Features. *8th Saudi Internatioal Conference.* London
4. ALDEHIM, G. & WENJIA, W. 2015. Weighted Heuristic Ensemble of Filters. *Internatioal Conference on Intelligent Systems (IntelliSys 2015).* London: IEEE, pp: 609-615.

Acknowledgements

I feel immense pleasure in taking the opportunity to thank those who helped me to complete this thesis work.

I would like to thank the Almighty Lord for giving me the strength and ability to complete this thesis and for placing such amazing people in my path. Also, I would like to express my sincere gratitude to my supervisor, Dr Wenjia Wang, for his continuous support during my PhD journey, for his patience, motivation, enthusiasm and immense knowledge. His guidance helped me throughout my time of researching and writing this thesis. I could not have imagined having a better advisor and mentor for my PhD study. I would also like to thank my second supervisor, Dr Beatriz de la Iglesia, for her encouragement and insightful comments. Also, I thank the reviewers of my papers for their critical comments, which added value to our work.

Beside those, my personal gratitude goes to my parents for the unconditional love and support that they have given me. I am very thankful to my husband, Bassam, for his support and endless encouragement and for doing everything possible to help every step of the way, as though he himself was doing the PhD research, as well as me. Also, special thanks to my children, Mohammad and Yara, who suffered so much during my study by not having the time they needed from me, along with my wonderful brothers and sister who encouraged me with their prayers, moral support and never-ending love. Without my family, this research would not have survived.

My acknowledgements go to my sponsors, Princess Nora bint Abdul Rahman University and the Ministry of Education in Saudi Arabia, represented in the Saudi Cultural Bureau and the Royal Embassy in the UK, for the full scholarship which was given to me.

And to so many others who helped along the way, thank you.

Chapter 1

Introduction

1.1 Background

With the rapid advances in computer and database technologies, datasets with thousands of features are now ubiquitous; however, most of the features in enormous datasets may be irrelevant or redundant, which can cause poor efficiency and over-fitting in the learning algorithms. Therefore, it is necessary to employ some feature selection methods to remove irrelevant and redundant features from data. This allows domain experts to shift their focus onto the most resilient and discriminating features, while also reducing model complexity (Saeys et al., 2007).

Methods for feature selection are roughly divided into two main categories: filters and wrappers. A filter method relies on the general characteristics, such as relevance or correlation, of the training data in order to select certain features without involving any learning algorithm. Generally, filters fall into two sub-categories: rank and subset. Rank filters (RF) usually calculate a feature relevance score and then rank the features according to their relevance score – the higher the score, the more relevant it is (Huang et al., 2007). On the other hand, subset filters (SF) only select a subset of the features that are considered to be relevant as their output. In general, filter methods are independent of classifiers, computationally simple and fast and thus have been widely used for many different feature selection tasks, particularly with very high-dimensional datasets such as genomic data.

The wrapper approach, on the other hand, depends on a learning algorithm to evaluate each subset of features. This approach may choose features that give a high prediction performance, but it has certain disadvantages. The most noticeable one is that a wrapper is highly model-dependent, that is, the "best" subset selected through a particular type of classifier may not be the "best" for other types of classifiers. It is also very computationally intensive, in particular when building a classifier that has a high computational cost (Saeys et al., 2007). Therefore, for large datasets with high dimensionality, the wrapper approach is too time-consuming to be used, and hence, the filter approach is preferable in practice.

There are, however, many different types of filters, and their performance in terms of accuracy and stability may vary considerably from one dataset to another. It is not clear when a particular filter should be used for a given dataset. Additionally, the various feature selection algorithms available may select feature subsets, which are often

different collections of local optima identified within the space of the feature subsets (Saeys et al., 2008) and their performance is unreliable and unpredictable. Thus, an ensemble of feature selection (EFS) method that combines the outputs of several individual methods may find a subset that is more stable and accurate than an individual feature selection method. Two key steps are necessary to develop a feature selection ensemble. The first entails identifying a set of particular feature selectors, each of which delivers an output, and the second entails aggregating the outputs of all the selectors in order to generate a final selection result, either a subset or a ranking of selected features. It is important that the feature selectors are diverse enough from each other to avoid falling into the same local optima, and this may be achieved by combining the outputs of different types of feature subsets (FS) or by combining the output from the same type of FS by using perturbations of data (Dietterich, 2000, Hoeting et al., 1999). Aggregating the outputs of the different feature selection procedures can be achieved by ranking the scores in order to generate a consensus feature ranking, or by simply counting the most frequently selected features in order to generate a consensus feature subset (Saeys et al., 2008).

1.2 Motivation

Several studies in recent years have focused on improving feature selection techniques. However, the problem with using a single FS is that each FS has a different nature and will have its own biases. Therefore, the different feature selection techniques available may select feature subsets which are different in quality and quantity; for example, even though they obtain high accuracy, they may neglect stability (Fahad et al., 2014). Such an instability issue dampens the confidence of researchers in relying on any of the various subsets of selected features. In addition to that, various feature selection algorithms available may select feature subsets that are often different collections of local optima identified within the space of the feature subsets (Saeys et al., 2008) and their performance is unreliable and unpredictable. Thus, an EFS method that combines the outputs of several individual methods may enhance the results by finding a subset that is a closer approximation of the relevant subset, or it may at least provide a more stable and accurate subset. However, there is still an amount of open research questions which needs to be taken into consideration and investigated in order to improve the accuracy and stability of the ensemble feature selection method.

In this thesis, we focus on some of these open research questions, such as determining the FS members of an ensemble, particularly among the high number of feature selection methods in the literature. The majority of the previous studies on feature selection ensemble use ranking filters only, which motivated this research to combine SF with RF in our ensemble algorithm to exploit the advantages of each and also to investigate the aggregation methods to produce a more accurate final model. The majority of previous studies on feature selection ensemble use an aggregation method with full rank lists as they use rank filters, which motivated this research to investigate different methods of dealing with a partial list. In addition to that, we investigated the benefit of adding more weight to FS members. Previous studies on feature selection ensemble treat all FS members equally, so this motivated this research to investigate different weights for calculating the total scores of the selected features, which may improve performance. In summary, the thesis will therefore set out to investigate different methods in order to improve the accuracy and stability of the EFS algorithm.

1.3 Research Aim and Objectives

This research aims to develop an effective ensemble that can improve the accuracy and stability of feature selection. In order to achieve this aim, the following objectives need to be completed.

1- To review the feature selection methods in the literature and identify the most appropriate ones for selecting a subset of original features.

2- To determine the numbers and types of filters to be members of our ensemble.

3- To investigate combination methods in order to produce a more accurate final model.

4- To determine the appropriate approaches for using data in feature selection.

5- To investigate the effect of assigning variable weights to FS members.

6- To evaluate the performance of EFS.

1.4 Research Questions

Our main thesis question is:

How can we develop an effective ensemble of feature selection that can improve the accuracy and stability of feature selection?

This research considers the following sub-questions to answer the main question:

1- What members should be used in an EFS and how many should be used?

2- What consensus methods should be used?

3- Should all the members be treated equally?

4- How do we evaluate the performance and the stability of the EFS?

5- What are the appropriate approaches for using data in feature selection?

1.5 Contributions

This thesis describes in detail the work done and the results achieved in my PhD research (see section 9.1). The contributions made to knowledge can be summarised as follows:

1- Developed a novel ensemble feature selection algorithm to improve the stability and accuracy of the feature selection. The experimental results on some benchmark datasets show that the proposed algorithm outperformed the other ensemble algorithms, individual filters and complete feature set, in most cases.

2- In feature selection, there exists a long ongoing issue, which is how to use data when doing feature selection, either use all or part of the available data, this research identified the appropriate approaches and guidelines for using data in feature selection based on the characteristics of a dataset.

3- Designed three novel methods for weighting the filter members. The proposed methods, however, have not achieved much of the expected improvement in accuracy, but have increased time and space complexity, and instability. Therefore, the

contribution made to knowledge is that, in practice, naively weighting the filter members according to their performance does not lead to better results.

4- Generated 21 synthetic datasets which cover different problems such as increasing the number of irrelevant features and decreasing the number of instances or varying the level of noise in the output. The synthetic datasets may therefore provide some benchmarks for other researchers to use.

1.6 Structure of Thesis

The remainder of this thesis is structured as follows.

Chapter 2: ***Literature Review on Ensemble Feature Selection*** – This presents the basic steps in feature selection, and it discusses related works on filters, wrappers and hybrids. In addition, it assesses the advantages and disadvantages of each of these methods. It also explains the methods used for constructing ensembles and provides details relating to EFS: concepts, components and the research studies in this field.

Chapter 3: ***Methodology*** – This chapter explains the experimental design to be used in our thesis. It begins by illustrating a general framework for the ensemble of feature selection and the main tasks that need to be considered to answer the research questions. It also describes the evaluation methods to be used. Then, it describes the data and the experiment design used in the research.

Chapter 4: ***Heuristic Ensemble of Filters*** – This chapter explains the framework of Heuristic Ensemble of Filters (HEF) and then provides explanations for the filters chosen in the HEF and the heuristic rules. The experiments are applied to 10 benchmark datasets and compared with the results from each filter member. The experiment results are analysed and discussed.

Chapter 5: ***Determining Appropriate Approaches for Using Data in Feature Selection*** – This chapter investigates the way of using data in FS, using all the data (ALL) or using just the training dataset in FS (PART). It starts by describing each approach and then shows the lack and weaknesses of existing studies on this topic. The experiments compare these two approaches with respect to 10 benchmark datasets and

21 synthetic datasets generated in terms of number of features, stability and accuracy. The experiment results are analysed and discussed.

Chapter 6: ***Improving Heuristic Ensemble of Filters*** – This chapter attempts to improve the HEF through three procedures. Firstly, it adds different wrappers after the HEF in order to identify the most important features while preserving the same accuracy and stability. Secondly, it adds more filters as members in the HEF. Thirdly, it discusses changing the aggregation method from counting the frequency of each feature selected to mean rank aggregation by sorting the selected features based on the means of their ranks in all the ranking filters. In addition, it discusses the partial rank and ways to deal with this situation.

Chapter 7: ***Weighted Heuristic Ensemble of Filters*** – This chapter investigates the effect of varying the weight for each filter in an ensemble. It then describes the frameworks of adding fixed weight, variable weight and selective filters. To the best of my knowledge, this is the first study thus far that gives weight to some filter members based on a validation set or by using prior knowledge. This chapter then provides the results and evaluates the three proposed approaches to conclude the findings.

Chapter 8: ***Evaluation and Discussion*** – This chapter gives an overview evaluation and discussion of the main findings of this research.

Chapter 9: ***Conclusion*** – This chapter concludes our work by summarising the concepts developed and the achievements made, and it makes some suggestions for how the work could be extended in the future.

Chapter 2

Literature Review on Feature Selection Ensemble

2.1 Introduction

Through the rapid advancement of information technology, it has become more and more economical to collect, store, re-possess and retrieve a large amount of data from the database. However, the majority of datasets may have irrelevant and redundant features which can lead to inefficient analysis. Thus, there is a need to remove irrelevant and redundant features from datasets (Blum and Langley, 1997). Also, analysis suffers from dimensionality – data dimensionality affects both the training and runtime phases of a classifier. Feature selection is one of the essential and frequently used techniques in data pre-processing to preserve useful features by removing irrelevant and redundant features and to solve the dimensionality problem, improve classification performance and speed up the data mining algorithm (Guyon and Elisseeff, 2003, Liu and Yu, 2005, Martín-Smith et al., 2015).

In summary, this chapter will start by giving an introduction to feature selection in Section 2.2. Then, Section 2.3 will describe how feature selection is performed using filters and it will also provide examples of some of the most commonly used filter methods. The main advantages and disadvantages of filters will also be outlined. Following this, a detailed explanation of wrapper feature selection methods will be presented in Section 2.4. The section will provide examples of some of the most frequently used wrapper methods. The main advantage and disadvantage of wrappers will also be detailed. This will be followed by a detailed explanation of hybrid feature selection methods established in Section 2.5. Various examples of some of the most commonly used hybrid methods will also be made available. Section 2.6 will give an introduction to the ensemble approach and describe the main methods for constructing the ensemble. Then, in Section 2.7, the feature selection ensemble will be introduced by explaining the ensemble ideas for feature selection and their combination methods. Finally, a number of studies on feature selection ensembles will be presented in Section 2.8.

2.2 Feature Selection

Feature selection is also known as variable selection, attribute selection or variable subset selection. This is a technique that can be deployed to select a subset of relevant features for building improved learning models (Guyon and Elisseeff, 2003). It can be described as a technique that finds a good subset of the original input features under some objective measure, such as prediction accuracy, structure size, or minimal use of input features. It is important in both supervised and unsupervised data mining; this research deals with supervised learning, and in particular with classification tasks.

Feature Selection (FS) has been a fertile field of study and its development has been going on since the 1970s in pattern recognition, machine learning and data mining (Liu and Yu, 2005). The process focuses simply on the relevant features in the dataset by removing any irrelevant, redundant or noisy data for the purpose of bringing immediate effect to the application. Some of the most advantageous aspects of this process are mentioned in detail below:

1- To enhance model performance and avoid over-fitting. This can be seen as an example of prediction performance in the case of supervised classification. Also feature selection has a vital role in building better cluster detection in the case of clustering.

2- To provide faster and more cost-effective models.

3- To gain a deeper insight into the underlying processes that generated the data (Saeys et al., 2007).

4- To reduce the amount of data; therefore, the data will be much easier to handle throughout the process of performing data mining, and it will be possible to recognise and reveal the relevancies within the data (Czekaj et al., 2008).

Furthermore, FS is different to other reduction methods (feature construction or principal component analysis), as it does not change the original representation, so it keeps the original semantics of the features, helping domain experts to obtain better understanding regarding their data. These remarkable and extraordinary benefits have led researchers to consider the idea of using FS in numerous types of tasks throughout their analysis; these include bioinformatics, text categorisation, image retrieval,

customer relationship management, intrusion revealing and genomic analysis (Yu and Liu, 2004).

Typically, feature selection can be formally defined in the following scenario (Jain and Zongker, 1997):

Assuming that X= $\{X_1 , X_2 , ... X_N\}$ is the given set of original features with cardinality N (where N is the number of features in set X), and X_i is the selected feature subset with cardinality N_{X_i} (where N_{X_i} is the number of features in set X_i), then $X_i \subseteq$ X. Also, let J(X) be the selection criterion for selecting feature from set X, and Z is subset of features, Z ⊆ X. We presume that a higher value of J represents a better feature subset. Thus, the goal is to maximise J ( ), so the problem of feature selection is to find a subset of features $X_i \subseteq$ X, this can expressed as:

$$J(X_i) = \max_{Z \subseteq X, |Z| = N_{X_i}} J(Z) \qquad (2.1)$$

Deriving a feature subset that maximises J( ), characteristically the expression consists of four key steps: search strategy, feature subset evaluation, stopping criterion and validation procedure (Liu and Yu, 2005). Further details on each of these four key steps are outlined in the following Sections.

2.2.1 General Procedure of Feature Selection

Most FS methods follow a four step process: subset generation, subset evaluation, stopping criterion, and result validation (Figure 2.1) (Dash and Liu, 1997, Liu and Yu, 2005). Starting with subset generation, the selected search strategy produces the next candidate subset. An evaluation criterion is then applied to evaluate and compare each subset with the others. The best subset is reserved and this process is repeated until a stopping criterion is reached. Finally, the selected subset is passed through a validation procedure to check the validity of the subset. Detailed explanations of each step are provided in the following Sections.

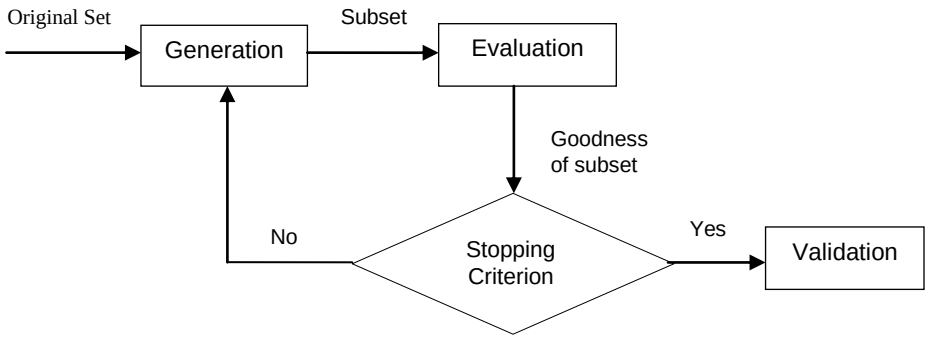


Figure 2. 1: Feature Selection Process (Dash and Liu, 1997)

2.2.1.1 Subset Generation:

Subset generation can mainly be determined by answering two basic questions: where to start and how to search. Firstly, a starting point (or points) has to be selected. Some algorithms start with an empty set or with no features, and then features are added incrementally (forward). Other algorithms start with a full set and then features are deleted incrementally (backward). In the third case, they start with both ends (bi-directional), so that features are iteratively added, removed or produced randomly thereafter. Finally, some algorithms may start with a predetermined number of randomly selected subsets attempts in order to avoid being surrounded by local optima (Liu and Yu, 2005).

Secondly, a searching strategy needs to be specified. Since an original feature set contains N number of features, the total number of competing candidate subsets to be generated is (2^N). This is a huge number even for medium-sized N. So, an exhaustive search is typically not practicable; for this reason, it is rarely used or even considered. Different approaches, such as complete, sequential and random can be implemented for solving this problem.

1) **Complete Search**

This generation procedure performs a full search for the optimal subset according to the evaluation function used after an in-depth search is complete. While an exhaustive search is complete, a strategy does not have to be exhaustive in order to be complete (Schlimmer, 1993). In fact, algorithms which use the complete search such as branch and bound, and beam search, can find the optimal subset much more quickly than an

exhaustive search. But with a high-dimensional dataset, the complete search is still impractical and exponential (Dash and Liu, 1997).

2) **Sequential (Heuristic) Search**

Algorithms with sequential search are simple to implement and fast in producing results as the order of the search space is usually O (N^2), or less. While sequential strategies are faster than complete strategies, the loss of completeness can also mean the loss of optimality, as it is no longer guaranteed that the optimal solution will be found (Dash and Liu, 1997).

Many variations to the greedy hill-climbing approach will be applied through the process. For example, sequential forward selection (SFS), sequential backward selection (SBS) or bi-directional selection (Kabir and Islam, 2010). All these approaches can add or remove features one by one at a time. Another alternative is to add (or remove) X_i features in one step and remove (or add) X_j features in the next step ($X_i > X_j$) (Liu and Yu, 2005). However, the problem of such a strategy is that once a feature is added (or removed) it cannot be added in a later stage. This method is widely known as the nesting effect and if it is intended to be initiated then a problem may occur while using SFS and SBS. In order to overcome this problem, the floating search strategy (Pudil et al., 1994), which can re-select the removed features or delete the already added features, is still effective. The performance of this strategy has been found to be more reliable than other search methodologies. In addition, the floating search strategy is computationally much more efficient than an FS method, or branch and bound (Kabir and Islam, 2010).

3) Random Search

The procedure of random search generally starts with a randomly selected subset. Then, the strategy can follow one of two directions: sequential or stochastic search. Sequential searches such as random-start, hill-climbing and simulated annealing, insert randomness into the above standard sequential approaches (Liu and Yu, 2005), while stochastic searches generate and initiate the next subset in a completely random manner. Examples are the Las Vegas filter (Liu and Setiono, 1996b) and the Las Vegas Wrapper (Liu and Setiono, 1996a).

2.2.1.2 Subset Evaluation:

Subset Evaluation is the technique that is used to measure the efficiency of a candidate subset by some generation procedure; subsequently the value generated is compared with the previous best achieved during the process. If the measure is found to be better, then it will replace the previous best subset. Since a great number of different evaluation techniques exist, it should first be pointed out that although a candidate subset may be found to be optimal or near optimal under a criterion, it may or may not be considered optimal under others. An evaluation criterion can therefore be categorised into two groups based on its dependency on the inductive algorithms that will finally be applied on the selected feature subset (Dash and Liu, 1997). The two groups are: independent criteria (filter) and dependent criteria (wrapper) which will be discussed in Sections 2.3 and 2.4.

2.2.1.3 The Stopping Criterion:

A pre-selected stopping criterion decides when a feature selection process needs to stop. There are a few variations in the stopping criterion used for most feature selection methods such as when the search completes. Also, generation procedures and evaluation functions can influence the choice for a stopping criterion, as follows:

A) Stopping criteria based on a generation procedure can be:

(i) A predefined number of features selected, and/or

(ii) A predefined number of iterations reached.

B) Stopping criteria based on an evaluation function can be:

(i) Addition (or deletion) of any feature that does not produce a better subset;

(ii) An optimal subset according to some obtained evaluation function. Therefore, the loop continues until a pre-set stopping criterion is satisfied.

The feature selection process stops the progress by producing a selected subset of features to a validation procedure (Liu and Yu, 2005).

2.2.1.4 The Validation Procedure:

A simple way of achieving results from validation is to directly measure the result using prior knowledge about the data. If the relevant features are known in advance, as in the case of synthetic data, then a simple way to validate the selected subset is by comparing it to the known optimal subset. Also, knowledge about the irrelevant or redundant features may help in validation as those features are not expected to be selected. In real-world applications, however, such prior knowledge is not available. In this case, the validation task relies on some indirect methods by monitoring the changes of mining performances with the change of features, for example by making a comparison of the classification accuracy rate on the full set of features with the classification accuracy rate on the selected set of features (Liu and Yu, 2005).

As we mentioned earlier, each newly generated subset needs to be evaluated by an evaluation criterion, as feature selection algorithms fall into two broad categories, the filter model and the wrapper model. Recently, research has proposed a hybrid model which combines the advantages of both filter and wrapper to deal with high-dimensional data (Yu and Liu, 2004, Gan et al., 2011)

2.3 Filter:

The filter model relies on the general characteristics of the training data to select some features without involving any learning algorithm. It starts by choosing a search strategy and determining the direction of the search, therefore, to start looking for the relevant features in the dataset. Then, it assigns a relevance score to each feature by statistical or information-based measures; the higher the score is, the more relevant a feature is (Saeys et al., 2007). In some cases, filters rank features according to their relevance. Those which are ranked top are most relevant and those ranked underneath are of least relevance (Huang et al., 2007). In other cases, features with high relevance scores will be selected and low scoring features will be discarded. Finally, the selected features which have high relevance scores are presented as inputs to the classifier (Saeys et al., 2007). The process which describes the way in which filters perform feature selection is shown in Figure 2.2.

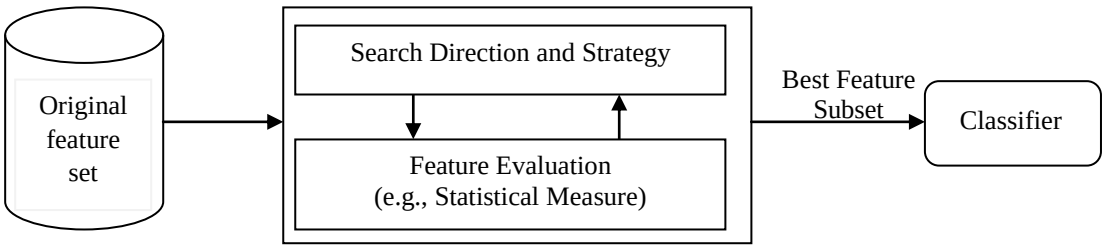


Figure 2.2: Illustration of the filter process

In general, there are many ways to divide filter methods; one of them relies on communicating with the features, namely univariate and multivariate (Zhu et al., 2007). Univariate filter methods consider each feature in the dataset separately when identifying relevant features, such as Information Gain, Mutual Information and Chi-Square, whereas, the multivariate methods consider the interactions among different features in the dataset such as Relief, Focus and Correlation-based Feature Selection (CFS). Other ways to divide the filter methods based on search strategies include complete, sequential and random. In this Section, the filter methods will be divided based on evaluation criteria including distance, information, dependency and consistency.

2.3.1. Distance Measures (Weight)

Distance measures are also known as separability, divergence, or discrimination measures. This method assigns weights to features individually then ranks them based on their relevance to the target concept. A feature is good and thus will be selected if its weight of relevance is greater than a threshold value (Yu and Liu, 2003). To put it more simply, this criterion tries to find features that can separate the class labels of the dataset as much as possible, measured by a metric unit (such as Euclidean distance). For example, in a two-class problem, feature x_1 is preferred over x_2, if x_1 generates a greater difference (distance) between the two classes of conditional probabilities than x_2 (Liu and Yu, 2005). In the literature, a lot of research has used the weight measures as evaluation criteria to generate their filters, such as Branch and Bound (B & B), Best-First Search Strategy (BFF) and Segen's method which was reported in (Dash and Liu, 1997). However, Relief family is a famous and important filter regarding the type of evaluation which is described in the following Sections.

1) Relief:

Relief was proposed by Kira and Rendell (1992). Relief is an easy-to-use, fast and accurate algorithm even with dependent features. It can also deal with discrete and continuous attributes but it is limited to deal only with two-class problems. Relief works by evaluating the worth of an attribute by repeatedly sampling an instance and considering the value of the given attribute for the nearest instance of the same and different class. The process of ranking the features in relief follows three basic steps – calculating the nearest miss and nearest hit, then calculating the weight of a feature, and finally, returning a ranked list of features or the top-K features according to a given threshold. Formally, RELIEF's estimate W $[x_1]$ of single feature x_1 is an approximation of the following difference of probabilities:

W $[x_1]$ = P (different value of x_1| nearest instance from different class) – P (different value of x_1| nearest instance from same class)

The rationale procedure states that a good feature should differentiate instances from different classes and should have the same value from the same class (Kononenko, 1994).

ReliefF (Kononenko, 1994) is an extension of the relief algorithm. It was extended by Kononenko, so that it can deal with multi-class problems, noisy values, missing values and can be used for regression problems. The basic idea of ReliefF is to draw instances at random, compute their nearest neighbours, and adjust a feature weighing vector to give more weight to features that discriminate the instance from neighbours of different classes.

In 2002, Liu et al. enhanced ReliefF by focusing on selective sampling which is referred to as ReliefS. When the training dataset is very large, random sampling is commonly used to overcome the problem. Active feature selection avoids pure random sampling and is realised by selective sampling. The intuitive idea is to select only instances with higher probabilities to be informative in determining feature relevance (Liu et al., 2002b).

However, many other algorithms in this group have similar problems as Relief does. They can only capture the relevance of features to the target concept, but cannot discover redundancy among features. Empirical evidence from feature selection

literature shows that, along with irrelevant features, redundant features also affect the speed and accuracy of learning algorithms and thus should be eliminated as well (Hall, 1999, Kohavi and John, 1997). Therefore, in the context of feature selection for high-dimensional data where many redundant features may exist, pure relevance-based feature weighting algorithms do not meet the need of feature selection very well (Yu and Liu, 2003).

2.3.2. Information Measures

Among non-linear correlation measures, many measures are based on the information-theoretical concept of entropy, a measure of the uncertainty of a random variable. Information measures normally quantify the information that can be gained from each feature. For example, the information gained from features x_1 is determined by the difference between the prior uncertainty and expected posterior uncertainty using x_1. Feature x_1 is preferred to feature x_2 if the information gain from x_1 is greater than that from x_2 (Liu and Yu, 2005).

1) Information Gain:

Information gain (IG) is based on the concept of entropy. In order to calculate information gain, an attribute *x* and a class attribute *y* can be considered. The information gain of a given attribute *x* with respect to class attribute *y* is the reduction in uncertainty about the value of *y* when the value of *x* is known. The value of *y* is measured by its entropy, H(y) (Altidor et al., 2011). The uncertainty about *y*, given the value of *x*, is given by the conditional probability of *y* given *x*, H (y|x).

$$I(y, x) = H(y) - H(y|x) \tag{2.2}$$

where *y* and *x* are discrete variables that take values in $\{y_1.....y_c\}$ and $\{x_1....x_d\}$ then the entropy of *y* is given by:

$$H(y) = -\sum_{i=1}^{c} P(y = y_i)\, log_2\, P(y = y_i) \tag{2.3}$$

The conditional entropy of *y* given *x* is

$$H(y|x) = -\sum_{j=1}^{d} p\left(x = x_j\right) H(y|x = x_j) \tag{2.4}$$

Alternatively the information gain is given by:

$$I(y, x) = H(x)+H(y)- H(x, y) \quad (2.5)$$

where H(x, y) is the joint entropy of *x* and *y*:

$$H(x, y) = -\sum_{i=1}^{c}\sum_{j=1}^{d} P(x = x_j, y = y_i)\ log_2\, p(x = x_j, y = y_i) \quad (2.6)$$

when the predictive attribute *x* is not discrete but continuous, the information gain of *x* with class attribute *y* is computed by considering all possible binary attributes that arise from *x* when we choose a threshold Θ on *x* (Vege, 2012). Θ takes values from all the values of *x*. Then the information gain is simply:

$$I(y, x) = argmax_{x_\Theta} I(y, x_\Theta) \quad (2.7)$$

The major drawback of using information gain is that it tends to choose attributes with large numbers of distinct values over attributes with fewer values even though the latter are more informative (Karegowda et al., 2010).

2) Gain Ratio

The IG measure is biased towards tests with many outcomes, as mentioned above (Karegowda et al., 2010). C4.5, a successor of ID3 (Quinlan, 1986), uses an extension to IG known as gain ratio (GR), which attempts to overcome the bias. Let B be a set consisting of *b* data samples with *c* distinct classes. The expected information needed to classify a given sample can be expressed by:

$$I(B) = -\sum_{i=1}^{c} p_i\, log_2(p_i) \quad (2.8)$$

where p_i is the probability that an arbitrary sample belongs to class y_i. Let attribute *x* have *d* distinct values. Let b_{ij} be the number of samples of class y_i in a subset B_j. B_j contains those samples in B that have value x_j of x. The entropy based on partitioning into subsets by *x* is given by:

$$E(x) = -\sum_{i=1}^{c} I(B)\frac{(b_{1i} + b_{2i} + \ldots + b_{di})}{b} \quad (2.9)$$

The encoding information that would be gained by branching on *x* is:

$$Gain(x) = I(B) - E(x) \quad (2.10)$$

C4.5 applies a kind of normalisation to information gain using a "split information" value defined analogously with Info (D) as:

$$SplitInfo_x(B) = -\sum_{j=1}^{d} \left(\frac{|B_j|}{|B|}\right) \log_2 \left(\frac{|B_j|}{|B|}\right) \quad (2.11)$$

This value represents the information computed by splitting the dataset D, into *b* partitions, corresponding to the *b* outcomes of a test on attribute *x*. For each possible outcome, it considers the number of tuples having that outcome with respect to the total number of tuples in D. The gain ratio is defined as:

$$GainRatio\ (x) = \frac{Gain\ (x)}{SplitInfo\ (x)} \quad (2.12)$$

The attribute with maximum gain ratio is selected as the splitting attribute (Vege, 2012).

3) Symmetrical Uncertainty

Correlation-based feature selection is the base for symmetrical uncertainty (SU) and it evaluates the merit of a feature in a subset using a hypothesis – "Good feature subsets contain features highly correlated with the class, yet uncorrelated to each other" (Ienco et al., 2009). Symmetric uncertainty is used to measure the degree of association between discrete features. It is derived from entropy (Chen et al., 2006). It is a symmetric measure and can be used to measure feature-feature correlation.

$$SU = 2.0 \times \frac{H(x) + H(y) - H(x,y)}{H(y) + H(x)} \quad (2.13)$$

Symmetrical uncertainty is calculated by the above equation. H(x) and H(y) represent the entropy of features *x* and *y*. The value of symmetrical uncertainty ranges between 0 and 1. The value of 1 indicates that one variable (either *x* or *y*) completely predicts the other variable (Ienco et al., 2009). The value of 0 indicates that both variables are completely independent (Vege, 2012).

4) Fast Correlation-Based Filter (FCBF)

Fast Correlation-Based Filter (FCBF) (Yu and Liu, 2004) selects good features for classification based on correlation analysis of features (including the class) by using Symmetrical Uncertainty (SU) as the goodness measure. This method starts by sorting features through correlation with a response using symmetric uncertainty, optionally removing the bottom of the list according to some user-specified thresholds. Then, the

feature that mostly correlates with the response is selected. After that, all features that have a correlation with the selected features higher than its correlation with the responses are considered redundant and removed. Then, the feature is added to the minimal subset and the search starts again with the next feature.

In summary, the FCBF method approximates relevance and redundancy analysis by selecting all the predominant features and removing the rest of the features. It uses both class-correlations and feature-correlations to determine feature redundancy, and combines sequential forward selection with elimination so that it not only circumvents full pair-wise feature-correlation analysis but also achieves higher efficiency than pure sequential forward selection or backward elimination. It is fairly straightforward to improve the optimality of the results by considering different combinations of features in evaluating feature relevance and redundancy, which in turn increases time complexity (Yu and Liu, 2004)

5) Minimal Redundancy and Maximal-Relevance (MRMR)

The MRMR method uses the mutual information between a feature and a class as relevance of the feature for the class (Peng et al., 2005). It also uses the mutual information between features as redundancy of each feature. MRMR (Gan et al., 2014) works in the following manner: assume X is the available set of features and X_m features have been already selected from X and *y* represents class label. For selecting the next best feature, MRMR is calculated as follows:

$$\mathrm{Max}_{X_j \in X - X_m} \left[I(X_j, y) - \frac{1}{m-1} \sum_{X_i \in X_m} I(X_j, X_i) \right] \tag{2.14}$$

The MRMR measure has the following form where $I(x, y)$ is the mutual information function defined in terms of the joint probability of *x* and *y* and their marginal probabilities as follows:

$$I(x; y) = \iint p(x, y) \log \frac{p(x,y)}{p(x)p(y)} dxdy \tag{2.15}$$

This method selects the feature with minimum redundancy to previously selected features and maximum relevance to the class label; it achieves this by maximising the MRMR value. The code provided by the first author of (Peng et al., 2005) had been used in the experiments for calculating mutual information, which uses an estimation of histogram-based probabilities that are required in the calculations (Gan et al., 2014).

6) Conditional Mutual Information Maximisation (CMIM)

CMIM selects a feature subset that carries maximum relevance to the target class by using conditional mutual information (Fleuret, 2004). It works by the following iterative scheme. v(i) stands for the feature number of the i_{th} feature in selected feature subset $\{x_{V(1)},\ldots\ x_{V(l)}\}$ (full features in dataset are shown $\{x_1, x_2, \ldots\ x_N\}$):

$$v(1) = \arg\max_N I(y, x_N) \quad (2.15)$$

$$v(i+1) = \arg\max_N \{\min_{l \le i} I(y, x_N | x_{v(l)})\} \quad (2.16)$$

$I(y;x_N|x_{v(l)})$ is the conditional mutual information between target class *y* and feature x_N when feature $X_{v(l)}$ has already been chosen. By taking the feature x_N with the maximum score $\min\min_{l \le i} I(y;x_N|x_{v(l)})$, we ensure that the new feature is both more informative and different than the preceding ones, at least in terms of predicting *y*. However the weakness of CMIM is that it requires both the feature values and output classes to be binary (Yun and Yang, 2007).

2.3.3. Dependency Measures (Correlation)

Dependency measures are also identified as correlation measures or similarity measures. They measure the ability to predict the value of one variable from the value of another. In other words, it applies a hypothesis which says a good feature subset is one that contains features highly correlated to the class, yet uncorrelated to each other. A feature x_i is chosen over feature x_j if the association between feature x_i and class *y* is higher than the association between x_j and *y* (Liu and Yu, 2005).

There are several benefits of choosing linear correlation as a feature goodness measure for classification. Firstly, it helps remove features with near zero linear correlation to the class. Secondly, it helps to reduce redundancy among selected features. It is known that if data is linearly separable in the original representation, it is still linearly separable if all but one of a group of linearly dependent features is removed (Dash, 1997). However, it is not safe to always assume linear correlation between features in the real world. Also, linear correlation measures may not be able to capture correlations that are

not linear in nature. Another limitation is that the calculation requires all the features to contain numerical values.

1) **Correlation-based Feature Selection (CFS)**

CFS (Hall, 1999, Hall, 2000) evaluate a subset of features rather than individual features. The key idea of this algorithm is that it employs a heuristic evaluation that assesses the efficacy of individual features in terms of predicting the class; it also assesses how far the features are inter-correlated. The heuristic identifies all those features that are highly correlated with the target class but that have low inter-correlation levels; these are given high scores. Any features that have low correlation values with the target class will accordingly be disregarded, but redundant features need to be removed as they will be highly correlated with one or more of the remaining features (Liu et al., 2002a). In other words, CFS is useful for identifying and discarding feature-correlations which can often be found as redundant and irrelevant to the target variable (Chrysostomou, 2008). As the feature subset space is usually huge, CFS uses a best-first search heuristic. Also, symmetrical uncertainties are used in CFS to estimate the degree of association between discrete features or between features and classes (Liu et al., 2002a). CFS starts from the empty set of features and uses the best-first search heuristic with a stopping criterion of five consecutive fully expanded non-improving subsets. The subset with the highest merit found during the search will be selected (Hall and Smith, 1997).

2) **Chi-Squared (X^2)**

Setiono and Liu (1995) present a statistically justified heuristic method for supervised discretisation. It is not just a metric but a statistical test, which, in this case, can be used to evaluate the value of the Chi-squared statistic with respect to the class, using 'features *x* are independent of the class' as the null hypothesis. A numeric feature is initially sorted by placing each observed value into its own interval. The next step uses a Chi-square statistic x^2 to determine whether the relative frequencies of the classes in adjacent intervals are similar enough to justify merging x^2.

2.3.4. Consistency Measures

Consistency measures are different from the previously mentioned measures because these rely on class information feature bias in selecting a subset. These measures try to find a minimum number of features which separate classes as consistently as the original set of features can. An inconsistency is defined as two instances having the same feature values but different class labels (Liu and Yu, 2005).

1) FOCUS

Almuallim and Dieterich (1991) describe an algorithm originally designed for Boolean domains called FOCUS. It exhaustively searches the space of feature subsets until it finds the minimum combination of features that divides the training data into pure classes. There are two main difficulties with FOCUS, as pointed out by Caruanna and Freitag (1994). Firstly, FOCUS uses an exhaustive search which is intractable if many features are needed to achieve consistency. Secondly, it can be statistically unwarranted to have a strong bias towards consistency; such a scenario might just lead to over-fitting occurring for the training data.

With some simple modification of Focus, Dash and Liu (2003) refer to FocusM that can work on non-binary data with noise by applying the inconsistency rate in place of the original consistency measure. As FocusM is an exhaustive search, it guarantees an optimal solution, only on the dataset used; but it may not on the test data (Dash and Liu, 2003).

2) Las Vegas Filter (LVF)

Liu and Setiono (1996b) describe an algorithm similar to FOCUS called the Las Vegas Filter (LVF). LVF is like FOCUS, because it is consistency driven, and it is unlike FOCUS because it can handle noisy domains if the approximate noise level is 'known a-priori'. LVF randomly searches the space of subsets using a Las Vegas algorithm (Brassard and Bratley, 1996), that makes probabilistic choices to help guide them more quickly to an optimal solution (Dash and Liu, 1997). LVF generates feature subsets randomly with equal probability, once a consistent feature subset is obtained that satisfies the threshold inconsistency rate (Dash and Liu, 2003).

3) INTERACT

Feature interaction is scrutinised by the INTERACT algorithm (Zhao and Liu, 2007). Although, based on its interrelationship with the class, a single feature can be considered to be irrelevant, it might become relevant when combined with other features. Interacting features can be found through the INTERACT algorithm; a measurement of Consistency Contribution (C-contribution) along with backward elimination is used in this process. In a feature, C-contribution can be defined as an indicator that shows how substantially consistency would be affected by elimination of that feature (as for example, C-contribution of an irrelevant feature is zero). The INTERACT uses backward elimination and begins with the full feature set; based on their C-contributions, it also consecutively eliminates features one at a time based on their C-contributions. A feature is removed from the feature set if the C-contribution of a feature is found to be less than the threshold δ (a sufficiently small predefined value) (Yun and Yang, 2007).

2.3.5 Advantages and Disadvantages of Filters

In general, filter methods have been widely used for many different FS tasks. The main reason for their wide usage is the fact that they can be easily scaled to very high-dimensional datasets. They are also computationally simple, fast and are independent of the classification algorithm. Thus, the filter method needs to be performed only once. This is beneficial especially if datasets consist of thousands of features, such as gene data (Saeys et al., 2007). Although filter methods have all these advantages, they also have some disadvantages.

Firstly, univariate methods do not take into account the effects of combinations of features. This means that each feature is considered separately, thereby ignoring feature dependencies, which may lead to a poor quality of classification performance when compared to other types of FS techniques. In order to overcome this problem of ignoring feature dependencies, a number of multivariate filter techniques need to be introduced, aiming at the incorporation of feature dependencies to some degree (Saeys et al., 2007). The second disadvantage also relates to univariate filter methods, and it is that features considered to be relevant may be redundant features, which leads to selecting more features than are really required. The third limitation applies to both

univariate and multivariate filter methods; filter methods ignore the interaction with the classifier. In this way, features selected by filters may not match the classifier intended for use (Zhu et al., 2007).

2.4 Wrappers

While filter techniques treat the problem of finding a good feature subset independently of the model selection step, wrapper methods play the role of embedding the model hypothesis search within the feature subset search.

The wrapper-based FS approach has received a lot of attention due to its better generalisation performance. It relies on the performance of a specific classifier to evaluate the quality of a set of features. Here, the classification algorithm is used as a black box (Kohavi and John, 1997). Wrapper methods search through the space of feature subsets using a learning algorithm to guide the search. To search the space of different feature subsets, a search algorithm is "wrapped" around the classification model. A search procedure in the space of possible feature subsets is defined, various subsets of features are generated, and the estimated classification accuracy of the learning algorithm for each feature subset is evaluated.

To gain a general idea of how the wrapper works, it is useful to look at the way the wrapper approach starts from a given subset X_0, as it can be an empty set, a full set, or any randomly selected subset. It then searches through the feature space using a particular search strategy. Generally, it evaluates each generated subset X_i by applying a learning model to the data with X_i. If the performance of the learning model with X_i is found to be better, X_i is considered as the recent best subset. For that reason, the wrapper approach then modifies X_i by adding or removing features to or from X_i and the search iteration continues until a predefined stopping criterion is achieved (Kabir and Islam, 2010).

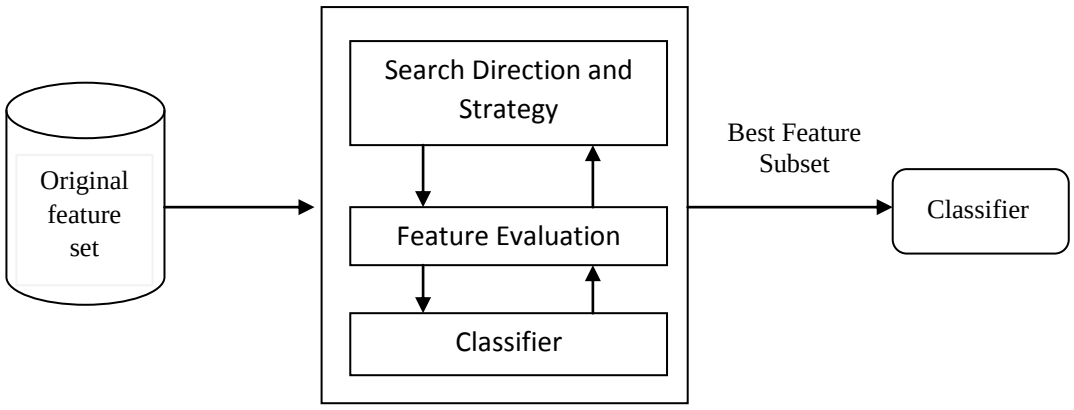


Figure 2.3: Illustration of the wrapper process

In the following Sections, wrappers are divided by their search strategies, including sequential, exponential and randomised search. Finally, these methods will be evaluated and their advantages and disadvantages will be discussed.

2.4.1 Sequential Search Techniques

Search methods are the most important part of wrapper subset selection (Devijver and Kittler, 1982). The effectiveness of the heuristic of the search determines the performance of the wrapper algorithm. Sequential search schemes add or remove features sequentially. Wrappers that perform sequential searches have a weakness of being trapped in local minima. The random algorithms inject some randomness to the search procedure to escape local minima. We will discuss the prominent feature selection search schemes in this Section.

2.4.1.1 Greedy Search

The two most commonly used wrapper methods that use a greedy search strategy are the forward selection and backward elimination search for FS (Gheyas and Smith, 2010). The selection of features involving a sequential strategy is fast and simple to implement. However, forward selection is robust to "multicollinearity problems" but sensitive to feature interaction. On the other hand, backward elimination is robust to interaction

problems but sensitive to multicollinearity. As a result, both of them can easily be trapped into local optima (Gheyas and Smith, 2010).

1) Forward Selection

The forward selection is a simple algorithm that starts with an empty set and adds one feature (or set of features) at a time until all features are considered. The features are added depending on whether they increase the performance of the learner. In the forward stage-wise selection technique, only one feature can be added to the set at one step (Raman and Ioerger, 2002).

2) Backward Elimination

The backward elimination works exactly in reverse to the forward selection. It starts with the complete feature set and drops a feature (or a set of features) and observes the performance of the learner. If the generalisation produced with the current set of features is better, then the feature is dropped and it carries on with the next feature (Raman and Ioerger, 2002).

Some researches use backward elimination but with different classifiers: a neural-network feature selector (NNFS) algorithm has been presented by (Setiono and Liu, 1997) for feature selection using neural networks as a classifier and backward elimination as the search method. It gives a different approach to other studies using decision tree methods; it shows effective results for selecting relevant features but it is very slow for high-dimensional data. Another study using neural networks has been conducted by Hsu et al. (2002); it presents a novel approach by incorporating a weight analysis-based heuristic called artificial neural net input gain measurement approximation (ANNIGMA) to direct the search in the wrapper model, and it allows effective feature selection feasible for neural net applications. It ranks features by relevance based on the weights associated with the features. The reasoning behind this heuristic is that neural net weights can be viewed as representing the gain of the input signal to the output node. Input signals that are noisy or irrelevant to the output will have a high error rate if they have high associated weights. In a similar manner, the weights of relevant and noise-free signals will be increased.

Moreover, Guyon and Elisseeff (2003), study a recursive feature elimination (RFE) which has been successfully applied to the task of gene selection by using support vector machines (SVMs) as the feature ranking method. A paired t-test is often used to

compute the probability of other subsets performing substantially better. If this probability is lower than a predefined threshold, the search is stopped. Another study using SVM (Maldonado and Weber, 2009) introduces a novel wrapper algorithm called Hold-out Support Vector Machines (HO-SVM) for FS, using SVM with kernel functions. This method is based on a sequential backward elimination, using the number of errors in a validation subset as the measure to decide which feature to remove in each iteration. It outperforms other filter methods such as the Fisher Criterion Score, and wrapper methods such as RFE-SVM and FSV, based on its ability to adjust better to a dataset because of the validation error measures, and avoiding over-fitting ensures a random split of the dataset in each iteration. However, the OH-SVM algorithm relies on backward feature elimination, which is computationally treatable but expensive if the number of input features is large; also, it uses datasets with two classes only.

In summary, sequential backward selection often finds difficulties in identifying the separate effect of each explanatory variable on the target variable, in case of high-dimensional data. Also, it is computationally expensive if the number of input features is large.

3) Bi-directional Search

In a bi-directional search, both forward selection and backward elimination are used (Doak, 1992). Convergence of the search procedure is guaranteed by not adding eliminated features and not eliminating added features. Other variants include the Plus-L Minus-R (Doak, 1992) searches, where 'R' features are deleted after adding 'L' features. If L >R we start with an empty set and if R >L then we start with the full set of features.

2.4.1.2 Floating Search Strategy

The problem with a sequential strategy is that once a feature is added (or deleted) it cannot be deleted (or added) later; this is called the nesting effect. The floating search strategy (Pudil et al., 1994) overcomes this problem by re-selecting the deleted features or deleting the already added features. This strategy is commonly used and the performance of this strategy has been found to be better compared with other search strategies and it is computationally much more efficient than some FS methods such as

Branch and Bound (Kabir and Islam, 2010). However, experimental studies demonstrate that the sequential floating forward selection (SFFS) is not superior to SFS (Bensch et al., 2005) and sequential floating backward selection (SFBS) is not feasible for feature sets of more than about 100 features (Ng et al., 1997).

In addition to that, only the wrapper method had been used with SFFS before a hybrid approach was proposed in 2006 (Somol et al., 2006), therefore SFFS had limitations in high-dimensional FS. In that hybrid SFFS, the wrapper approach is much more dominant than the filter approach. Although efficiency has been improved, it is still computationally too expensive for high-dimensional feature selection (Gan et al., 2014).

2.4.1.3 Best-first Search Wrapper

The best-first search (Ginsberg, 1993) selects the most promising but not expanded features for the search. Kohavi and John (1997) use this search for a wrapper approach; for G features there are G bits in each state and each bit indicates whether a feature is present (1) or absent (0). Compound operators are used to connect states. The operators used are the addition or deletion of a single feature. The search is started, aiming to find a state with maximum prediction accuracy. Because of the complexity of the search space O (2^N), the state space search is stopped if there is no improvement in accuracy after a number of attempts. The authors found that their wrapper performed better than Relief when used with ID3 and Naïve Bayes classifiers (Kohavi and John, 1997).

2.4.2 Exponential and Randomised Search Algorithms

2.4.2.1 Beam Search

The Beam search (Aha and Bankert, 1996) is similar to the best-first search except that only the best K features at each level are placed at the beginning of the search queue and are used for further searches. The Beam search becomes exhaustive if there are no bounds on queue size. If the queue size becomes one, it becomes a forward selection search. The beam search is extremely powerful on datasets with a small instance space and large number of features.

2.4.2.2 Simulated Annealing

Simulated Annealing (Kirkpatrick and Vecchi, 1983, Haykin, 1994) is another application of a stochastic optimisation search scheme to FS. In simulated annealing, the system state is subjected to a small random change and it accepts the new state if it is better than the previous state. In the case of FS, the transformation will consist of adding or removing the features.

2.4.2.3 Genetic Algorithms

Genetic algorithms (Mitchell, 1997) begin from a random initial population and generate a better population by mating or crossover between pairs of solutions, and they try to improve their fitness or some objective function. The instance space is represented using bit strings indicating whether a feature is present (1) or absent (0). One of the first studies that used the genetic algorithm in the FS method is called ADHOC (Richeldi and Lanzi, 1996) and it consists of two steps. In the first step, ADHOC identifies irrelevant features by constructing a profile for each feature. In the second step, it uses genetic algorithms to find a subset of the most important features.

2.4.3 Advantages and Disadvantages of Wrappers

Wrapper approaches include the interaction between the feature subset search, the model selection and the ability to take into account the main functionality of feature dependencies (Saeys et al., 2007). Thus wrapper methods have also the ability to select more accurate feature subsets than the filter methods (Li and Guo, 2008). Although they often achieve very good classification accuracies, they also have some disadvantages.

The main disadvantage of the wrapper is that it depends on the classifier. There is a higher risk of overfitting than with filter techniques and it is very computationally intensive when the number of features available for selection and the samples are too large. In particular, it has a high computational cost when building the classifier (Saeys et al., 2007).

2.5 Hybrid

The hybrid approach was proposed to handle large datasets and to overcome the limitations of both an independent measure (filter) as well as a dependent measure (wrapper) and to provide reasonably efficient and accurate selection (Singh and Silakari, 2009). It is similar to the filter approach in the search step, where it selects a small number of candidate subsets of features. The hybrid approach evaluates the quality of a small number of candidate subsets, which leads to speeding up the model. The selected features produce the best classification accuracy. Therefore, the hybrid approach is less expensive than a wrapper and more effective than a filter.

Hybrid algorithms have various forms. One of the typical forms is a single filter and a single wrapper (SFSW). However, there are others forms which are known as ensemble feature selection (EFS), similar to multi-filters (MF), multi-wrapper (MW), multi-filters signal wrapper (MFSW), single filter multi-wrapper (SFMW) and multi-filter multi-wrapper (MFMW). In the following Sections, the hybrid approach is divided based on search strategies, which include sequential searches with hybrid evaluation and random searches with hybrid evaluation.

2.5.1 Sequential Searches with Hybrid Evaluation

A new group of hybrid search methods exists which is a two-phase hybrid search: firstly filter ranking or subset creation and secondly, a sequential forward search along with wrapper evaluations, in order to guide the search. The concept behind this new group is to find out the application of a filter measure that can obtain a ranking of the relevance of attributes with respect to the class. Afterwards, a sequential algorithm is applied; the algorithm is carried out to go through the ranking by incrementally adding variables that are completely relevant to the classification process. Here a wrapper method is used to measure the relevance of the inclusion of a new variable. This approach retains a considerable portion of the wrapper advantages, which is the main advantage of using it. This approach also reduces the computational cost to O(n) and, unlike pure wrapper approaches, wrapper evaluations happen instead of $O(n^2)$. This advantage makes the distinction between the task becoming computationally attainable or not, while dealing

with thousands of variables (Bermejo et al., 2008). Some well-known sequential searches with hybrid evaluations are listed below:

1) Filter Dominating Hybrid Sequential Floating Forward Selection (FDHSFFS)

FDHSFFS was proposed by Gan et al. (2014) using two filters: MRMR (Peng et al., 2005) and the Davies Bouldin index (DBI) (Davies and Bouldin, 1979), and three wrappers: LDA, SVM and K-nearest neighbour (KNN). This research aims to avoid the complexity of Wrapper Dominating Hybrid Sequential Floating Forward Selection **(**WDHSFFS) (Somol et al., 2006) by controlling the number of features pre-selected by the filter and passed to the wrapper, as well as improving the efficiency. The novelty of this study is mainly in the strategies of adding and deleting steps, where a filter is only applied to compare feature subsets of the same cardinality when selecting a new feature or removing an existing selected feature, while a wrapper is applied to compare the selected best feature subset of different cardinalities. The result of FDHSFFS is compared with SFFS (pure wrapper) and WDHSFFS and it shows that WDHSFFS is faster than SFFS but FDHFSFFS is 10 times faster than WDHSFFS with similar classification performances when the dimensionality is very large. Moreover, in terms of accuracy, FDHSFFS outperforms WDHSFFS when MRMR is used as a filter.

2) Best Incremental Ranked Subset (BIRS)

Best Incremental Ranked Subset for FS (BIRS) (Ruiz et al., 2006) first produces a filter ranking and then it performs an incremental best-first selection throughout the ranking. Firstly, the features are ranked by symmetrical uncertainly. Secondly, the algorithm deals with the list of features once, crossing the ranking from the first feature to the last ranked feature. The classification accuracy with the first feature in the list is obtained and it is marked as selected. Then, the classification rate is obtained again with the first and second features. The second feature is selected depending on whether the accuracy obtained is significantly better or not. The process will be repeated until the last feature on the ranked list is reached. Finally, BIRS returns the best subset found, and it is stated that it does not contain irrelevant or redundant features.

3) Best Agglomerative Ranked Subset (BARS)

Best Agglomerative Ranked Subset for FS (BARS) (Ruiz et al., 2008) is iterating between two phases: (a) ranking of subsets (CFS-SU, wrapper) and (b) generation of new candidate subsets by combining (based on wrapper evaluation) those previously ranked. 'BARS' allows the evaluation of a reduced number of candidate subsets and it obtains very compact subsets. In this method, non-linear correlation (CFS) is used as an evaluation measure. Two subset evaluation measures are used, one for each type of approach (wrapper and filter-CFS). For instance, $CFBA^{CF}$ shows that CFS-SU will be used as an individual measure in the first part and CFS as a subset in the second part, and $CFBA^{WR}$ shows that NB or C4.5 classifier will be used as a subset evaluator in the second part.

4) Linear Forward Selection (LFS)

Linear forward selection (LFS) (Gutlein et al., 2009) is a simple complexity optimisation of SFS. It starts by ranking the features based on Symmetrical Uncertainty (SU), then it selects the top-K features; after that, the SFS search method is run over the selected features. LFS limits the number of features that are considered in each step, so this significantly reduces the number of evaluations, and thus improves the runtime performance of the algorithm. Gutlein and his colleagues investigated two methods for limiting the number of features, including: Fixed Set which firstly ranks all features and simply selects the top-K ranked features as input to forward selection; and Fixed Width which keeps the number of extensions in each forward selection step constant to a fixed width K.

5) Incremental wrapper-based subset selection (IWSS)

Incremental wrapper-based subset selection (IWSS) (Bermejo et al., 2008) starts by using SU to evaluate the predictive features that are ranked in increasing order; that is, more important features are placed first. IWSS uses a relevance criterion to decide when a new feature must be included in the selected subset. The relevance criterion is based on a t-test as an alternative to just comparing the mean accuracy, and the results show that the use of this relevance criterion frees the algorithm from noise. As a result, more compact subsets can be obtained with similar accuracy, considering another statistical test (the Wilcoxon signed rank test) and a simple heuristic criterion.

6) Incremental wrapper-based subset selection with replacement (IWSSr)

Incremental wrapper-based subset selection with replacement (IWSSr) (Bermejo et al., 2009) seeks to alleviate some of the weaknesses of using IWSS, the essential one being its greedy behaviour. IWSS always tries first the best ranked features, and once a feature is included in the selected set, it is preserved there until the search stops. Therefore, the IWSSr design obtains more compact subsets, and allows not only the addition of new features, but also an interchange with some of the already included features in the selected subset. In this technique, relevant features that become irrelevant can be eliminated from the selected subset instead of preserving both. However, IWSSr increases the worst-case complexity of IWSS up to $O(N^2)$, although, as in the case of the SFS (pure wrapper), the actual number of wrapper evaluations is found to be considerably smaller.

7) Incremental Wrapper Subset Selection by Re-ranking

Bermejo et al. (2011) propose a new technique that aims to significantly reduce the number of wrapper evaluations while maintaining good performance (e.g. accuracy and size of the obtained subset). The search starts by ranking all the features, then the ranking is split into blocks of size G, and an incremental filter-wrapper algorithm is applied, but only on the first block. Let X_i be the subset of features selected from this first block. Then the rest of the ranking is re-ranked again but the previously selected subset X_i is taken into account. The incremental filter-wrapper algorithm is run again over the first block in this new ranking, but the X_i subset is selected for initialisation instead of the empty set and so on. The search stops when no feature is selected in the current block. This search leads to a reduced number of re-ranks, which means that only a few blocks and features need to be analysed in this method, but it does not decrease the accuracy of the output obtained. Even the size of the selected subset is reduced.

2.5.2 Random Searches with Hybrid Evaluation

1) ReliefF-GA-Wrapper

The ReliefF-GA-Wrapper (Zhang et al., 2003) aims to gain the advantages of both filter and wrapper. It starts by running ReliefF to rank the original features, and the resulting estimation is embedded into genetic algorithms. It applies to a search for the optimal

feature subset with the training accuracy of the classifier. The ReliefF-GA-Wrapper has better performance than the ReliefF and GA-Wrapper methods.

2) Hybrid Genetic Algorithm (HGA)

The Hybrid Genetic Algorithm (HGA) (Huang et al., 2007) presents a method for FS which contains the filter and wrapper approaches in a cooperative manner. It uses mutual information as a local search to rank features, and genetic algorithms as a global search to find a subset of important features from the ranked features.

3) Wrapper and filter feature selection algorithm (WFFSA)

The wrapper and filter feature selection algorithm (WFFSA) (Zhu et al., 2007) presents a novel hybrid method using a memetic framework. It integrates a filter ranking method into the traditional genetic algorithm (GA) to improve classification performances and it speeds up the search by identifying the important feature subsets. Furthermore, the authors investigate a number of key issues of memetic algorithms (MA) to identify a good balance between local search (LS) and GA to maximise search quality and efficiency in the hybrid filter and the wrapper MA. In the first step, the method adds or deletes a feature from a candidate feature subset based on the filter ranking method. Then, the GA population is initialised randomly, with each chromosome encoding a candidate feature subset. Subsequently, on all or portions of the chromosomes, LS is applied. Genetic operators are then used to generate the next population. This process repeats until the stopping conditions are satisfied.

2.5.3 Advantages and Disadvantages of Hybrid Methods

In general, the family of incremental wrapper-based subset selection outperforms most of the hybrid algorithms for the reason that it is a very fast search through the feature space, and any classifier can be embedded into it as an evaluator. Also, the evaluation is much less expensive as only a few features are selected. However, due to its greedy behaviour, it always tries first the best ranked features and once a feature is included in the selected set, it is preserved there until the search is stopped. Consequently, many studies have been conducted in order to alleviate these disadvantages. IWSSr and WDHSFFS arise as the better choices, because they allow not only the addition of new attributes but also their interchange with some of those already included in the selected subset; they also include fewer features in the selected subset. But the disadvantage here

is that time complexity grows up to $O(N^2)$, the same as with SFS. While FDHSFFS outperforms the above methods, it has the advantage of IWSSr and WDHSFFS but it is faster than them.

On the other hand, the hybrid approach with single filter single wrapper still has drawbacks, such as the selected features depend on the choice of a specific filter and wrapper. Consequently, an ensemble feature selection approach uses multiple filters and/or multiple wrappers; it is another way to identify potential and reliable features and also to improve the accuracy and robustness of the classification.

2.6 Introduction to Ensemble

An ensemble in the context of machine learning can be broadly defined as "a machine learning system that is constructed with a set of individual models working in parallel and whose outputs are combined with a decision fusion strategy to produce a single answer for a given problem"(Wang, 2008). Also, it can be described as machine learning methods that leverage the ability of multiple models to achieve better prediction accuracy than any of the individual models on their own (Oza, 2000). The models can be classifiers, predictors or filters, depending on the types of task, such as classification, prediction, regression or clustering, that the ensemble is designed to do. The rationale behind the ensemble approach is based on the bare fact that no individual models can be perfectly developed for solving non-trivial real-world problems.

The performance of an ensemble can be evaluated in terms of complexity, stability and accuracy. Complexity is concerned with the computational time and memory space required and can be measured in the usual ways, however, it is a not major problem because computing power and resources can usually cope with most applications except in case of extremely large and complex problems. Stability of an ensemble is about the level of stability of the answers produced by ensembles. It may be measured by the probability that a model would be chosen incoherently from an ensemble, and the probability of success or failure of those models on randomly selected test data is the stability. However, in practice, it is the accuracy that people are more interested in, as achieving a similar or higher accuracy with reliable results is one of the main motivations for using ensemble methods (Wang, 2008).

2.6.1 Methods for Constructing Ensemble

Many methods for constructing ensembles have been developed. Here we will review general purpose methods that can be applied to many different learning algorithms (Dietterich, 2000)

1) By Manipulating the Training Set

In this approach, the original data is re-sampled according to some sampling distribution to create multiple training sets. Then, a classifier is built from each 'training set' using a particular learning algorithm. Some studies have shown that this approach works especially well for unstable learning algorithms such as decision tree, neural network, and rule learning algorithms. While, linear regression, nearest neighbour, and linear threshold algorithms are generally very stable (Dietterich, 2000). Also, bagging and boosting are two examples of ensemble methods that manipulate their training sets (Tan et al., 2006).

2) By Manipulating the Input Features

In this approach, from each training set, a subset of input features is chosen. The subset can be either chosen randomly or based on some methods. This approach works very well with datasets that contain highly redundant features. Random forest is an ensemble method that manipulates its input features and uses decision trees as its base classifiers (Tan et al., 2006).

3) By Manipulating the Class Labels

This approach can be used when the number of classes is large. The training data is transformed into a binary-class problem by randomly partitioning the class labels into two disjointed subsets. An example of this approach is the error-correcting output coding method (Tan et al., 2006).

4) By Manipulating the Learning Algorithm

This approach can be applied to the algorithms several times on the same training data and may result in different models. For example, an artificial neural network can produce different models by changing its network topology or the initial weights of the links between neurons. Also, an ensemble of decision trees, instead of choosing the best

splitting features at each node, can randomly choose one of the top-K features for splitting (Tan et al., 2006).

The first three approaches are general methods that are appropriate to any classifiers, while the fourth approach depends on the type of classifier used.

The base learners for most of these approaches can be generated as: parallel (all at once) or serial (one after another). The former combines independently constructed and diverse base learners; Random Forest (RF) is an example of a parallel ensemble. In serial ensembles, each new learner relies on previously built learners so that the weighted combination forms an accurate model. The Adaboost algorithm was introduced by Freund and Schapire (1996) and it is an example of a serial ensemble. Also, boosting shows dramatic improvement in accuracy even with very weak base learners (such as decision stumps, single split trees) (Tuv et al., 2009).

2.7 Ensemble of Feature Selection

Feature selection has become the essential step in many data mining applications. However, using a single feature subset selection method may generate local optima. Ensembles of feature selection (EFS) methods attempt to combine multiple FS methods instead of using a single one. EFS techniques can be superior to the individual feature selection techniques. The reasons for using EFS techniques are various. Firstly, different FS methods produce different feature subsets, so combining different "opinions" from different FS methods appears to be a rational result. Secondly, each FS method has its own ability to search in the dataset, so may yield equally optimal results, while EFS combines the search abilities of each FS method in order to obtain the more important results. Thirdly, different feature subsets produced by different FS methods may show complementary effects because of the non-independence between features, therefore aggregating these subsets may give better approximation to the optimal subset or ranking of features (Yang and Mao, 2011). Fourthly, in microarray data, it is often reported that several different FSs may produce equally good results, but different subsets and EFSs may reduce the risk of choosing an unstable subset (Saeys et al., 2008).

2.7.1 The Ensemble Idea for Feature Selection

It can be stated that the ensemble idea for feature selection is somewhat similar to the development of ensemble models for supervised learning. Two essential steps can be identified while creating a feature selection ensemble: a set of different feature selectors are created in the first step, and then each selector provides their output. The results of the single models are aggregated in the second step. Several methods can achieve the variation in the feature selectors, such as different feature selection techniques, stochasticity or haphazardness in feature selectors, perturbations at the instance level, feature level perturbation, as Bayesian model averaging. These techniques or combinations of these techniques are used (Dietterich, 2000, Hoeting et al., 1999). Weighted voting can be used to aggregate the different feature selection results.

2.7.2 Combination Methods of Ensemble Feature Selection

There are two main types of aggregation methods based on the nature of output of the feature selection – whether it is ranking of features or subset of features. In the case of ensemble ranking, average ranking or average ranking score will be used. In the case of ensemble subset, counting the most frequency feature will be applied. In fact, the ensemble of subset feature had been rarely studied, while ensemble ranking possesses more intentions, as illustrated below:

A general formulation for the ensemble ranking *f* is obtained by summing the ranks over multiple samples or over multiple filters; and w_i denotes a bootstrap dependent weight (Abeel et al., 2010):

$$f = \left(\sum_{i=1}^{t} w_i f_i^1 \dots\dots \sum_{i=1}^{t} w_i f_i^N\right) \qquad (2.17)$$

In order to create the ensemble result, this method uses the complete ranking of all the features. Afterwards, the ensemble ranking *f* is obtained; it is done by simply carrying out a summation of the ranks over all filters (or over multiple samples). This amounts to assigning all the weights w_i equal to 1 in the general formulation (2.17). K features with the lowest summed rank are selected from *f*. This is done in order to select the final set of features for a signature of size K (Abeel et al., 2010). In addition, the existing ensemble ranking by only the feature score methods uses various aggregate functions

such as mean, median etc. (Olsson and Oard, 2006). In the ensemble mean, each feature's score is determined by the average of the ranking scores of the feature in each ranking list, while in the median combination, each feature's combined score is the median score in all ranking lists (Wang et al., 2011).

In general, there are two ways in which an EFS can be performed. They are ensemble of a single feature selection technique with instance level perturbation, and ensemble of multiple feature selection techniques. In this thesis, the ensemble of multiple feature selection techniques will be used.

2.8 Researches in Feature Selection Ensemble

2.8.1 Ensemble of Single Feature Selection Technique with Instance Level Perturbation

In the ensemble of a single feature selection technique, bootstrap aggregation and other algorithms can be used to generate different bags of data. For each of the bags, a separate feature selection is performed, and the ensemble is performed by aggregating the single set by weighted voting in the case of ranking, using linear aggregation (Saeys et al., 2008). Bootstrap aggregating, also known as bagging, is a technique used to generate multiple versions of data. The multiple versions are formed by making a bootstrap replication of the dataset and using these as datasets for model fitting.

In general, the aim of this category of EFS is to produce more robust and stable results than using only a single run of the FS method. In addition, the accuracy performance and the quality of the final feature subset is selected. Both accuracy performance and stability should be considered while evaluating an FS algorithm, because a stable but classified ineffective FS result does not make any sense. Also, most of these studies focus on the microarray datasets which always have a large number of features and a small number of samples.

- **Existing Ensemble Methods for a Single Feature Selection Technique**

Saeys et al. (2008) examined two aspects of EFS techniques: stability and classification performance in the bioinformatics domain. They used four FS techniques: two filter methods (Symmetrical Uncertainty and ReliefF) and two embedded methods (Random Forests and linear SVM). For each of the four FS techniques, an ensemble version was created by instance perturbation. Bootstrap aggregation was used (Breiman, 1996), to generate 40 bags from the data. For each of the bags, a separate feature ranking was performed, and the ensemble was produced by aggregating the single rankings by using voting linear aggregation. In order to assess the stability, they compared feature ranking using the Spearman rank correlation, and the feature subset using the Jaccard index, choosing the top 1% and top 5% best features of the ranking. In terms of stability, the result showed that EFS provides most robust results than a single FS method; in particular, Random Forests clearly outperforms other FSs. On the other hand, in terms of classification performance, the EFS technique is better than just using the full set of features but it is similar to, or slightly better than, using FS without an ensemble, i.e. using only a single feature selector such as Symmetrical Uncertainty, ReliefF or linear SVM. The exception to this was the Random Forest feature selection technique, which, when used in an ensemble, performed poorly – worse than either using it on its own or just using the full set. Therefore, the substantial increase in robustness affects the result of accuracies for all datasets.

Also, Abeel and his colleagues (2010) discuss the stability and classification performance of biomarker identification on four cancer diagnosis datasets using EFS methods. Support Vector Machines (SVM) with recursive feature elimination algorithm (RFE) is used to aggregate the different rankings, obtained by bootstrapping the training data. A linear SVM is estimated from the training samples, and features are sorted according to the absolute value of their weight in the hyper-plane. Then RFE is started from the full feature set which adopts a backward elimination strategy to iteratively remove the least important features. To aggregate the different rankings, obtained by bootstrapping the training data into a final ranking, they propose two aggregation schemes: complete linear aggregation (CLA) and complete weighted linear aggregation (CWA). The results of the CLA and CWA ensemble methods clearly improve upon the baseline, both in terms of stability and classification performance. Moreover, the gains increase as signature sizes become smaller. In three out of four datasets, the ensemble

methods even perform better with fewer than all features, thus showing that ensemble methods are more capable of eliminating noisy and irrelevant dimensions.

Recently, Yang et al (2011) proposed EFS using multiple runs of an unstable filter: ReliefF (Robnik-Šikonja and Kononenko, 2003) and tuned ReliefF (TuRF) (Moore and White, 2007), aiming to increase the stability and power of gene-gene interaction filtering. They found that these filters are sensitive to the order of the samples in the dataset which leads to unstable and sub-optimal results. Therefore, they assume that aggregating the results generated from the multiple runs of the filter may improve filtering performance. Therefore an ensemble approach has been proposed which extends the idea of a classification-oriented ensemble feature selection (Abeel et al., 2010). It uses a bootstrap sampling procedure with multiple filters to produce different rankings, then uses a rank score aggregation approach. The results show that TuRF-E performs the best in the average cumulative success rate in all cases examined in their study, regardless of the sample size or heritability of the simulated datasets.

2.8.2 Ensemble of Multiple Feature Selection Techniques

Ensembles of multiple feature selection techniques combine outcomes of various feature selection techniques. Two steps are essential in creating a single feature subset from a multiple feature selection set. First, a set of different feature selections is created and in the second step, these sets are combined to produce a final set of selected features.

- **Existing Ensemble Methods for Multiple Feature Selection Techniques**

The earliest study on ensembles of multiple feature ranking techniques was done by (Olsson and Oard, 2006); they conducted studies on ensembles of multiple feature ranking techniques, in order to resolve text classification problems. They used 3 filters: document frequency thresholding, information gain, and the Chi-square method (χ^2max and χ^2avg).

After that, Wang and his colleagues (2010a) also studied the EFS of 6 filter-based rankers, and later in other research, (Wang et al., 2010b, Wang et al., 2012) examined EFS methods for predicting faulty program modules. 18 different filter-based ranking

techniques (7 well-known commonly used filter-based feature ranking methods in addition to 11 threshold-based feature selection [TBFS] techniques) were proposed and implemented by their research group within the Waikato Environment for Knowledge Analysis (WEKA) (Witten and Frank, 2005). The aggregated method used in this study is arithmetic mean, where each feature's score is determined by the average of the ranking scores of the features in each ranking list; the highest ranked attributes ($\log_2 N$) are then selected from the original data. They examined the performance of classifiers with selected features using 17 different ensembles of rankers. The classification performance results show that no particular ensemble method outperforms the others in most cases, but in general, the ensembles of very few rankers usually perform similarly or even better than ensembles of many or all rankers.

The same idea of aggregating multiple filters by mean rank aggregation measure was proposed by (Sarkar et al., 2012). Also, in 2013 the same author (Sarkar et al., 2013) proposed a robust correlation-based feature selection method using rank aggregation (CRA) which consists of two main steps: CFS filter step then rank aggregation step by using 3 filters: IG, SU and Chi-x^2. After that, their idea was extended in 2014 (Sarkar et al., 2014) by developing the rank aggregation-based FS method with Kemeny and Borda. They used 3 filters, IG, SU and Chi-x^2, with 5 different classifiers over 8 datasets. They also used post-processing steps to generate a feature subset from the final rank aggregation feature set. Firstly, for each classifier, they determined the classification accuracies from the first top feature to the total number of features in the feature subset. Then, they selected the feature subset with the maximum classification accuracy across all the classifiers used as final feature subset and considered this subset as the optimal subset. The experiment shows that the classification accuracy improves by approximately 3-4% compared with using a single filter.

Furthermore, there are some studies that used a wrapper after the fusion of a different number of filter methods. (Min and Fangfang, 2010) proposed a Filter-Wrapper Hybrid Method (FWHM) to optimise the efficiency of feature selection. FWHM is divided into two phases; in the first phase, the fusion of 6 different filter methods (correlative family selection, Relief, class separability, Mahalanobis distance, multivariate correlation coefficient and mutual information) are adopted to obtain a better pre-selection feature subset. They give weights to different rankings and then combine the multi-ranking orders through the weighted average value of each feature. Then the initial antibody is generated which is based on the weights of the pre-selection feature subset rather than

randomly obtaining the joining of the two phases. In the wrapper phase, the improved clonal selection algorithm (CSA) (De Castro and Von Zuben, 2000) is used to carry out additional FS and to obtain the final feature subset. Key features have more opportunity to be selected, with the help of the weights that are submitted from the filter phase. The results show that FWHM can improve both the efficiency and accuracy of the FS.

In addition to that, in the same year, (Yang et al., 2010) proposed a similar methodology aiming to improve the hybrid system for gene selection based on a recently proposed genetic ensemble (GE) system. In a multi-filter enhanced genetic ensemble (MF-GE) system, the gene selection process is sequentially divided into two phases. In the filtering phases, multiple filtering algorithms (Chi-x^2, ReliefF, SU, IG and GR) are applied to give scores for each candidate gene in the microarray dataset. The scores of each gene are then integrated for wrapper process. In the wrapper phases, the GE algorithm is used to select discriminative genes using the information provided by the filtering process. The detail of this genetic ensemble algorithm is described (Zhang et al., 2009).

On the other hand, Gheyas and Smith (2010) presented an ensemble of two wrappers: simulated annealing (SA) and genetic algorithm (GA), for selecting optimal feature subsets efficiently, without including any filters. SAGA generalised regression neural networks and a greedy search algorithm by combining the ability to avoid being trapped in a local minimum of SA with a very high rate of convergence of the crossover operator of GA. Unlike existing hybrid algorithms, SAGA does not compromise accuracy for speed. The strength of SA is good global search ability, while its major disadvantage is its slow convergence speed. On the other hand, GA implements both crossover and mutation operations and the strength of GA is its rapid convergence, but the combination of crossover and a low fixed mutation rate often traps the search in a local minimum. In addition, the local search capability of SA and GA is weak. By contrast, greedy algorithms have good local search ability, but lack global search ability.

Leung and Hung (2010) propose a multiple filter multiple wrapper (MFMW) approach that makes use of multiple filters and multiple wrappers to improve the accuracy and robustness of the classification, and to identify important genes. The MFMW approach works as follows: a number of filters are employed, each for selecting a predefined number of genes. The filtered gene subset is formed by taking the union of the lists of the genes obtained by all the filters. After that, the genes are selected by means of a

wrapper consisting of multiple classifiers; because different classifiers may provide different classification labels for the same sample, there is a need to resolve this conflict when it occurs. It is natural to resort to some kind of voting scheme among the classifiers. Two possibilities are majority voting and unanimous voting. Leung and Hung (2010) have chosen to use unanimous voting to decide on the overall classification output based on the outputs of the classifiers.

Moreover, Yang and Mao (2011) proposed a multi-criterion fusion-based recursive feature elimination (MCF-RFE) algorithm aiming also to improve the stability and classification performance of the FS results. The FS methods used in the study are 3 filters: Fisher's ratio, Relief, ADC (asymmetric dependency coefficient) and one embedded method: AW-SVM (absolute weight of SVM). Both score-based and ranking-based fusion methods are used. After the aggregated feature is ranked then the RFE search strategy is applied to remove a portion of the worst features, then the second iteration is run until the stop criterion is satisfied. The results of 5 microarray datasets show that the MCF-RFE algorithm outperforms the SVM-RFE in classification performance with reasonably good stability (Guyon et al., 2002).

Recently, Fahad, et al. (2014) proposed a robust approach called the Global Optimisation Approach (GOA) to discover both the most important and stable features across different traffic datasets, by using multi-filters and an information theoretic method. Firstly, GOA starts by aggregating the output of 6 filters by counting the frequency of each feature, then it ranks them based on their frequency. Secondly, the feature subsets propose an adaptive threshold to compute a cut-off and to automatically cull robust features from unstable selected features, in order, so as to extract the stable features. Finally, a new goodness measure based on a Random Forest framework is proposed to estimate the final optimum feature subset. The data used in this study is a network traffic data in spatial and temporal domains. The results show that GOA outperforms the commonly used FS methods for traffic classification tasks in terms of accuracy and stability, but the pre-processing time of GOA is more computationally expensive.

While there are a number of studies that attempted to propose ranking aggregation methods, only a small number of studies have been focused on comparing the existing rank aggregation methods. In fact, the comparison between these methods is important to help the researchers understand which aggregation methods are part of the same

group, and how these groups behave when applied to different problems. Also, it helps to select simpler methods if two methods produce similar results but with very different complexity. A number of studies compares between these methods which are discussed in the following Sections:

Prati (2012) proposed an EFS framework using 6 ranking filters with different ranking aggregation methods, which are Borda (BC), Condorcet (CD), Schulze (SSD) and Markov Chain (MC4), aiming to evaluate the classification performance of the implemented ranking aggregation methods and to compare them with single filter. An extensive evaluation using 39 datasets, 3 classifiers and 3 different performance measures show that EFS provides better feature ranking than a base ranking filter. Also, the SSD ranking aggregation method is considered to be the best method for overall comparison with all classifiers and performance measures. However, Condorcet, Schulze and Markov Chain are computationally expensive and not suitable to cases of extremely large search spaces (Wald et al., 2012).

Also, Wald and his colleagues (2012) made an extensive comparison of 9 rank aggregation methods in terms of similarity. They used mean, median, lowest rank, highest rank, robust rank aggregation (Kolde et al., 2012), stability selection (Haury et al., 2011), exponential weighting (Haury et al., 2011), enhance Borda (Wald et al., 2012) and round robin (Neumayer et al., 2011). They found a number of groups with similar rank aggregation techniques, as follows: the first group consisted of mean, median, stability selection, exponential weighting, enhance Borda and robust rank aggregation, and the second group consisted of highest rank and round robin, while the lowest rank aggregation was not similar to any ranking techniques. Also, we can note that two of the well-known ensemble types, mean and median, are each mathematically equivalent to more complex methods, as long as all the lists being aggregated are full lists. Mean aggregation is equivalent to the Borda and median is equivalent to the Spearman footrule (Wald et al., 2012).

Recently, Burkovski et al. (2014) have analysed different aggregation methods by separating them into two groups: early and late aggregation. They have classified mean and median as early aggregation methods, while Borda (Dwork et al., 2001), Copeland's (Copeland, 1951), Robust Rank Aggregation (Kolde et al., 2012), Pick-a-Perm (Ailon et al., 2008), Speman's Footrule and Canberra Distance were classified as late aggregation methods. In early aggregation, features are first aggregated then ranked, while in late

aggregation, features are first transformed into an ordinal scale then aggregated into a consensus ranking using different methods. The experimental results on real datasets show that Broda's and Copeland's methods are on the par with the mean, but they are more robust predictors then the median. Moreover, it is found by Wang et al. (2011) that mean performs better than median in terms of accuracy.

2.9 Summary

This chapter presented an introduction of feature selection and reviewed the commonly used feature selection methods, namely filters and wrappers, in addition to hybrid. Filter methods do not use classifiers but instead use statistics and the general characteristics of the data to determine relevant features. However, wrapper methods rely on classifiers to select the most relevant sets of features. This means that filters are classifier-independent and wrappers are classifier-dependent, while hybrid was proposed to overcome the limitations of both independent and dependent measures. Nevertheless, many studies have shown that the hybrid approach with single-filter-single-wrapper still has drawbacks, such as the fact that the selected features are dependent on the choice of a specific filter and wrapper.

This chapter also introduced the ensemble and briefly reviewed the methods for constructing an ensemble. Finally, it described in detail EFS and presented the existing studies in this area. Since an EFS approach uses multiple filters and/or multiple wrappers, it is a way to identify potential and reliable features and also to improve the accuracy and robustness of the classification. However, there are still several open research questions in this research field.

In this thesis, we focus on some of these open research questions such as which members should be used in EFS and how many, which consensus methods should be used and whether all the members should be treated equally. We ask these questions along with the main question, which is how to develop an ensemble of feature selection that can improve the stability and performance of the selected features.

In the next chapter, we will present a general framework of the proposed ensemble of feature selection and will describe in detail the evaluation method we used in this thesis to assess the performance of EFS by measuring the classification performance and the stability.

Chapter 3

Methodology

3.1 Introduction

A review of two main topics was presented in the previous chapter, including the methods in feature selection and ensemble. In the feature selection, we presented the details of the three methods (filter, wrapper and hybrid) with an explanation of their characteristics, and presented the research for each category. Moreover, in terms of the ensemble, we explained the methods for constricted ensemble. However, as pointed out in the previous chapter, using a single feature selection method may result in generating local optima. Ensembles of feature selection (EFS) methods can be superior to the individual feature selection techniques and the reasons for that are various, as described in Section 2.7.

This chapter presents a general framework of the proposed ensemble of feature selection in Section 3.2. How to use data in FS is described in Section 3.3. The evaluation methods are also explained in Section 3.4 to better understand how the proposed ensemble of feature selection can be evaluated by measuring the stability and classification accuracy. The comparison strategies used in this study are described in Section 3.5. In Section 3.6, the system software design is discussed, and finally, in Section 3.7, the experimental design is presented with details.

3.2 Proposed Ensemble of Feature Selection

The proposed ensemble approach (as illustrated in Figure 3.1) is performed through using ensemble of filters with a consensus function in order to improve the overall results in term of accuracy and stability. In order to explain the proposed ensemble, each step is discussed here separately, clarifying the concept underpinning each and describing the issues related to each step.

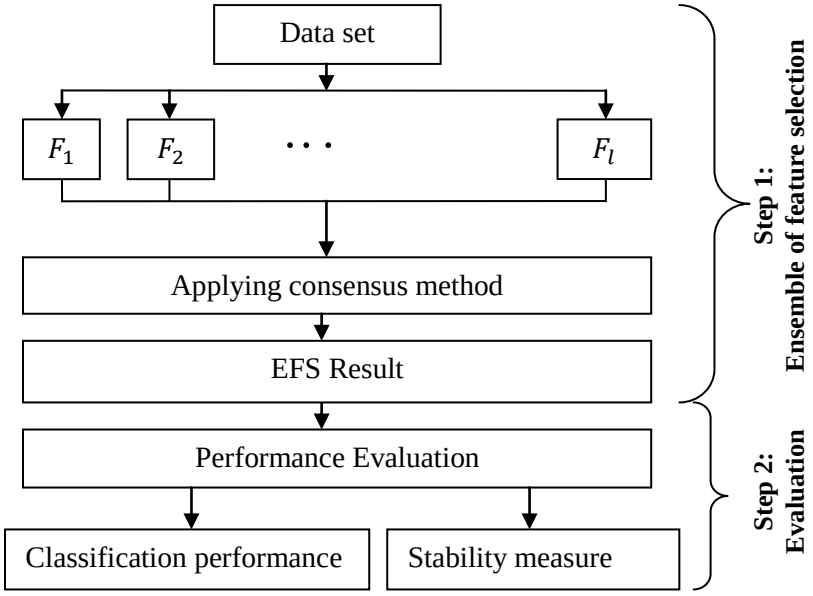


Figure 3.1: The proposed ensemble of feature selection

The first step begins with selecting a number of filters as members in the ensemble. This step entails determining the most appropriate filters to be included in our ensemble, which involves a further analysis of the literature. Many researchers have conducted comparisons between various numbers of filters and then sought to conclude which filters were found to be better than others in some cases, also explaining reasons for using some filters more frequently than others. Although various filters are available in the literature, they differ widely in functionality and some merely deliver collections of local optima identified within the space of the feature subsets. Moreover, no particular filter outperforms any other in all cases. For these reasons, we were motivated to adopt the concept of an ensemble of filters, rather than adopting just a single filter, in order to deliver results that are more stable and accurate.

The key issues to be taken into consideration in this study are the types of filters and the quantity of filters that should be included in the proposed ensemble. In this study, we will categorise the filters based on evaluation criteria into groups broadly based on the following studies (Fahad et al., 2014, Liu and Yu, 2005, Saeys et al., 2007): distance, information, dependency, statistics and consistency. After that, we will study the popular filters under each of these categories in order to be able to choose the appropriate filters from each category. It should be noted that each filter we will choose will use a different criterion for evaluating the relevance of the candidate features in the

datasets. When combined, candidate features are assessed from many different aspects. Moreover, diversity may be achieved in this work by using various filters.

In terms of determining the number of member filters, we will follow the guidelines given in Wang et al. (2012) – that is, an ensemble of a very few carefully selected filters is similar to or better than ensembles of many filters. So, in this concept demonstration study, we will initially choose four filters; all these filters were described in Chapter 2.

The second step in the proposed ensemble is to aggregate the diverse outputs from different FS methods into a single result, which is a key component in a feature selection ensemble. Hence, choosing a suitable aggregation method is important. Aggregating the outputs of the different filtering procedures is not a simple task due to the different formats of the outputs produced by the feature selection methods. However, it can be achieved by ordering the features' score to generate a consensus feature ranking, or by counting the most frequently selected features in order to generate a consensus feature subset. In our work, the counting of the most frequently selected features will be used as the initial work in order to generate a consensus feature subset. Then, in the following chapters, we will use the rank list aggregation technique of mean aggregation, but with some changes to deal with the partial list. More detail with respect to the reasons for choosing this aggregation method will be presented in Section 6.4.

The question that arises after these two steps is: should we weight the filter members in an ensemble differently?

It is reasonable that the filters should be treated differently in accordance with their performance, as in reality, there are some differences in the performances of filters. Thus, the use of different weights for calculating the total scores of the selected features may improve the performance. Therefore, in order to answer this question, we will investigate how to determine the appropriate weight for each filter in an ensemble. To the best of my knowledge, this is the first study that gives weight to some filter methods based on their performance by using validation dataset. More details about the proposed methods are presented in Chapter 7.

3.3 Using Data in Feature Selection

In general, it is reasonable to assume that the quality of the selected features correlates with the number of samples available during training. So in order to increase the chance of selecting the most relevant features and then to build better models, we should use all the available data (Refaeilzadeh et al., 2007) for FS. However, using the entire dataset for FS before classification learning may produce over-optimistic results, as it has seen the test data in training. On the other hand, holding out one fold might exacerbate the "small sample" problem with FS, as many datasets have small numbers of samples, which may lead to underestimating the relevant features under some conditions.

This issue is important to be investigated, since it is a general issue in FS and needs to be answered. Consequently, we investigate this issue in Chapter 5 before continuing this research, and then we build the remaining studies based on the results in this chapter.

In Chapter 4, FS methods are applied to 10 real benchmark datasets using the entire datasets and then using the selected features as an input for the classifier (ALL method). In Chapter 5, FS methods are performed inside the cross-validation loop by executing the FS method on the training set before applying classifier construction in each iteration (PART method). The motivation for this study is to investigate whether PART or ALL is more appropriate as an evaluation method; to the best of our knowledge, the literature does not provide any clear answer as to which evaluation method (PART or ALL) is more appropriate, especially when using filters.

3.4 Evaluation Methods

In this thesis, we aim to have more accurate and reliable FS results than just stable FS results. The meanings of words stable and reliable are explained in the Oxford English Dictionary as follows:

stable : "*Not likely to change , strong or steady* " (Dictionaries, 2010), while, reliable : "*Consistently good in quality or performance; able to be trusted*" (Dictionaries, 2010).

As can be seen, there is a clear difference between their meanings.

Therefore, selecting stable FS does not always mean selecting important features; also, improving the stability of FS without having accurate results will be meaningless. However, we cannot measure the reliability of FS without measuring the stability by using a similarity measure, in conjunction with evaluating the effectiveness by the classification accuracy.

Therefore, in this research the methods of FS are evaluated in two ways: the first one is by estimating their reliability through using stability measures, independent of involving any classifier; and the second is by evaluating their effectiveness in terms of the classification accuracy of the classifiers that are generated using the features selected by a FS, which is dependent on the classifiers.

3.4.1 Stability Methods as an Indicator of the Reliability Measure of Feature Selection

The stability of FS was defined (Kalousis et al., 2007) as the robustness of the feature preferences it produces to differentiate in training sets drawn from the same generation distribution, which quantifies how different training sets affect the feature preferences. Also, it is defined (Han and Yu, 2012) as the insensitivity of the result of a feature selection algorithm to variations in the training set. The stability issue in feature selection has received much attention recently. As there is no single method that is the best for all domains and problems (Awada et al., 2012), in practice, high stability of feature selection is equally important as "high classification accuracy" (Jurman et al., 2008). While many feature selection algorithms have been proposed, they do not necessarily identify the same candidate feature subsets if we repeat the feature selection procedure with some variations (Yu et al., 2008). Even for the same data, one may find many different subsets of features (either from the same feature selection method or from different feature selection methods) that can achieve the same or similar predictive accuracy (Michiels et al., 2005). It is widely believed that a study that cannot be repeated has little value (Zhang et al., 2009). Consequently, the instability of feature selection results will reduce our confidence in selecting optimal features. However, an algorithm should not be selected based solely on the results of the stability assessment, although the stable results can be used to inform the researcher about the most appropriate feature selector, as long as the assessment is conducted in conjunction with

a classification algorithm. This will increase the level of confidence in the methodology and in the overall results, assuming that the feature selection is proven to be stable (Kalousis et al., 2007).

There are mainly three sources of instability in feature selection (He and Yu, 2010). Firstly, FS algorithm design without considering stability; secondly, the existence of multiple sets of potential true features in real data (Yu et al., 2008); and thirdly, a small number of samples in high-dimensional data (Loscalzo et al., 2009). Knowing the reason enables the researchers to better understand the problem. On the other hand, such knowledge will facilitate the design of new methods for stable feature selection.

Until now, for stable feature selection, many procedures have been available. Firstly, the ensemble feature selection method; secondly, the method that uses prior feature relevance that incorporates stability consideration into the algorithm design stage. Thirdly, the group feature selection approach treats feature cluster as the basic unit in the selection process to increase fortitude in order to handle data with highly equitable features. Fourthly, in order to increase the sample size to address the small-sample-size vs. large-feature-size issue, the sample injection method is implemented (He and Yu, 2010).

The stability measure can be used in different situations; it is necessary for evaluating different algorithms in performance comparison. Also, it can be used for internal validation in feature selection algorithms that take stability into account (He and Yu, 2010).

Measuring stability requires a similarity measure for FS results. There are three types of representation methods: subset of features, ranking vector and weighting score vector. In this work, we focus on a subset of features because our filter-based ensemble algorithm produces subsets of features. There are quite a few similarity measures available for the comparison of sets, as reviewed by He and Yu (2010). We follow the categorisation presented by Somol and Novovičová (2010):

1- Feature-focused versus subset-focused measures: feature-focused measures evaluate feature selection frequencies over all feature subsets considered together as a whole, as with Somol and Novovičová (2008), while subset-focused measures evaluate similarities within every pair of selected feature subsets, as with (Kalousis et al., 2007,

Kuncheva, 2007). Both kinds offer complementary information; consequently, we want to have at least one of each in our investigation.

2- Subset-size biased versus subset-size unbiased measures: The former measures yield values bounded more tightly than [0, 1], with most notably the lower bound strongly increasing with the proportion of selected features, while the latter measures are adjusted to be actually bounded by [0, 1]. For better generalisation we want to use subset-size unbiased measures.

In the following section, the similarity and stability measures used in our investigation are defined:

Relative Weighted Consistency (CWrel) is defined by correcting Weighted Consistency (CW) to be actually bounded by [0, 1] regardless of the proportion of the selected features. A value of 0 indicates the highest possible instability, while a value of 1 indicates the highest possible stability.

$$\text{CW}(X_s) = \sum_{x \in X} \frac{q_x}{N_s} \cdot \frac{q_x - 1}{k - 1} \tag{3.1}$$

$$CW_{rel}(X_s, \text{X}) = \frac{CW(X_s) - CW(N_s, k, X)}{CW_{max}(N_s, k) - CW_{min}(N_s, k, X)} \tag{3.2}$$

Let X= $\{x_{1,\ldots\ldots,}x_N\}$ be the set of all features of size N, X_s= $\{X_{s_{1,\ldots\ldots,}}\ X_{s_n}\}$ is a set of k subsets of features obtained from k folds, where $X_s \subset X$. N_s is the total number of occurrences of any feature in X_s and q_x is frequency of feature x in X_s. Among the stability measures reviewed in He and Yu (2010), the relative weighted consistency CWrel (Somol and Novovicova, 2010) is the only one that has both feature-focused and subset-size unbiased measures, so we selected it to be one of the measures used in our research. In order to complement it, we need to use other measures with a focus on subsets. Křížek et al. (2007) and Kuncheva (2007) are both subset-focused, but they can only be used on subsets of equal cardinality. However, in our research, the subset cardinality was not equal, so we used the Average Tanimoto Index (ATI).

Average Tanimoto Index (ATI) is computed over all subset pairs, and then averaged (Somol and Novovicova, 2010). It is a continuous value from [0, 1], with 0 representing empty intersection between subsets X_i , X_j and 1 representing that all subsets obtained from *k* folds are identical:

$$\text{ATI}(X_s) = \frac{2}{k(k-1)} \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} S_K\ \left(X_i, X_j\right) \tag{3.3}$$

ATI is based on Kalousis' similarity measures S_K between two sets X_i , X_j, where X_i , $X_j \in X$ (Kalousis et al., 2007):

$$S_K\ (X_i, X_j) = \frac{|X_i \cap X_j|}{|X_i \cup X_j|} \tag{3.4}$$

ATI evaluates pair-wise similarities between subsets in the method, while CW_{rel} evaluates the overall occurrence of features in the method as a whole. Assessing the stability of an FS process based on only a single measure may lead to a misleading conclusion. For example, if ATI produces a very low value it may not necessarily mean that one will be unsuccessful in identifying important features; it may produce different combinations of redundant features in each subset. Therefore, no single measure is capable of expressing all the information that can be useful in assessing the stability of an FS process. It is recommended to consider evaluating a set of measures of different types (feature-focused and subset-focused as well as a subset unbiased one) to gain rational information on the evaluated FS process (Somol and Novovicova, 2010).

All the measures discussed above consider intra-measures, which are used for evaluating the internal stability of one FS process, as in the PART method (Section 5.2). We cannot use it for the ALL method (Section 5.2) because the entire dataset is used and there is no change in the dataset during each run. Also, with these measures, we cannot compare the subset produced from each FS with the optimal answer (relevant features), because we do not know the optimal answer when using the real-world dataset, so for this reason, we generate the synthetic dataset in Chapter 5. Therefore, we include more measures in our investigation, called inter-measures, in order to compare the result of each method (ALL, PART in Section 5.2) with the relevant features on synthetic data. The inter-measures should provide complementary information to the intra-measures. Therefore, each of the following inter-measures is defined as an equivalence to some intra-measures, based on the same or related principle (Somol and Novovicova, 2010).

The Inter-method Weighted Consistency ICW($X_s{}^1$,$X_s{}^2$) between the results of two methods $X_s{}^1$ and $X_s{}^2$, where k_1 is number of folds in $X_s{}^1$ and k_2 is number of folds in $X_s{}^2$. ICW($X_s{}^1$,$X_s{}^2$) takes values from [0,1], with 0 representing that no feature appears in more than one method and 1 representing that the relative frequencies are equal for

each feature in both results of two methods (Somol and Novovicova, 2010). It is defined as: $\mathrm{ICW}(X_s^1, X_s^2) = 1 - \sum_{x \in X} w_x \left| \frac{q_x^1}{k_1} - \frac{q_x^2}{k_2} \right|$ (3.5)

$$where, w_x = \frac{\max(\frac{q_x^1}{k_1}, \frac{q_x^2}{k_2})}{\sum_{g \in X} \max(\frac{q_g^1}{k_1}, \frac{q_g^2}{k_2})}.$$

The Inter-method Average Tanimoto Index (IATI_R) between the results of two methods X_s^1 and X_s^2 takes values from [0,1] with 0 indicating empty intersection between any pair of subsets, and 1 indicating that all subsets in the results of both methods $X_s^1$and $X_s^2$are identical (Somol and Novovicova, 2010). The original $\mathrm{IATI}_R$is defined as:

$$\mathrm{IATI}_R(X_s^1, X_s^2) = \frac{1}{k_1 \cdot k_2 \cdot |X|} \sum_{i=1}^{k_1} \sum_{j=1}^{k_2} \frac{|X_i^1 \cap X_j^2|}{|X_i^1 \cup X_j^2|}. \quad (3.6)$$

However, we found that this definition is highly affected by the size of *X*, which leads to decreasing the similarities when the number of features was increased. Therefore, we modify it by removing |*X*| to avoid this drawback. It is now defined as follows:

$$\mathrm{IATI}(X_s^1, X_s^2) = \frac{1}{k_1 \cdot k_2} \sum_{i=1}^{k_1} \sum_{j=1}^{k_2} \frac{|X_i^1 \cap X_j^2|}{|X_i^1 \cup X_j^2|}. \quad (3.7)$$

3.4.2 Classification Performance as Effectiveness Measure of Feature Selection

In supervised classification, a sample (S) is defined as an instance of a problem represented by a set of predictive features (x) and a class variable (y) which represents where the class sample belongs to. When we store a set of samples with the same format, we have a dataset (D). If y is Numeric, then the supervised classification task is known as Regression. In this thesis, the class feature is assumed to be Nominal, having a predefined and finite number of possible labels.

There are a large variety of metrics to measure the performance of the classifier; also, there are a number of methods to compute these metrics to avoid over-fitting conclusions and to evaluate the model. In this section, the most relevant validation methods and the most frequently used metrics are presented; in addition to describing the most common algorithms for supervised classification.

3.4.2.1 Validation Techniques

A training set is a set of samples from which the classifier is built and it must not be used to evaluate the goodness of the learned classifier. A test set, which is another set of labelled samples, is needed for evaluation purposes. The classifier will be run to predict a label for each sample in the test set then compared with the real label of such samples. A number of metrics can be computed from these results and there are several methods to construct the training and test sets; also there are several evaluation techniques. These are detailed below:

1) Percentage Split

This is the straightforward evaluation technique that just divides the dataset into two sets: the first set is used in the building process as a training set, and the rest for testing.

2) k-fold Cross-Validation

This is the most commonly used technique that is formed by randomly splitting the dataset D into k disjoint splits (folds) of the same size. Then, a process is run k times and one dissimilar fold is used as a test set. Finally, the performance is measured as the mean of the computed k scores. Commonly, folds are constructed in a stratified manner. This means that each fold keeps the distribution of the class variable from the whole dataset D (Tan et al., 2006). Ambroise and McLachlan (2002) recommend using 10-fold rather than leave-one-out cross-validation, because the latter can be highly variable. Thus, we will be using 10-fold Cross-Validation as evaluation criteria in our thesis.

3) Leave-one-out

This kind of evaluation technique is a special case of the k-fold Cross-Validation which occurs when k = |S|, where |S| represents the total number of samples. In this case, each sample serves as its own test set. It is commonly used for small datasets and provides as many training instances as possible to the classifier, but this method leads to a very computationally expensive evaluation.

3.4.2.2 Classification Performance Measures

Classification performance measures are fundamental in assessing the quality of classifier and classification models. In classification of binary (two classes) problems, there are four possible outcomes in terms of classifier prediction: true positive (T_P), false positive (F_P), true negative (T_N), and false negative (F_N). These four values form the basis for several other performance measures that are well-known and commonly used within the data mining and machine learning community. Table 3.1 displays a confusion matrix for a two-class classification problem (Tan et al., 2006).

Table 3. 1: Confusion matrix for a two-class prediction problem

		Prediction	
		Yes	No
Actual	Yes	T_P (true positive)	F_N (false negative)
	No	F_P (False positive)	T_N (true negative)

A performance matrix, such as accuracy and error rate, are defined below, and provide a useful measure of performance.

1) $$\boldsymbol{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} = \frac{T_P+T_N}{N_p} \quad (3.8)$$

Equivalently, the performance of a classifier can be expressed in terms of its error rate, which is given by following equation:

2) $$\boldsymbol{Error\ rate} = \frac{\text{Number of wrong predictions}}{\text{Total number of predictions}} = \frac{F_P+F_N}{N_p} \quad (3.9)$$

Accuracy can be understood as the mean of precisions for all possible class labels without weighting by the number of available instances for each label.

There are also a number of classification performance measures such as precision, recall, F-measure and AUC (Tan et al., 2006). The specificity (precision) and sensitivity (recall) are statistical measures of the performance of the binary classification test, which primarily looks at one class (a specific class such as cancer gene). However, most of the datasets we use in this thesis are multi-class, so in our case, we do not have a

specific class to target. Hence, we do not use specificity and sensitivity. Also, in this study, we do not focus on classification as a main topic; we just use classifiers as an evaluation method to compare the performance of our EFS with other FS methods. So we chose accuracy as the classification performance measure, because it is simple to calculate, and most studies on machine learning use it.

3.4.2.3 Statistical Tests for Comparison

Many studies adapt various statistical techniques to decide whether the differences between the algorithms are real or random. The selection of the test should be based on statistical appropriateness and also on what we intend to measure. So, there are essential differences between the test used to assess the difference between two algorithms on a single dataset, such as the t-test, and the differences over multiple datasets such as the Friedman test.

In this section, we will present the statistical tests used in our research:

1) The paired t-test is a frequently used technique to test whether the difference between two algorithms over different datasets in non-random manner. It verifies whether the average difference in their performance over the datasets is significantly different from zero. We used the paired t-test in Chapters 4 and 5 in order to compare the classification performances using all the datasets without using FS, and the classification performance using individual FS and HEF.

2) The Friedman test (Demšar, 2006) is a non-parametric test that ranks the algorithms for each dataset independently. The best performing algorithm receives the rank of 1, the second best is ranked 2 ... and so on. In the case of ties, average ranks are assigned. Then, if the null hypothesis is rejected, the Nemenyi test can proceed. It is used when all the algorithms are compared to each other using multiple testing datasets. The performances of two algorithms are significantly different if the corresponding average ranks differ by at least the critical difference:

$$CD=q_\alpha \sqrt{\frac{A(A+1)}{6D}} \qquad (3.10)$$

Where, A is the number of algorithms, D is the number of datasets used and q_α is the critical value; these are based on the Studentized range statistics divided by

$\sqrt{2}$ (Demšar, 2006). We used the Nemenyi test in Chapters 6 and 7 in order to compare different ensemble results.

3.4.2.4 Algorithms for Classification

There are many classifiers of different natures which can be used for supervised classification. Generally, the classification performance may be dependent on the types of classifiers used, under exactly the same conditions, subset of features, number of samples, and training procedure. To verify the consistency of the feature selection methods, we have used three types of classifiers in our experiments: NB (Naïve Bayesian) (John and Langley, 1995), KNN (*k*-Nearest Neighbours) (Aha et al., 1991) and SVM (Support Vector Machine) (Platt, 1999). These three algorithms have been chosen because they represent three quite different approaches in machine learning, and they do not contain any embedded feature selection mechanisms; also, they are state-of-the-art algorithms that are commonly used in data mining practice.

1) Naïve Bayesian (NB) (John and Langley, 1995): is a classifier based on Bayes' rule of conditional probability. It is simple, very efficient and able to outperform other more advanced and sophisticated algorithms. It is based on the assumption of conditional independence among the predictor features. While this assumption is highly unlikely in real-world data, research has shown that NB often performs well on datasets with highly correlated features.

2) K-Nearest Neighbours (KNN) (Aha et al., 1991): is a simple classifier algorithm and it belongs to the category of instance-based learners which is also called lazy learner. Since the actual generalisation process is delayed until classification is performed, there is no model building process. KNN is based on the principle that samples within a dataset will generally exist in close proximity to other samples that have similar properties. So, it classifies each new instance by a majority vote of their neighbours and it is assigned to the class most commonly among its k-nearest neighbours, where *k* is an odd number to avoid duplicate counts.

3) Support Vector Machine (SVM) (Platt, 1999): builds a linear discriminate function using a small number of critical boundary samples from each class, while making sure of maximum possible separation. There may be several kernels for separating classes,

but the best kernels are those which maximise the distances between the nearest instances of such groups. In (Boser et al., 1992), several new kernels are presented so that SVMs can be used as non-linear classifiers.

In this research, the experiments are carried out in two phases: the feature selection phase and the evaluation phase. The first phase is to run the proposed ensemble in order to produce a subset of ranked features as well the subsets selected by each individual filter. The second phase is to evaluate the effectiveness of the selected features with three kinds of models (NB, KNN and SVM), and also to evaluate the stability of the selected features by using two measures as described in the previous section.

3.5 Comparison Strategies

We compare the results from this study using two strategies. Firstly, comparing our ensemble results to individual results, and secondly, comparing our ensemble results to other ensemble results, in terms of accuracy, stability and the number of features selected.

The first strategy is to compare our ensemble results to individual FS results including either our own filter members in the ensemble or other FSs used in the literature. In each chapter of this study, we regularly compare the results obtained from our ensemble with the results obtained from each filter member separately. Furthermore, in Appendix C, we compare our ensemble results with different FS methods used in the literature if they used the same datasets (Table C.1).

The second strategy is to compare our ensemble results to other ensemble results, either our own previous ensemble versions or other ensemble studies in the literature. In more detail, in Chapter 5, we compare the ensemble results from Chapter 5 with the ensemble results from Chapter 4. Also, in Chapters 6 and 7, we compare the different versions within each chapter. Moreover, in the discussion chapter, we compare our ensemble results with different ensemble studies in the literature if they used the same datasets.

3.6 System Software Design

The proposed ensemble framework is implemented in Java and uses the modules available in WEKA (Waikato Environment for Knowledge Analysis) (developer version 3.7.8) and other standalone filter software.

3.6.1 WEKA

WEKA is a collection of machine learning algorithms for data mining tasks, and the algorithms can be directly applied to datasets through Java code. The input files to WEKA are datasets that are in the ARFF format but there are two primary modes to consider: Explorer and Experimenter. The first is a data preparation stage that is designed to assist the researcher with gaining a clear overview, as well as an in-depth understanding of the data; it entails the use of WEKA's data pre-processing, learning, attribute selection and data visualisation modules. The other mode facilitates the implementation of experiments, and allows the researcher to store the results in a database that may be accessed and exploited as the researcher wishes (for further analysis, etc.) (Witten and Frank, 2005).

3.6.2 Java Code

As our work (ensemble algorithm) is not part of the standard capability of the WEKA toolset, we implement it within the Java environment. In addition, the evaluation stage, which entails a stability measure and classification performance, is also implemented in Java.

The inputs to our algorithm are l filters and Q heuristic consensus rule (details in Chapter 4). The variables l and Q can be altered. The procedure starts by running F_1, F_2,....., F_l; after this step, our algorithm selects different aggregation methods (details in Chapters 4 and 6).

3.7 Experiment Design

3.7.1 Data

Ten benchmark datasets from different domains are used in our experiments to test the performances of our proposed ensemble of feature selection. Six of them, Zoo, Dermatology, Promoters, Splice, Multi-feature-factors and Arrhythmia, are from the UCI Machine Learning Repository,[1] two others, Colon and Leukaemia, are from the Bioinformatics Research Group[2], and the final two, SRBCT and Ovarian, are from the Microarray Datasets website.[3]

Table 3.2 summarises the general information on these datasets. Note that these datasets differ greatly in sample size (ranging from 62 to 3,191) and number of features (ranging from 17 to 15,154). Also, they include binary-class and multi-class classification problems; this should provide a basis for testing and should be well-suited to the feature selection methods under differing conditions.

Table 3.2: Description of the benchmark datasets

No.	Dataset	No. of Samples (S)	No. of Classes (y)	No. of Features		
				Total (N)	Categorical	Numeric
1	Zoo	101	7	17	17	0
2	Dermatology	366	6	34	33	1
3	Promoters	106	2	57	57	0
4	Splice	3,191	3	61	61	0
5	M-feat-factors	2,000	10	216	0	216
6	Arrhythmia	452	13	279	73	206
7	Colon	62	2	2,000	0	2,000
8	SRBCT	83	4	2,308	0	2,308
9	Leukaemia	72	2	7,129	0	7,129
10	Ovarian	253	2	15,154	0	15,154

[1] http://repository.seasr.org/Datasets/UCI/arff/

[2] http://www.upo.es/eps/aguilar/datasets.html

[3] http://csse.szu.edu.cn/staff/zhuzx/Datasets.html

3.7.2 Experiment Procedure

For each dataset, the experiments are carried out in two phases: the feature selection phase and the evaluation phase. The first phase is to run the proposed ensemble in order to produce a subset of ranked features as well the subsets selected by each individual filter. The second phase is to evaluate the effectiveness of the selected features with three kinds of models: NB, KNN and SVM. In addition, the stability of the algorithms is measured with two kinds of measures: ATI and CWrel.

In order to increase the statistical significance of the results, as well as to achieve impartial results, the average values over 10 fold cross validation are used. Each experiment is then repeated 10 times, with different data partitions shuffled with different random seeds in order to assess the consistency of the results.

The statistical significance of the results of multiple runs for each experiment is calculated, and the comparison between accuracies is performed with "Student's two-tailed *t*-test", with a significance level of 0.05 (in Chapters 4 and 5) in order to compare the classification performance using all datasets without using FS, and the classification performance using each FS and EFS. After that, (in Chapters 6 and 7) the comparisons between different results were tested with the Friedman test with a significance level of 0.05 (Demšar, 2006).

Moreover, in addition to accuracy, we will measure the stability of FS, as in each fold the FS method may produce different feature subsets, and in order to identify the factors that play the most important roles. Measuring stability requires a similarity measure for the FS results. In this work, we focus on subsets of features because our filter-based ensemble algorithm produces subsets of features. The stability measures used in our investigation are: Relative Weighted Consistency (CWrel) and Average Tanimoto Index (ATI) (Somol and Novovicova, 2010), as the subset cardinality is not equal in our research. ATI evaluates pair-wise similarities between subsets in the system (10 folds), while CW_{rel} evaluates the overall occurrences of the features in the system (10 folds) as a whole. CW_{rel} and ATI may produce different results in each run, so the average of 10 runs will be used. Also, we included more measures in Chapter 5, called inter-measures, in order to compare the features selected from the PART method (in each fold) with the ALL method. The Intersystem Weighted Consistency IWC and the Intersystem Average

Tanimoto Index (IATI) which is provided in (Somol and Novovicova, 2010), are used in this investigation.

Chapter 4

Heuristic Ensemble of Filters

4.1 Introduction

In the previous chapter we have provided the methodology of this research. The review on the previous studies in the area of FSE found that the majority of these studies were predominantly limited to using one filter with instance level perturbation or using different types of rank filters, as the member components of an ensemble, which produces a ranking of features. Moreover, some additional work needs to be performed to decide a cutting off point to produce a subset of selected features. In this chapter, we will present the proposed heuristic methods which consist of two parts: the heuristic cut-off rule and the heuristic consensus rules. The heuristic cut-off rule will apply before combining the results of filters (SF and RF) by choosing the highest number of features selected by the SF to cut off the top-ranking features for the remaining ranking filters. The heuristic consensus rules will apply after combining the results of the filters by removing any features selecting by only a few filters, in order to reduce the number of feature selected and to obtain the more important features. Our algorithm is implemented and tested on various benchmark datasets and the results are promising. The work in this chapter has been published in the International Conference on Pattern Recognition Applications and Methods in 2014.

The remainder of the chapter is organised as follows: Section 4.2 introduces the methodology of this research. Section 4.3 describes the experiments, and Section 4.4 details the results of the experiments conducted on 10 datasets in order to evaluate the performance of our approach. The conclusions are presented in Section 4.5.

4.2 Heuristic Ensemble of Filters (HEF)

4.2.1 Proposed Heuristic Ensemble of Filters (HEF)

The proposed heuristic ensemble of filters (HEF) is composed of two types of filters – subset filters (SFs) and rank filters (RFs) – as its members, counting the frequency of selected features as its consensus function, as shown in Fig. 4.1. The idea of combining SF and RF is to exploit the advantages of each. Firstly, an RF usually assesses individual features and assigns them weights according to their degree of relevance, but this does not ensure conditional independency among the features, and may lead to

selecting features that are redundant or have less discriminative ability. An SF takes into account the existence and effect of redundant features, which to some extent approximates the optimal subset. However, this method entails high computational cost in terms of the subset searches, making SF inefficient for high-dimensional data, although it is much better than wrapper. As a result, to obtain the benefits of SF without suffering the high computational cost, we choose very fast SF by modifying their search strategy to make it much quicker, as described in Section 4.2.2. Secondly, by running the chosen SF, we can obtain quick answers for cutting off the number of features in the ranker.

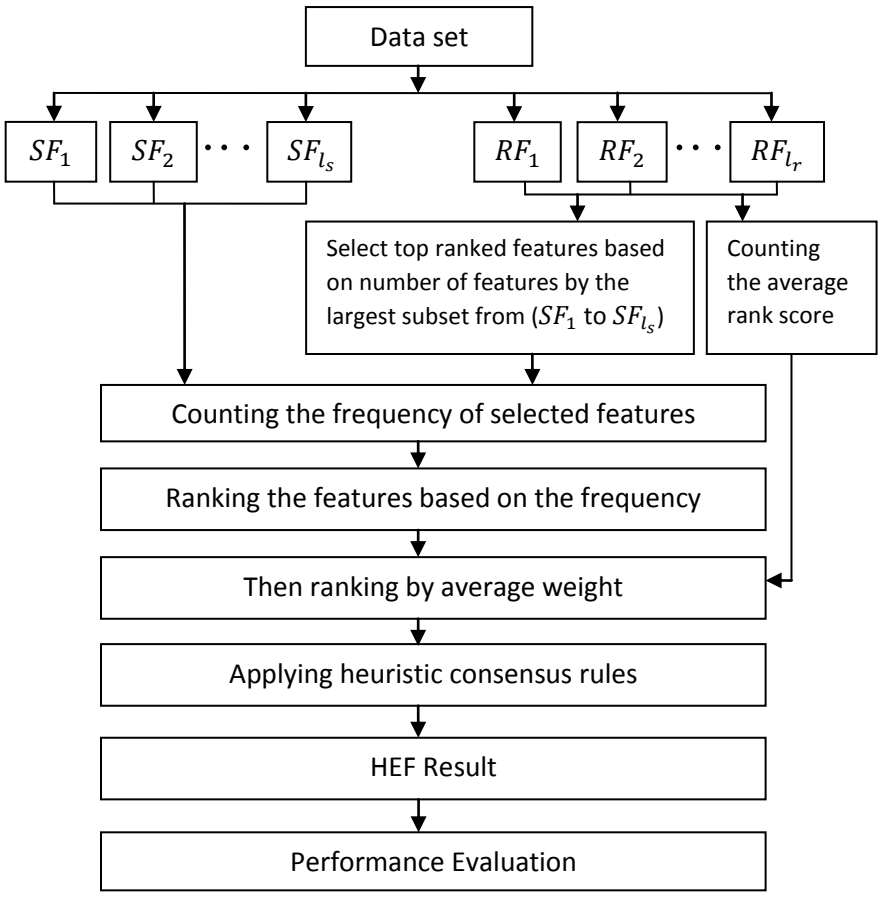


Figure 4.1: Framework of HEF for feature selection

The process of heuristic ensemble of filters is shown in algorithm 4.1. It firstly runs SF in this experiment then RF. After that, the highest number of features selected by the SF is taken as a cut-off point for the rankings generated by the RF. The next step (line 5) aggregates the results from the above sets by counting the frequency of the selected features, and then ranking these features based on their frequency (outer rank). However, as the probability of any two features having the same frequency is high, and

to resolve the issue of frequency collision (and to take advantage of RF by knowing the most important features), we introduce a mean ordering strategy derived from RF (line7); each feature's score is determined by the average ranking score in all the ranking lists. The sorting is performed in increasing order (inner ranking). The intention of adding the inner ranker, which uses the score of each feature in the RFs to rank the features in the outer rank, is to assist with identifying those important features that have equal frequency. Also, it may help to determine the important features in the wrapper stage (Section 6.2). Moreover, to obtain more important features, a heuristic consensus rule is applied (line 8) to produce the final output of the HEF. Different heuristic rules can be derived based on the purpose of the analysis.

Input
l_s // number of SF
l_r // number of RF
Q // heuristic consensus rule
1. begin
2. Run subset filters SF_1, SF_2... SF_{l_s}.
3. Run ranking filters RF_1, RF_1... RF_{l_r}
4. Select top ranked features from (RF_1, RF_1... RF_{l_r}) based on the highest number of selected feature subset by (SF_1, SF_2... SF_{l_s}).
5. Count the frequency of the selected features.
6. Rank the features based on the frequency (outer rank).
7. Rank the features based on the mean weight RFs (inner rank).
8. Apply heuristic consensus rule, Q.
9. Remove features based on Q.
10. end
Output: A final subset of ranked features

Algorithm 4.1: HEF Algorithm

4.2.2 Choice of Individual Filters

In principle, any filters of each type can be used as the member filters of our HEF. However, some factors should be considered when choosing the filters, which include efficiency, stability and scalability. In terms of determining the number of member

filters, we followed the guidelines given in Wang et al. (2012), that is, an ensemble of a very few carefully selected filters is similar to or better than ensembles of many filters. However, filters are designed with different evaluation criteria which may work well with some datasets but possibly not with other datasets. Therefore, it is clear that each filter uses a different criterion to evaluate the relevance of the candidate features in the datasets. When combined, candidate features are assessed from many different aspects. So, in order to improve HEF to select more reliable feature selection, we categorised these evaluation criteria into groups broadly based on the following studies (Saeys et al., 2007, Liu and Yu, 2005): distance, information, dependency and consistency. Then, we studied the popular filters under each of these categories in order to be able to choose the appropriate filters from each category. Each category of evaluation is described here briefly to give an idea of why they are selected in this study:

1) **Distance:** This criterion tries to find features that can separate the class labels of the dataset as much as possible and is measured by a metric unit (such as Euclidean distance). For example, in a two-class problem, a feature x_1 is preferred over x_2, if x_1 generates a greater difference (distance) between the two classes of conditional probabilities than x_2 (Liu and Yu, 2005). The Relief filter family is a famous and important filter in this type of evaluation. Relief was proposed by Kira and Rendell in 1992 (Kira and Rendell, 1992), then ReliefF (Kononenko, 1994) was extended by Kononenko, so that it can deal with multi-class problems, and noisy and missing values. Thus, we chose ReliefF as the RF in our HEF.
2) **Information:** This criterion determines the information gain from a particular feature by using an entropy measure. It prefers the feature with a high information gain. There are a number of filters under this type of evaluation: Information gain (IG), Gain ratio (GR), symmetrical uncertainty (SU) and Conditional Mutual Information Maximisation (CMIM). IG prefers to select features having a large number of possible values over features with fewer values, even though the latter are more informative (Karegowda et al., 2010). Due to this bias, we did not choose this filter and we selected GR instead, which overcomes this weakness. Also, CMIM selects a feature subset that carries maximum relevance to the target class by using conditional mutual information (Fleuret, 2004). However, CMIM requires that both the feature values and output classes have to be binary. Due to this limitation, we discarded it as well (Yun and Yang, 2007).

3) **Dependency:** Dependency or correlation measures estimate the correlation between features and classes, which means a good feature subset is one that contains features highly correlated to the class, yet uncorrelated to each other (Liu and Yu, 2005). The most famous and important filter in this type of evaluation is Correlation-based Feature Selection (CFS) proposed by (Hall, 2000). It is useful for identifying and discarding features which can often be redundant and irrelevant to the target variable (Chrysostomou, 2008). In order to avoid a high computational cost of CFS as SF, we used linear forward selection (LFS) as a search method together with CFS, instead of using best-first search. LFS is a simple complexity optimisation of sequential forward selection (SFS) (Gutlein et al., 2009). Also, the Fast Correlation-Based Filter (FCBF) is a fast filtering algorithm that ranks features by sorting them through correlation with a response using symmetric uncertainty. It uses both Classes-correlations and Features-correlations to determine feature redundancy and combines sequential forward selection with elimination. We chose CFS and FCBF as SF filter members in our HEF.
4) **Consistency:** This is different from the above measures because it relies on class information feature bias when selecting the subset. Consistency measures attempt to discover the smallest amount of features that separate classes as consistently as the original set of features (Liu and Yu, 2005). FOCUS (Almuallim and Dietterich, 1991) is a famous filter in this type of evaluation. However, there are two main problems with FOCUS, as pointed out by Caruanna and Freitag (1994). Firstly, FOCUS uses an exhaustive search which is intractable if many features are needed to achieve consistency. Secondly, it can be statistically unwarranted to have a strong bias towards consistency; such a scenario might just lead to over-fitting for the training data. In order to repair even a single inconsistency, the algorithm will keep adding features. Liu and Setiono (1996b) describe an algorithm similar to FOCUS called the Las Vegas Filter (LVF) but it can handle noisy domains if the approximate noise level is known a-priori. LVF randomly searches the space of subsets using a Las Vegas algorithm (Brassard and Bratley, 1996). However, these two filters are slow and consume more time compared to other filters selected as a member of HEF. So we did not select any filter from this category in order to increase the runtime performance of HEF.

In summary, in this concept demonstration study, we chose four filters in total – two rank filters, namely ReliefF (Robnik-Šikonja and Kononenko, 2003) and Gain Ratio (Quinlan, 1993) and two subset filters, namely Correlation-based Feature Selection

(CSF) (Hall, 1999) and Fast Correlation-Based Filter (FCBF) (Yu and Liu, 2004). All of these filters have been explained in detail in Chapter 2. It should be noted that the idea of combining SF and RF has not been used in other studies on feature selection ensembles.

4.2.3 The Heuristic Rules

As we mentioned in the introduction of this chapter, our heuristic methods consist of two parts: heuristic cut-off rule and heuristic consensus rules. The heuristic cut-off rule was run before the aggregation step by choosing the highest number of features selected by the SF (FCBF or CSF) to cut off the top-ranking features for the remaining ranking filters (ReliefF and Gain Ratio). By running this heuristic step, we can obtain quick answers for cutting off the number of features in the ranking, which will accelerate the ensemble algorithm. Therefore, we will not need to select various feature numbers to test the performance, or to use a wrapper to choose the appropriate number of features.

Aggregating the outputs of the different feature selection procedures can be achieved by averaging the score of each feature in order to generate a consensus feature ranking, or by simply counting the most frequently selected features in order to generate a consensus feature subset (Saeys et al., 2008). In this chapter, we focus on ensemble feature selection (EFS) techniques that work by aggregating the feature subsets provided by the different filters into a final consensus subset. The most frequently selected features are placed at the top, while the least frequently selected features are placed at the bottom. Then we rank-based on a mean ordering derived from RF; each feature's score is determined by the average ranking score in all the ranking lists. The sorting is performed in increasing order (inner ranking). One issue with integrating multiple scores is that different filtering algorithms often provide evaluation scores with different scales. In order to combine the evaluation results of multiple filters, it must transform the evaluation scores into a common scale. Therefore, the softmax scaling (Yang et al., 2010) process is adopted to squash the feature evaluation results of each filtering algorithm into the range of [0-1].

The second heuristic method is heuristic consensus rules, which are run after the aggregation step. This step is required after aggregate the outputs by counting the most frequently selected features may produce a high number of selected features, including

features with low frequency levels selected by only a couple of filters (or even a single one). In order to address this issue and also to obtain more important features, a heuristic consensus rule is applied to produce the final output of the HEF. Various heuristic rules can be derived based on the purpose of the analysis; in the following some simple rules are defined just to demonstrate the concept.

HEF-R=
- R0 ⟶ remove nothing from HEF (no rule)
- R1 ⟶ remove features selected by only one filter
- R2 ⟶ remove features selected by only two filters
- ⋮
- Rg ⟶ remove features selected by g filters (g = *l*-1)

Where *l* is the number of filter members in the HEF, and *g* is less than the number of filter members in the HEF by one. The first heuristic ensemble of filters, named HEF-R0, has all the features selected by RF and SF, whereas HEF-R1 is the heuristic ensemble of filters after removing any features selected by only one filter, and so on. In this experiment, HEF-R0 (simply called HEF) and HEF-R1 are used.

Other heuristic consensus rules which remove any features selected by *g* filters or less were tested in the pilot study; however the accuracy of these heuristic rules were worse than those of HEF-R1, HEF. Accordingly we did not include them in the thesis.

4.3 Experiments

4.3.1 Data

Ten benchmark datasets from different domains are used in our experiments to test the performance of our proposed heuristic ensemble of filters. Six of them, Zoo, Dermatology, Promoters, Splice, Multi-feature-factors and Arrhythmia, are from the UCI Machine Learning Repository; two others (Colon and Leukaemia) are from the Bioinformatics Research Group, and the final tow (SRBCT and Ovarian) are from the Microarray Datasets website. Table 3.2 in Section 3.7.1 summarises the general information on these datasets.

4.3.2 Experiment Design and Procedure

As it is generally accepted that the effectiveness of feature selection can be indirectly evaluated by measuring the classification accuracy of classifiers that are trained with the selected features, we thus conducted several series of experiments with a variety of datasets to empirically evaluate the accuracy of the HEFs. We compared them with each individual filter used in this study, and also the full feature set without any feature selection performed.

As mentioned earlier, the classification accuracy may be dependent on types of classifiers used even under exactly the same conditions, subset of features and samples, and training procedure. To verify the consistency of the feature selection methods in our experiments, we used three types of classifiers: Naïve Bayesian Classifier (NB) (John and Langley, 1995), K-nearest neighbour (KNN) (Aha et al., 1991) and Support Vector Machine (SVM) (Platt, 1999). These three algorithms were chosen because they represent three quite different approaches in machine learning and they are commonly used in data mining practice.

For each dataset, the experiments are carried out in two phases: the feature selection phase and the evaluation phase. The first phase runs HEF to produce a subset of ranked features, as well the subsets selected by each of the individual filters. The second phase is to evaluate the effectiveness of the selected features with three kinds of models – NB, KNN and SVM. Specifically, it firstly trains the model of each type with the full set of features and the subsets produced by FCBF, CFS, ReliefF, Gain Ratio, HEF and HEF-R1, using the 10-fold cross-validation strategy for each classifier. Each experiment is then repeated 10 times with different shuffling random seeds in order to assess the consistency of the results. In total, 7 (All+4FS + 2ensemble) × 10 (datasets) × 3 (classifiers) × 10 (runs) × 10 (folds) = 21,000 models that were built for the experiments. The statistical significance of the results of multiple runs for each experiment is calculated and the comparison between accuracies is calculated with Student's paired two-tailed *t*-test with a significance level of 0.05.

4.4 Results

4.4.1 Number of Selected Features

Table 4.1 lists the number of features selected by each filter in addition to two heuristic ensembles: HEF and HEF-R1. We observe from the table that the average number of selected features dramatically reduced the dimensionality of the data by selecting only a small proportion of the original features in those datasets. Although HEF has the total number of features selected from all the four filters, it is still less than the average full set by up to 50 times for genetic datasets.

Table 4. 1: Number of selected features for each dataset by the four filters and two ensembles

Dataset	All features	FCBC	CFS	ReliefF	Gain Raito	HEF	HEF-R1
Zoo	17	7	10	10	10	11	11
Dermatology	34	16	19	19	19	28	24
Promoters	57	6	6	6	6	7	6
Splice	61	22	22	22	22	29	25
M-feat-factor	216	38	47	47	47	82	62
Arrhythmia	279	12	21	21	21	52	17
Colon	2,000	14	23	23	23	50	21
SRBCT	2,308	82	77	82	82	177	92
Leukaemia	7,129	51	52	52	52	111	58
Ovarian	15,154	30	36	36	36	76	43
Average	2,725.5	27.8	31.3	31.8	31.8	62.3	35.9
St. Dv.	4,829.8	22.59	20.76	21.88	21.88	49.33	25.92

4.4.2 Accuracy Evaluation

Tables 4.2 – 4.4 show the average accuracy of NB, KNN and SVM models on the 10 datasets; each value presented in the tables is the average over 10 runs of 10-fold cross-validation outcomes. For each classifier, the accuracies of classification on the datasets with all the original features are given in the "All features" column as a comparison. The notations '+' or '-' denote that the result of the classification of the models trained with the features selected with the current selector is significantly better or worse than that of models trained with all the original features in the statistical test mentioned earlier (t-test). The bold value in each row shows the best classification result. The last three rows in each table show Average (the average accuracies), St. Dv. (the standard deviations for the accuracies) and W/T/L (which summarises the wins/ties/losses in accuracy by comparing the models trained with all the features and the features selected by the four filters, and two heuristic ensembles: HEF and HEF-R1 over all the datasets). It should be noted in comparison that when we state that filter A is better or worse than filter B for simplicity, it means that the models trained with the features selected by filter A are better or worse than the models trained with the features selected by filter B, under the same experimental set-ups.

Table 4.2 shows the results on the 10 datasets with the Naïve Bayesian Classifier and the accuracy comparison between the NB classifiers trained with all the features and the features selected by four individual filters and two ensembles. As expected, each single filter performed well in some datasets (in bold) but poorly in others. That confirms the perception that the performance of individual filters is inconsistent, and no obvious or meaningful pattern can be extracted to indicate when they do better and when they do not. Nevertheless, The NB classifiers trained with the features selected by HEF-R1 have a higher average accuracy for all the datasets and a lower standard deviation, which indicates that HEF-R1s are more accurate than the individual filters in feature selection. In addition, HEF-R1 achieves the highest accuracy on three datasets. Comparing the results for this classifier using the full feature set with others, it can be observed that in most cases, the accuracy is increased in HEF-R1, HEF, CSF and FCBC, while in the RF (ReliefF and Gain Ratio), the accuracy is poorer than in the others but still better than the full feature set.

Table 4.2: The accuracies of NB models trained with all the features and the features selected by filters and heuristic ensembles.

Dataset	All features	FCBC	CSF	ReliefF	Gain Raito	HEF	HEF-R1
Zoo	93.96	93.56	94.25	92.27 -	**95.24 +**	95.05	95.05
Dermatology	97.43	97.86	**98.55 +**	96.06 -	85.32 -	98.2 +	98.52 +
Promoters	90.19	**94.62 +**	94.52 +	93.86 +	**94.62 +**	93.71 +	94.57 +
Splice	95.41	96.16 +	96.16 +	96.24 +	95.98 +	96.04 +	**96.33 +**
M-feat-factor	92.47	93.6 +	**93.68 +**	87.16 -	89.98 -	92.53	92.98
Arrhythmia	62.39	65.86 +	68.93 +	65.66 +	53.25 -	68.87 +	**69.6 +**
Colon	55.81	84.67 +	85 +	85.8 +	83.06 +	**85.86 +**	85.55 +
SRBCT	99.04	99.63	**100 +**	**100 +**	99.51	**100 +**	**100 +**
Leukaemia	98.75	**99.44 +**	98.61	95.97 -	95.97 -	98.61	98.61
Ovarian	92.411	**99.92 +**	99.84 +	98.34 +	98.02 +	98.81 +	98.81 +
Average	87.78	92.53	92.95	91.13	89.09	92.76	**93.00**
St. Dv.	14.67	9.86	9.03	9.51	12.99	8.87	**8.74**
W/T/L		7/3/0	8/2/0	6/0/4	5/1/4	7/3/0	7/3/0

The results in Table 4.3 show the accuracy of the KNN (k = 1) classifiers and similar patterns to those that appeared in Table 4.2 can be observed. The one exception is that the CFS filter produced similar accuracy under this experiment condition.

Table 4.3: The accuracies of KNN models trained with all the features and the features selected by filters and heuristic ensembles

Dataset	All features	FCBC	CSF	ReliefF	Gain Raito	HEF	HEF-R1
Zoo	96.14	96.04	96.04	**97.03 +**	96.04	96.04	96.04
Dermatology	94.64	95.57 +	**97.1 +**	94.29	86.45 -	95.54 +	96.91 +
Promoters	79.71	**91.13 +**	**91.13 +**	89.99 +	**91.13 +**	90.19 +	**91.13 +**
Splice	74.43	81.21 +	81.21 +	80.52 +	**82.06 +**	79.59 +	80.46 +
M-feat-factor	96.03	96.36 +	**96.44 +**	93.48 -	95.32 +	96.31 +	96.36 +
Arrhythmia	53.2	59.82 +	61.39 +	57.76 +	43.52 -	57.52 +	**61.88 +**
Colon	76.83	78.38 +	81.45 +	81.45 +	77.74	**86.3 +**	80.71 +
SRBCT	82.39	99.87 +	**100 +**	**100 +**	**100 +**	**100 +**	**100 +**
Leukaemia	88.39	**99.58 +**	97.49 +	95.41 +	94.44 +	98.48 +	98.77 +
Ovarian	94.86	**100 +**	99.96 +	99.13 +	98.85 +	**100 +**	**100 +**
Average	83.72	89.79	90.221	88.90	86.55	89.99	**90.226**
St. Dv.	12.93	12.33	11.63	12.17	15.91	12.47	11.70
W/T/L		9/1/0	9/1/0	8/1/1	6/2/2	9/1/0	9/1/0

Table 4.4 lists the accuracies of the SVM models and the comparisons between the filers. It can be observed that the ensembles performed consistently; this time HEF is the overall winner as it has a marginally higher average accuracy and a lower standard deviation than all the others. One different phenomenon observed is that SVM models trained with the full feature set performed not as badly as the other two types of models (NB and KNN), and even gave the highest accuracy on three datasets (Zoo, Multi-Feature Factor and Arrhythmia). The average accuracy of SVM models trained with all the features is similar to that trained with features selected by the ReliefF filter. It is not much worse than the rest in terms of accuracy, but SVMs using the full features are less efficient than the SVMs using fewer features. Therefore, feature selection is still beneficial with SVM as classifiers.

Table 4.4: The accuracies of SVM models trained with all the features and the features selected by filters and heuristic ensembles.

Dataset	All features	FCBC	CSF	ReliefF	Gain Raito	HEF	HEF-R1
Zoo	**96.24**	96.03	96.13	95.24	95.14 -	95.45 -	95.45 -
Dermatology	96.04	97.67 +	**98.06 +**	95.63	88.71 -	**98.06 +**	98.01 +
Promoters	91.03	92.83 +	92.83 +	91.98	92.83 +	91.86	**92.86 +**
Splice	93.13	95.92 +	95.91 +	**95.98 +**	95.95 +	94.15 +	94.30 +
M-feat-factor	**97.7**	97.15 -	97.26 -	96.12 -	96.91 -	97.62	97.43 -
Arrhythmia	**71.06**	58.6 -	67.83 -	68.36 -	59.13 -	69.62 -	61.86 -
Colon	84.52	88.7 +	88.22 +	87.42 +	83.06	**88.93 +**	86.69 +
SRBCT	99.63	99.63	99.87	**100**	98.67 -	**100**	**100**
Leukaemia	98.04	**99.3 +**	97.49	97.22 -	97.08 -	98.32	98.32
Ovarian	99.96	**100**	100	99.56 -	99.56 -	100	100
Average	92.73	92.58	93.36	92.75	90.70	**93.40**	92.49
St. Dv.	8.46	11.78	9.12	8.82	11.54	**8.62**	10.88
W/T/L		5/3/2	4/4/2	2/4/4	2/1/7	3/5/2	4/3/3

4.5 Conclusion

In this experiment, a framework of a heuristic ensemble of filters has been proposed to overcome the weaknesses of single filters and to improve the accuracy of feature selection. It combines the outputs from two types of filters – SF and RF, with heuristic rules as consensus functions to improve the accuracy and stability in feature selection.

The novelty of the study that has been achieved in this chapter can be summarised as follows:

(1) We have combined SF with RF in our ensemble algorithm to exploit the advantages of each type, whilst the majority of the previous studies on feature selection ensembles focus on ranking filters only. Since RF usually assesses individual features and assigns them weights according to their degree of relevance, while SF takes into account the existence and effect of redundant features. To obtain the benefits of SF without suffering the high computational cost, we chose very fast SFs by modifying their search strategies to make them much quicker.

(2) We use the highest number of features in the SF as a cut-off point for the top-ranking features for the remaining ranking filters, which should accelerate the ensemble algorithm. This is because we do not need to select various feature numbers to test the performance of the rankers (as other researchers have done) or to use a wrapper to choose the appropriate number of features.

(3) We have applied heuristic consensus rules to remove the selected features that have low frequency and also to obtain more important features. As the combination method used counts the most frequently selected features, it is therefore possible to have a high number of features selected by the ensemble filters including features with low frequency levels selected by only a couple of filters (or even a single one).

The proposed HEF and HEF-R1 have been tested on 10 benchmark datasets where features varied from 17 to as many as 15,154. The statistical analysis on the experimental results shows that the ensemble technique performed more consistently and in some cases even more accurately than individual filters.

Specifically,

(1) HEF-R1 performed best for NB and KNN, while HEF performed best when using the SVM classifier, which demonstrates that our proposed ensemble is more accurate and consistent than using single filters.

(2) There is no single best approach for all the situations. In other words, the performance of the single filter varies from dataset to dataset and also was influenced by the type of models chosen as a classifier. Thus, one filter may perform well in a given dataset for a particular classifier but perform poorly when used on a different dataset or with a different type of classifier.

(3) Among the four filters we used in our heuristic ensemble of filters, the SF (FCBF and CSF) were more frequently better and less frequently worse on average in term of accuracy than the RF.

(4) The experiment results show that the ensemble technique performed better overall than any individual filter in terms of consistency and accuracy.

However, some important issues have been identified in this initial study and thus need to be investigated in the remaining chapters of this research. Firstly, we need to determine appropriate approaches for using data in feature selection. This issue is important since it is a general and important issue in FS, and no clear answer has been obtained from the existing studies. Consequently, we have designated the next chapter to investigate this issue before carrying on with the remaining research. After that, we will build the remaining studies based on the results in Chapter 5. Secondly, we should consider the types of filters and number of filters that should be included in the proposed ensemble, in addition to choosing a suitable aggregation method, which is an important decision to make. Furthermore, we should consider how to extend the HEF by applying different wrappers after analysing the results obtained by HEF, aiming to reduce the number of features selected, while preserving the same accuracy and stability. Finally, we will investigate whether weighting the filter members in an ensemble differently may lead any further improvement of the performance of the HEF.

Chapter 5

Determining Appropriate Approaches for Using Data in Feature Selection

5.1 Introduction

In the previous chapter, the framework for HEF was proposed to overcome the weaknesses of single filters and to improve the accuracy of FS. FS methods were applied on 10 real benchmark datasets by using entire datasets and then using the selected features as an input for the classifier (ALL method). In this chapter, we evaluate the FS method on generated synthetic datasets in addition to the same real-world benchmark datasets that we used in Chapter 4, but with a different method, which performs FS inside the cross-validation loop by executing the FS method on the training set before classifier construction in each iteration (PART method).

Accordingly, if the aim is to treat FS as a pre-processing step for dimensionality reduction, it would be appropriate to use the ALL method by separating FS from classifier learning, and using the whole dataset with the FS step (Refaeilzadeh et al., 2007). On the other hand, if the aim is to compare two FS algorithms or to search for important features in the dataset (and we need the classifier as an indirect evaluation tool), then in these two cases, to the best of our knowledge, the literature does not provide any clear answer as to which evaluation method (PART or ALL) is more reliable, especially when using filters.

Therefore, the motivation of this chapter is to investigate whether PART or ALL is more appropriate in FS. In order to answer this question, firstly, we compare the results of the PART method with the ALL method which was described in Chapter 4, on the same real-world benchmark datasets. Secondly, we generate synthetic datasets with different numbers of features and samples as well as levels of noise (Section 5.4). We also use a bench mark synthetic dataset. Thirdly, we use suitable stability measures to evaluate the stability of each method and to evaluate the ability of each method to identify more relevant features, in addition to the traditional way of evaluating FS by using a classifier.

In this chapter, 21 synthetic datasets will be generated and described to check the accuracy of several FS methods and their evaluation approach in an artificially controlled experimental scenario. A stability measure will be introduced to compute the degree of matching between the output given by the algorithm with both (PART and ALL) methods and the known optimal solution, as well as the classification accuracy. Finally, the conclusion extracted from this empirical study can be extrapolated to the remainder of this research. The work in this chapter has been published at the thirty-

fourth SGAI International Conference on Artificial Intelligence in 2014 and International Journal of Machine Learning and Cybernetics in 2015.

This chapter is organised as follows: Section 5.2 describes the PART and ALL methods in more detail. Section 5.3 presents the related work about the PART and ALL methods. Section 5.4 describes how to generate the synthetic dataset and explains the experimental design. Section 5.5 shows the experiment's results by measuring the classification accuracy and the stability of FS. Discussions are presented in Section 5.6 and conclusions are presented in Section 5.7.

5.2 The PART and ALL Methods

In general, it is reasonable to assume that the quality of the selected features is correlated with the number of samples available during training. So, in order to increase the chance of selecting the most relevant features and then to build better models, we should use all the available data (Refaeilzadeh et al., 2007) in FS. The ALL method has been commonly used in FS, using the entire dataset in the selection step, and the selected subsets of features are then used as the inputs for building classifiers, as seen in Fig 5.1. However, using the entire dataset for FS before classification learning may produce over-optimistic results, as it has seen the test data in training. This is called 'feature subset selection bias'; some studies (Ambroise, 2002, Lecocke and Hess, 2006, Singhi and Liu, 2006, Chen et al., 2006, Refaeilzadeh et al., 2007) have discussed this issue and attempted to solve it by using the PART method.

The PART method employs a *k*-fold cross-validation mechanism in the hope of avoiding this bias. *k* -1 folds are used as the training data for each filter, the selected features are used as the inputs for the classification base learner to build the classifier with the same *k* -1 folds of the data, and then the remaining fold is used as a validation set to test the classifier, as in Fig 5.2. This procedure is repeated for a *k* times 'round robin'. The average accuracy of the classification over *n* runs will be calculated as an indicator of the effectiveness of the feature selection. Nevertheless, holding out one fold for FS in the PART method might exacerbate the 'small sample' problem with FS, as many datasets have small numbers of samples, which may lead to underestimating the relevant features under some conditions.

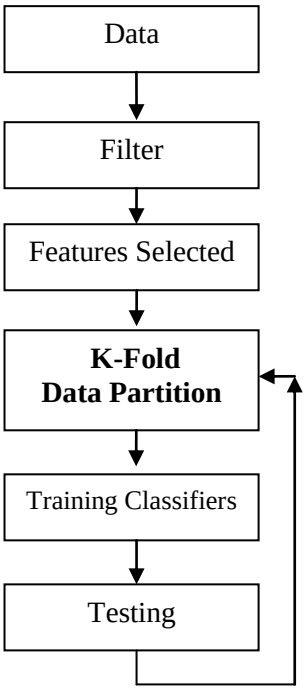


Figure 5. 1: ALL Method

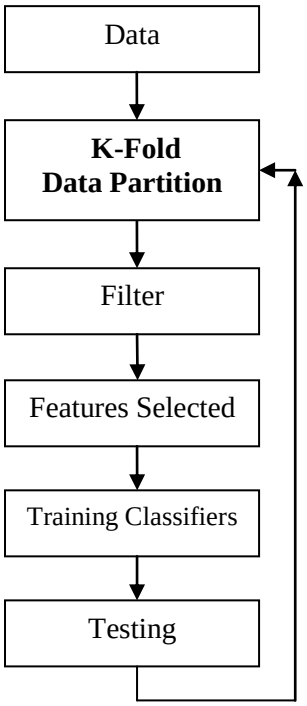


Figure 5. 2: PART Method

5.3 Related Works about PART and ALL Methods:

In recent years, a few studies (Refaeilzadeh et al., 2007, Singhi and Liu, 2006, Ambroise, 2002, Reunanen, 2003, Lecocke and Hess, 2006) have discussed the influence of using FS on the whole dataset and have attempted to solve any problems by performing FS inside the CV loop; however, these studies have certain limitations. Ambroise's (2002) study was the first attempt to correct the selection bias by either performaing cross-validation or bootstrap on the selection process. In that study, they used both backward (with SVM) and forward selection (with LDA wrapper approaches) and no filter model was used. Also, they recommended using 10 folds rather than leaving one out for cross-validation. Reunanen (2003) studied the FS evaluation method using wrapper models only, but did not address issues specifically relating to the pair-wise comparison of FS algorithms. Also, Lecocke and Hess (2006) presented an empirical study in which the PART method with 10-fold CV is applied to filters (t-tests) and wappers (genetic algorithms; GA). In this study, two measures of bias are considered. Firstly, the optimism bias where the "estimate represents the bias incurred from using the same data to both train the classifier and estimate the performance of the classifier" (Lecocke and Hess, 2006). Secondly, the selection bias where the "estimate

represents the bias incurred from using the same data to both select the gene subsets and estimate the performance of the classification rule based on these subsets" (Lecocke and Hess, 2006). They found that the optimism bias estimates from the GA analyses were half the t-test, while the selection bias estimates from the GA were 2.5 times that of the t-test results. This means that the filter model had higher optimism bias and lower selection bias than the wrapper model. However, the limitation of this study is that they used just binary classification with microarray data and only two FS methods.

Moreover, Refaeilzadeh, Tang et al. (2007) studied which evaluation method (PART or ALL) is more reliable when conducting pair-wise comparisons of FS algorithms by concentrating on filter models and by using 10-fold CV with paired t-test. Additionally, they generated 5 data sources (2 continuous and 3 discrete) but the highest number of features was only 60 and the maximum number of instances was 1,000. They explained that there is a potential for bias in both the PART and ALL methods; with ALL, the FS method looked at the test set, so the accuracy estimate is probably inflated, whereas with the PART method, the FS method is looking at less data than would be available in a real experimental setting, which may have led to underestimating the accuracy. The results obtained from that study include: (1) PART and ALL "have different biases, and bias is not a major factor" in determining which one is more truthful in pair-wise comparison (Refaeilzadeh et al., 2007); (2) in a greater majority of cases, PART and ALL approaches are not significantly different; (3) the PART approach tends to be more truthful if the two FS methods are performed identically; (4) given two FS methods A1 and A2, for two cases, "(a) A1 is better and (b) A2 is better, if PART is better for case (a), then ALL is better for case (b) "(Refaeilzadeh et al., 2007). However, some of their conclusions are not clear, such as they "recommend to run both methods ALL and PART, trust the method indicating that one algorithm is better than the other, and use that better algorithm to select features using the entire dataset. In the worst case scenario, the selected features will be no worse than the subset selected by the alternative algorithm." Also, other limitations of their study are that they only used synthetic datasets with relatively low dimensions (<= 60), and a small number of samples, with the highest number of instancs equal to 1,000.

Finally, these studies attempted to determine whether PART or ALL is more appropriate as an evaluation method, but this question is still open, especially when using filters, and no clear answer has been obtained. For this reason, we decided to

evaluate these two approaches systematically and determine their stability and effectiveness while using filter methods.

5.4 Experiments

5.4.1 Data

5.4.1.1 Real world Bench Mark Data

10 benchmark datasets from different domains were used in our experiments (the same as we used in Chapter 4) in order to study the differences between the PART and ALL methods. Table 3.2 in Section 3.5.1 summarises the general information pertaining to these datasets.

5.4.1.2 Generation of Synthetic Datasets

In practice, using synthetic data represents a useful strategy for testing the effectiveness of FS for the following reasons (Belanche and González, 2011):

1- Knowing the optimal features in advance is the main advantage of synthetic data. Then, we can compute the degree of matching between the output given by the algorithm and the known optimal solution.

2- Being able to conduct the investigations in a systematic way, by modifying the experiment conditions, like changing the ratio between the number of samples and number of features, or adding more irrelevant features or noise to the input.

In fact, this technique allows one to draw more useful conclusions and to assess the strong and weak points of the existing algorithms.

The datasets generated for this study try to cover different problems, such as increasing the number of irrelevant features, and decreasing the number of instances and varying level of noise in the response variable. These are some of the factors that make the FS task difficult.

The synthetic datasets generated are subsequently described in general, and then each step in this process is illustrated. The synthetic datasets generated are of linear problems

as shown by equation (5.1) and all features have continuous values (even the response variable). However, in order to use these datasets in the classification problem, we convert the response variable to binary.

The following steps were taken to generate these datasets, where N_R represents the number of relevant features, N_I the number of irrelevant features, N the number of total features, S the number of instances, and y_c the response variable.

Step 1: Random matrix D (N, S) of S samples is generated with N independent and identically distributed random features (iid), with a given mean μ and a standard deviation σ.

$$D(N,S)=\begin{bmatrix} x_{11}\ x_{12}\ \ldots\ldots x_{1N} \\ \vdots \\ x_{S1}\ x_{S2}\ \ldots\ldots x_{SN} \end{bmatrix}$$

Then we expand this matrix by increasing N and S. However, for simplicity we remove the index. So, any instance (d) has N features and the response variable y_c , as follows:

$$\forall\ d = [x_1,\ x_2,\ x_N,\ y_c]$$

Step 2: N_R is selected as relevant features, and then coefficient β_i is generated and N_R is multiplied ($x_1 \ldots\ldots x_{N_R}$) with the β value.

$$\beta = \{\beta_1, \beta_2, \ldots\ldots \beta_{N_R}\}$$

$$\text{S.T: } \sum_{i=1}^{N_R} \beta_i = 1$$

Step 3: The response variable y_c is computed by summing the value of the relevant variable $\sum_{i=1}^{N_R} \beta_i\, x_i$, without including irrelevant features or noise, as shown in the following equation:

$$y_c = \sum_{i=1}^{N_R} \beta_i\, x_i \ + \sum_{j=1}^{N_I} \gamma_j\, x_j \qquad (5.1)$$

where all x_i and x_j are iid, $\gamma_j = 0, \gamma_j (j = 1, \ldots\ldots, N_I)$ is set to be zero, so that , N_I features become irrelevant.

Step 4: The response variable y_c is converted from continuous to binary by

$$y = \begin{cases} 0, & y_c < \bar{y} \\ 1, & y_c \geq \bar{y} \end{cases} \qquad (5.2)$$

where $\mu = \bar{y} = \frac{\sum_{i=1}^{S} y_{ci}}{S}$ (5.3)

There are a number of key points to this synthetic data generation strategy, which can be explained as follows:

Relevance: Relevant features (N_R) or "optimal" features are defined as those having influence on the output (response variable) and whose role cannot be assumed by any other subset. In these experiments, we set $N_R = 10$, considering relevant features, while the remaining features are irrelevant. Then N_R will be changed to 4 and 16, to note the effect of N_R on the performance of individual FSs and ensemble methods.

Irrelevance: Irrelevant features (N_I) are defined as those not having any influence on the output. The number of irrelevant features $N_I$varies from 84 to 9,996 features, which are generated randomly for each instance.

Total Number of Features: N is the total number of features (N_R + N_I) in these experiments; N varies from 100 to 10,000. This means that the greatest variation is usually in the number of irrelevant features N_I because the number of relevant features N_R is fixed in the first 9 datasets, then changed to 4, and after that to 16 relevant features.

Sample Size: In these experiments, the number of instances S varies from 100 to 10,000, similar to the changes in the total number of features N.

Noise Injection Mechanism: ε is a noise injected into some samples of the response variable, with differing levels. The levels of noise in the response variable are regulated by two noise parameters. The first parameter, denoted by e (e =5%, 10%), is used to determine the number of samples injected by noise. The second parameter, denoted by ε, which is a random number varying between $\varepsilon = -0.1 \rightarrow 0.1$, represents the proportion of noise injected to response variable. y_e is the response variable injected by noise, defined as follows:

$$y_e = \sum_{i=1}^{N_R} \beta_i x_i + \sum_{j=1}^{N_I} \gamma_j x_j + \varepsilon \quad (5.4)$$

Where $\varepsilon \in N(0,1)$, e= 5% or 10% of S and $\gamma_j = 0$

Other Parameters: The mean is denoted by μ and standard deviation by σ. The β_{N_R} value starts with β_1, then each β_{i+1} is added by Δβ and so on, for the first 9 datasets. Then β_{N_R} will change as illustrated in Section 8.2.2

$$\beta_{i+1} = \beta_i + \Delta\beta \quad (5.5)$$

$\beta = \{\beta_1, \beta_2, \ldots\ldots\ldots\ldots\ldots \beta_{10}\}$

S.T: $\sum_{i=1}^{N_R} \beta_i = 1$

A. Synthetic Datasets with Different Numbers of Samples and Irrelevant Features

Table 5.1 shows a summary of the 9 synthetic datasets generated with different numbers of samples S, N is total number of features, N_R is number of relevant attributes (which should be selected by the feature selection methods), N_I is number of irrelevant features.

These 9 synthetic datasets as shown in Table 5.1 have 10 N_R and their class values are computed by summing the first 10 features, after multiplying them with β_i as follows:

$$y_c = \beta_1 x_1 + \beta_2 x_2 + \ldots\ldots\ldots + \beta_{10} x_{10} \quad (5.6)$$

$$\beta_{i+1} = \beta_i + \Delta\beta$$

where $\beta_1 = 0.01 \; and \; \Delta\beta = 0.02$

Table 5. 1: Summary of the 9 synthetic datasets from S1 to S9 without noise injection

Dataset	***S***	***N***	N_R	N_I
S1	*100*	*100*	*10*	*90*
S2	*1000*	*100*	*10*	*90*
S3	*10000*	*100*	*10*	*90*
S4	*100*	*1000*	*10*	*990*
S5	*1000*	*1000*	*10*	*990*
S6	*10000*	*1000*	*10*	*990*
S7	*100*	*10000*	*10*	*9990*
S8	*1000*	*10000*	*10*	*9990*
S9	*10000*	*10000*	*10*	*9990*

This means x_{10} is the most relevant feature, while x_1 in the least relevant feature to the class, based on the above equations. The remaining features are irrelevant to the response variable and were generated randomly.

Firstly, we started to construct S1 with S = 100, $N_R = 10$ and N_I = 90, and then S2 was constructed by adding 900 samples to S1. Similarly, S3 was constructed by adding 9,000 samples to S2. On the other hand, S4 was constructed by adding 900 irrelevant features to S1. In the same way, S5 was constructed by adding 900 irrelevant features to S2, and S6 by adding 900 irrelevant features to S3. The final 3 datasets were constructed by increasing the total features with 9,000 irrelevant features, S7 was constructed by adding 9,000 irrelevant features to S4, and S8 by adding 9,000 irrelevant features to S5; finally, S9 by adding 9,000 irrelevant features to S6.

We are aiming to cover different situations from an uncomplicated problem, which has a low number of irrelevant features with a high number of samples, to a challenging problem that has a high number of irrelevant features and a low number of samples; this case reflects the challenge in microarray data.

B. Synthetic Datasets with Different Numbers of Relevant Features

In this section, we change the number of relevant features, aiming to identify the effect of the number of relevant features on the ability of FS to identify these features. Accordingly, we selected three datasets from the above group (S2, S5 and S8), which have a reasonable number of samples (1,000) in order to focus on selecting relevant features and avoiding the influence of the sample number.

Table 5.2: Summary of the 6 synthetic dataset with different N_R without noise injection

Dataset	***S***	***N***	N_R	N_I
S2NR4	*1000*	*100*	*4*	*96*
S5NR4	*1000*	*1000*	*4*	*996*
S8NR4	*1000*	*10000*	*4*	*9996*
S2NR16	*1000*	*100*	*16*	*84*
S5NR16	*1000*	*1000*	*16*	*984*
S8NR16	*1000*	*10000*	*16*	*9984*

Table 5.2 presents a summary of the 6 synthetic datasets generated with different numbers of relevant and irrelevant features, with the same number of samples. The first 3 synthetic datasets, as shown in table 5.2, have $N_R = 4$ and their class values are computed by the equation below:

$$y = \beta_1 x_1 + \beta_2 x_2 + \ldots\ldots\ldots + \beta_4 x_4 \quad (5.7)$$

$$\beta_{i+1} = \beta_i + \Delta\beta$$

where $\beta_1 = 0.1 \; and \; \Delta\beta = 0.1$, so that $\sum \beta_i$=1.

This means x_4 is the most relevant feature while x_1 is the least relevant feature to the class, based on the above equations. The remaining features are irrelevant to the class label and were generated randomly.

S2NR4 was constructed by adding 900 samples to the basic matrix, then multiplying the first four features with β (0.1,0.2,0.3 and 0.4) sequentially, to construct the class label (response variable), having converted the response variable y from continuous to binary, by using Equations 5.2 and 5.3. S5NR4 and S8NR4 were constructed by adding 900 and 9,000 irrelevant features to S2NR4 sequentially.

The last three synthetic datasets, as shown in Table 5.2, have N_R = 16 and their class value is computed by summing the first 16 features after multiplying it with β_i as follows:

$$y = \beta_1 x_1 + \beta_2 x_2 + \ldots\ldots\ldots + \beta_{16} x_{16} \quad (5.8)$$

$$\beta_{i+1} = \beta_i + \Delta\beta$$

where $\beta_1 = 0.025 \; and \; \Delta\beta = 0.005$

This means that x_{16} is the most relevant feature while x_1 is the least relevant feature to the class based on the above equations. The remaining features are irrelevant to the class label and were generated randomly.

Similarly, S2NR16 was constructed by adding 9,900 samples to the basic matrix, then multiplying the first 16 features with their corresponding β to construct the class label. S5NR16 and S8NR16 were constructed by adding 900 and 9,000 irrelevant features to S2NR16 sequentially.

There is not any particular reason to choose $N_R = 4$, it is just for convenience, by adding a fixed number to $\Delta\beta = 0.1$ and making S.T: $\sum_{i=1}^{N_R} \beta_i = 1$. Also the same reason applies to $N_R = 16$, by adding a fixed number to $\Delta\beta = 0.005$ and making $\sum_{i=1}^{N_R} \beta_i = 1$.

C. Synthetic Dataset with Injected Noise

The aim of the above synthetic datasets was to evaluate the performance of the PART and ALL methods on individual FS and ensemble methods in the presence of the relevant number, as well as the ratio between the number of samples and the number of features. In this section, we seek to simulate real datasets, which usually have different degrees of noise, by injecting the noise to response variable with different rates into 3 datasets. The first parameter, denoted by e (e =5%,10%), is used to determine the number of samples injected by noise. The second parameter, denoted by ε, which is a random number varying between $\varepsilon = -0.1 \rightarrow 0.1$, represents the proportion of noise injected to response variable.

Table 5.3: Summary of the 6 synthetic datasets after adding noise to the class y

Dataset	***S***	***N***	N_R	N_I	α_k
S2Noise5	*1000*	*100*	*10*	*90*	*5%*
S2Noise10	*1000*	*100*	*10*	*90*	*10%*
S5Noise5	*1000*	*1000*	*10*	*990*	*5%*
S5Noise10	*1000*	*1000*	*10*	*990*	*10%*
S8Noise5	*1000*	*10000*	*10*	*9990*	*5%*
S8Noise10	*1000*	*10000*	*10*	*9990*	*10%*

Table 5.3 presents a summary of the 6 synthetic datasets generated with different rates of class noise (injected) and different numbers of irrelevant features but with same numbers of relevant features and samples. The first 2 synthetic datasets (S2Noise5, S2Noise10), as shown in Table 5.3, have the same parameters except the rate of noise. S2Noise5 was injected with 5% of the samples by adding or subtracting a random number between $\varepsilon = -0.1 \rightarrow 0.1$ to the response variable, which may cause a change in

the class label from 0 to 1 or from 1 to 0. S2Noise10 was injected with 10% of the samples through $\varepsilon = -0.1 \rightarrow 0.1$ to the response variable. The second 2 synthetic datasets (S5Noise5, S5Noise10), and the last 2 synthetic datasets (S8Noise5, S8Noise10) had the same process; the only difference between these datasets is the number of irrelevant features.

5.4.2 Experiment Design and Procedure

As it is generally accepted that the effectiveness of feature selection can be indirectly evaluated through measuring the classification accuracy of those classifiers that are trained on the selected features, we thus conducted several series of experiments with a variety of datasets to empirically evaluate the accuracy of the PART method and to compare it with the ALL method. In our experiments, we use three types of classifier: NB (John and Langley, 1995), KNN (Aha et al., 1991) and SVM (Platt, 1999). These three algorithms were also used in Chapter 4.

For each dataset, the experiments with the ALL are carried out in two phases: feature selection then evaluation by classifiers. The ALL method uses the entire dataset with each FS method, and the subsets produced by these FS methods (4 filters and 2 ensembles) are used as input for the classifier. A 10-fold cross-validation strategy is used with the classifier, and after that we average the accuracy of 10 folds. Then, each experiment is repeated 10 times with different shuffling random seeds in order to assess the consistency of the results. The average accuracy as well as the similarity of 10 runs will be presented in the final result.

The experiments with the PART are carried out in one phase and in the same fold: feature selection and evaluation. We firstly run individual filters to produce a subset of features, as well as to compute the HEF in order to produce subsets of rank features. Then, we evaluate the effectiveness of the selected features with three kinds of models: NB, KNN and SVM. Specifically, in each fold, we firstly run FS methods (FCBF, CFS, ReliefF, Gain Ratio, HEF and HEF-R1) by using 90% of all the instances (9 folds), after which the subsets produced by each FS are used as input to the classifier with the same 90% of instances (9 folds). Following this, the accuracy of this subset was estimated over the unseen 10% of the data (1 fold). This was performed 10 times, each time

proposing a different possible feature subset. In this way, estimated accuracies and selected attribute numbers were the result of a mean over 10 cross-validation samples. Ambroise and McLachlan (2002) recommend using 10-fold rather than leave-one-out cross-validation, because the latter one can be highly variable. Each experiment is then repeated 10 times with different shuffling random seeds in order to assess the consistency of the results. In total, 46,800 models were built for the experiments as follows: by using synthetic data 28,800 were built (6 (FS + ensemble) × 2 (PART + ALL) × 24 (21 synthetic datasets + 4 bench mark synthetic data) × 10 (run) × 10 (folds)) and by using real-world bench mark 18,000 models were built (6 (FS + ensemble) × 10 (real would bench mark) × 3 (classifiers) × 10 (runs) × 10 (folds)).

The statistical significance of the results of the multiple runs for each experiment is calculated, and the comparison between accuracies is done with Student's paired two-tailed *t*-test with a significance level of 0.05, which is a test that takes into account the variance in the accuracy estimates (Dietterich, 1998), and it is often used in machine learning.

Moreover, in addition to accuracy, we will measure the stability of FS, as in each fold the FS method may produce different feature subsets with the PART method, and in order to identify the factors that play the most important roles. Measuring stability requires a similarity measure for the FS results. There are three types of representation methods: subset of features, ranking vector and weighting score vector (He and Yu, 2010). In this work, we focus on subsets of features because our filter-based ensemble algorithm produces subsets of features. The stability measures used in our investigation are: Relative Weighted Consistency (CWrel) and Average Tanimoto Index (ATI) (Somol and Novovicova, 2010), as the subset cardinality is not equal in our research. ATI evaluates pair-wise similarities between subsets in the system (10 folds), while CW_{rel} evaluates the overall occurrence of the features in the system (10 folds) as a whole. CW_{rel} and ATI may produce different results in each run, so the average of 10 runs will be used. Also, we included more measures in our investigation, called inter-measures, in order to compare the features selected from the PART method (in each fold) with the ALL method. The Intersystem Weighted Consistency (IWC) and the Intersystem Average Tanimoto Index (IATI), which is provided in (Somol and Novovicova, 2010), are used in this investigation.

The IATI was used to measure the amount of overlapping between any two sets. In this case, the first set is the optimal features (in the case of synthetic dataset) and the second set is the subset selected from the FS methods, while ICW was used to compare the frequencies of the more frequent features. The third and fourth measures are ATI and CWrel, respectively, which evaluate the stability of the FS process with the PART method by changing the samples using cross-validation.

5.5 Results

5.5.1 Real-World Bench Mark Dataset

5.5.1.1 Number of Selected Features

Table 5.4 lists the average number of features selected by each filter in addition to two heuristic ensembles: HEF and HEF-R1.We observed from the table that the average number of selected features dramatically reduced the dimensionality of the data by selecting only a small portion of the original features in those datasets. Although HEF represents the total number of features selected from all the four filters, it is still less than the average full set by up to 50 times for genetic datasets.

Also, compared with the results given in Tables 4.1 and 5.4, it can be noted that there is no big difference between the PART (on average) and ALL methods in the number of selected features; the PART method has one or two fewer features (on average) than the ALL method for all filters and HEF-R1, while HEF has same features (on average) as the ALL method.

Table 5.4: Average number of selected features by each filters and ensemble

Number of Features	*All features*	*FCBC*	*CFS*	*ReliefF*	*Gain Ratio*	*HEF*	*HEF-R1*
Zoo	*17*	*7*	*10*	*10*	*10*	*11*	*10*
Dermatology	*34*	*15*	*18*	*18*	*18*	*28*	*24*
Promoters	*55*	*6*	*6*	*6*	*6*	*8*	*8*
Splice	*61*	*21*	*21*	*21*	*21*	*27*	*25*
M-feat-factor	*216*	*39*	*45*	*45*	*45*	*87*	*57*
Arrhythmia	*279*	*12*	*21*	*21*	*21*	*55*	*17*
Colon	*2,000*	*14*	*19*	*19*	*19*	*43*	*19*
SRBCT	*2,308*	*72*	*72*	*72*	*72*	*170*	*83*
Leukaemia	*7,129*	*45*	*55*	*55*	*55*	*122*	*59*
Ovarian	*15,154*	*27*	*32*	*32*	*32*	*75*	*35*
Average (PART)	*2,725.3*	*25.8*	*29.9*	*29.9*	*29.9*	*62.6*	*33.7*
The average number of features by ALL method from Table 4.1							
Average(ALL)	*2,725.3*	*27.8*	*31.3*	*31.8*	*31.8*	*62.3*	*35.9*
Δx=ALL-PART	*0*	*2*	*1.4*	*1.9*	*1.9*	*-0.3*	*2.2*

However, comparing the average number of selected features using the PART method with the number of selected features using the ALL method, without a close look at each fold in the PART method, may overlook some useful information. This is because when we go deeper inside each fold, we find variations in the number of selected features, from one fold to another and from one run to another. Figures 5.3 and 5.4 illustrate an example of this variation in the number of selected features by using the PART method.

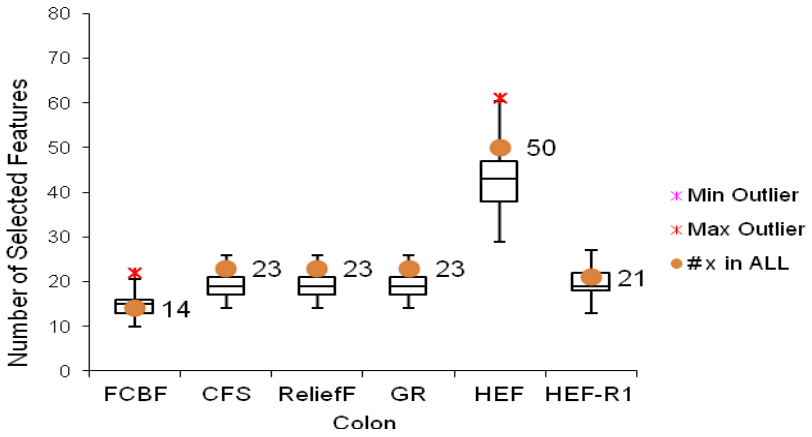


Figure 5.3: Number of selected features by the PART method on the Colon dataset

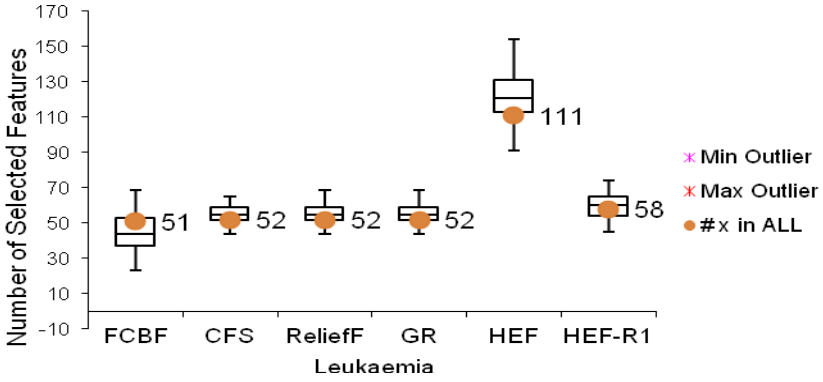


Figure 5.4: Number of selected features by the PART method on the Leukaemia dataset

Figures 5.3 and 5.4 show that the number of features selected by each FS method changes over 10 runs of 10-fold cross-validation by the PART method. Also, the pink star represents the minimum number of features selected and the red star represents the maximum number of features selected, while the orange circle illustrates the number of features selected by the ALL method (the remaining dataset figures are provided in Appendix A). We can observe that this change varies based on the dataset and FS method used. As we can see, HEF has the highest level of change, as it is aggregating the outputs of four filters, while HEF-R1 has an almost similar level of change and number of features as the other filters, which shows that many of the features selected by HEF were selected by only one filter. HEF-R1 thus selects a lower number of features by removing them. Also, FCBF usually selects a lower number of features than CFS. In sum, there is actually a difference between the PART and ALL methods in the number of selected features, which is not clear when we use the average number of features with the PART method as seen in Table 5.4.

5.5.1.2. Accuracy Evaluation with Different Classifiers

Tables 5.5, 5.6 and 5.7 show the average accuracy of the NB, KNN and SVM models on the 10 datasets; each value presented in the tables is the average over 10 runs of 10-fold cross-validation outcomes using the PART method. For each classifier, the classification accuracies on the datasets with all the original features are given in the 'All features' column for comparison purposes. The notations '+' or '-' denote that the result of the classification of the models trained with the features selected with the current selector by the PART method is significantly better or worse than that of the models trained with the same selector by the All method in the statistical test mentioned earlier. The bold value in each row shows the best classification result. The last three rows in each table show Average (the average accuracies), St. Dv. (the standard deviations for the accuracies) and W/T/L (which summarises the wins/ties/losses in accuracy by comparing the models trained with the PART method and the ALL method.

Table 5. 5: The accuracies of Naïve Bayesian classifier trained with all the features and the features selected by filters and heuristic ensembles by the PART method

Dataset NB	***All features***	***FCBC***	***CFS***	***ReliefF***	***Gain Raito***	***HEF***	***HEF-R1***
Zoo	*93.96*	*93.45*	*93.3 -*	*94.28 +*	*93.59 -*	***94.46***	*94.07*
Dermatology	*97.43*	*97.49*	*98.09 -*	*95.91*	*85.45*	*97.79*	***98.31***
Promoters	*90.19*	***92.48 -***	***92.48 -***	*90.39 -*	*92.19 -*	*91.7*	*92.01 -*
Splice	*95.41*	*95.84 -*	*95.84 -*	***96.32***	*95.98*	*96.21*	*96.18*
M-feat-factor	*92.47*	***93.93 +***	***93.96 +***	*87.82 +*	*89.82 -*	*92.45*	*93.01*
Arrhythmia	*62.39*	***68.1 +***	*68.72*	*63.48 -*	*54.71 +*	*66.89 -*	*67.32 -*
Colon	*55.81*	*80.22 -*	*82.21 -*	*84.33 -*	*79.12 -*	***85.4***	*84.29*
SRBCT	*99.04*	*95.56 -*	*97.21 -*	*99.06 -*	*99.17*	***99.28 -***	*98.67 -*
Leukaemia	***98.75***	*95.68 -*	*96.09 -*	*95.18*	*95.8*	*95.82 -*	*95.96 -*
Ovarian	*92.411*	***99.72***	*99.45*	*97.7 -*	*97.81*	*98.49 -*	*98.97*
Average	*87.78*	*91.24*	*91.73*	*90.44*	*88.36*	*91.849*	***91.879***
St. Dv.	*14.67*	*9.17*	*8.91*	*9.99*	*12.60*	*9.17*	*9.16*
W/T/L		***2/3/5***	***1/2/7***	***2/3/5***	***1/5/4***	***0/6/4***	***0/6/4***
The accuracy results of the ALL method from Table 4.2							
Average	*87.78*	*92.53*	*92.95*	*91.13*	*89.09*	*92.768*	***93.002***
St. Dv.	*14.67*	*11.788*	*9.12*	*8.82*	*11.54*	*8.62*	*10.88*

Table 5.5 shows the results on the 10 real datasets with the Naïve Bayesian classifier and the accuracy comparison between all the features without FS and the features selected by four individual filters and two ensembles. As expected, the accuracy using the PART method decreases on average by -1.292, -1.219, -0.689, -0.731, -0.919 and-1.123, respectively, relative to the ALL method in Table 4.2. FCBF and CFS show the highest decline with the PART method, followed by HEF-R1, HEF and Gain Ratio, while ReliefF has the smallest decrease. Furthermore, the microarray dataset (Colon to Ovarian) in particular, shows a significant decline with the PART method in most of FS methods.

In addition, each single filter performed well in some datasets (in bold) but poorly in others. This confirms the perception that the performance of individual filters is such that no meaningful pattern can be extracted to indicate when they do better and when they do not. Nevertheless, the NB classifiers trained with the features selected by HEF-R1 have a higher average accuracy for all the datasets, which indicates that HEF-R1s are more accurate than the individual filters in FS.

Table 5.6: The accuracies of the KNN models trained with all the features and the features selected by filters and heuristic ensembles by the PART method

Dataset KNN	*All features*	*FCBC*	*CFS*	*ReliefF*	*Gain Raito*	*HEF*	*HEF-R1*
Zoo	*96.14*	*95.13 -*	*95.63*	*96.35 -*	*96.22 -*	***96.44***	*96.23*
Dermatology	*94.64*	*95.0 -*	***96.64***	*93.55 -*	*86.47*	*95.8*	*96.61*
Promoters	*79.71*	*87.61 -*	*87.61 -*	*84.67 -*	***90.11 -***	*85.47 -*	*87.75 -*
Splice	*74.43*	*80.9*	*80.9*	*81.22 +*	***82.37 +***	*79.4*	*80.36*
M-feat-factor	*96.03*	*96.29*	***96.42***	*94.1 +*	*95.24*	*96.15 -*	*96.17 -*
Arrhythmia	*53.2*	*60.94*	***61.46***	*55.84-*	*45.93 +*	*56.61*	*59.01 -*
Colon	*76.83*	*79.17*	*79.38*	*78.57 -*	***80.0 +***	*77.79 -*	*79.21*
SRBCT	*82.39*	*98.21 -*	*99.65*	***100.0***	*99.65*	*99.75*	*99.76*
Leukaemia	*88.39*	***94.88 -***	*94.2 -*	*93.45 -*	*92.66 -*	*94.48 -*	*94.55 -*
Ovarian	*94.86*	*99.76 -*	*99.68 -*	*98.97*	*98.86*	*99.52 -*	***99.84***
Average	*83.724*	*88.789*	***89.157***	*87.673*	*86.751*	*88.141*	*88.949*
St. Dv.	*12.93*	*11.45*	*11.53*	*12.69*	*15.00*	*12.96*	*12.64*
W/T/L		*0/4/6*	*0/7/3*	*2/2/6*	*3/4/3*	*0/5/5*	*0/6/4*
The accuracy results of the ALL method from Table 4.3							
Average	*83.724*	*89.796*	*90.221*	*88.906*	*86.555*	*89.997*	***90.226***
St. Dv.	*12.93*	*12.33*	*11.63*	*12.17*	*15.91*	*12.47*	*11.70*

The results in Table 5.6 show the accuracy of the KNN (k = 1) classifiers. The accuracy using the PART method decreases on average by -1.007, -1.064, -1.233, +0.196, -1.856 and -1.277, respectively, relative to the ALL method in Table 4.3. HEF has the highest decline with the PART method, followed by the other FS models, while Gain Ratio increases the accuracy using the PART method. Moreover, the degree of significant changes in the accuracy between the PART and ALL methods differs from one classifier to another, as well as from one FS to another.

Table 5.7: The accuracies of the SVM models trained with all the features and the features selected by filters and heuristic ensembles by the PART method

Dataset SVM	*All features*	*FCBC*	*CFS*	*ReliefF*	*Gain Raito*	*HEF*	*HEF-R1*
Zoo	***96.24***	*95.13 -*	*95.84*	*94.85*	*95.73*	*95.74*	*95.44*
Dermatology	*96.04*	*97.03-*	*97.51-*	*95.6*	*88.16*	*97.29-*	***97.71***
Promoters	*91.03*	***92.25***	*92.15*	*88.89-*	*91.65*	*90.02-*	*90.89-*
Splice	*93.13*	*95.48-*	*95.48-*	***96.14+***	*95.9*	*95.68+*	*95.79+*
M-feat-factor	***97.7***	*97.25*	*97.42*	*96.13*	*96.51-*	*97.68*	*97.17-*
Arrhythmia	***71.06***	*60.45-*	*66.24-*	*67.46-*	*59.16*	*69.18*	*65.29+*
Colon	*84.52*	*83.79-*	*85.43-*	*85.19-*	*82.0*	***87.26-***	*84.79-*
SRBCT	*99.63*	*98.57-*	*99.04-*	*99.18-*	*99.29+*	***99.63***	*99.4-*
Leukaemia	***98.04***	*96.52-*	*96.21-*	*96.53-*	*95.52-*	*96.39-*	*96.64-*
Ovarian	*99.96*	*99.96*	***100.0***	*99.33-*	*99.17-*	***100.0***	*99.96*
Average	*92.735*	*91.643*	*92.532*	*91.93*	*90.309*	***92.887***	*92.308*
St. Dv.	*8.46*	*11.23*	*9.60*	*9.16*	*11.53*	*8.76*	*9.95*
W/T/L		*1/3/6*	*0/4/6*	*1/3/6*	*1/6/3*	*1/5/4*	*2/3/5*
The accuracy results of the ALL method from Table 4.4							
Average	*92.735*	*92.5835*	*93.36*	*92.751*	*90.7045*	***93.401***	*92.4924*
St. Dv.	*8.46*	*11.78*	*9.12*	*8.82*	*11.54*	*8.62*	*10.88*

One different phenomenon observed is that SVM models trained with the full feature set performed not as badly as with the other two types of models (NB and KNN) and even gave the highest accuracy on five datasets. However, the SVMs using the full set of features were less efficient than the SVMs using fewer features, therefore HEF is still beneficial with SVM as classifiers.

5.5.1.3. Stability Evaluation

In this chapter, in addition to accuracy, we measured the stability of FS because by using the PART method, each fold of the FS method may produce a different feature subset, so we need to know which FS method is more stable to changes in the samples. On the other hand, for the ALL method (in Chapter 4), we did not need to measure the stability of FS because we had not included FS inside the cross-validation loop, as it always uses all the samples in the dataset before the classification phase; also, each run with differently shuffled random seeds of FS produces identical results.

Table 5.8 shows how each filter, as well as the two ensemble types (HEF and HEF-R1), have different stability in the same dataset; thus, it is apparent that some filters are more stable than others when the number of sample changes. As we can see, ReliefF has a higher average stability for all the datasets, and after that, Gain Ratio scored 0.73, which indicates that rank filters are more stable in changing samples than other FS methods. In contrast, the subset filters (FCBF and CFS) were unstable in the face of changes in the samples, while HEF and HEF-R1 scored in between the rank and subset filters. This proves that the ensemble method improves the level of stability, even if some of the members are relatively unstable. Also, FS methods are more stable in some datasets than in others, based on certain factors such as number of samples, number of features and number of class labels. As we can see, FS on microarray datasets is less stable than on other dataset types, as the number of features tends to be high and the number of samples very low. Also, FS with the M-feat-factor and Arrhythmia datasets is less stable than the first four datasets, because the numbers of class labels are higher, equal to 10 and 13, respectively.

Table 5.8: The stability measures of ATI with the features selected by filters and heuristic ensembles over 10 runs of 10-fold cross-validation

ATI	*FCBC*	*CFS*	*ReliefF*	*Gain Ratio*	*HEF*	*HEF-R1*
Zoo	***0.96***	*0.94*	*0.91*	*0.91*	*0.93*	*0.94*
Dermatology	*0.81*	*0.92*	*0.93*	***0.97***	*0.94*	*0.96*
Promoters	*0.75*	*0.75*	*0.75*	***0.81***	*0.71*	*0.74*
Splice	*0.76*	*0.76*	*0.91*	***0.94***	*0.8*	*0.82*
M-feat-factor	*0.64*	*0.7*	***0.89***	*0.75*	*0.8*	*0.78*
Arrhythmia	*0.43*	*0.56*	***0.77***	*0.72*	*0.7*	*0.52*
Colon	*0.28*	*0.36*	***0.66***	*0.41*	*0.46*	*0.4*
SRBCT	*0.36*	*0.44*	***0.66***	*0.61*	*0.57*	*0.5*
Leukaemia	*0.22*	*0.26*	***0.61***	*0.55*	*0.44*	*0.32*
Ovarian	*0.29*	*0.34*	***0.76***	*0.7*	*0.5*	*0.51*
Average	*0.55*	*0.60*	***0.78***	*0.73*	*0.68*	*0.65*
St. Dv.	*0.25*	*0.23*	***0.11***	*0.17*	*0.17*	*0.21*

The results in Table 5.9 presents the detailed stability measures for CWrel with the features selected by filters and heuristic ensembles over 10 folds of 10 runs. Similar patterns to those that appeared in Table 5.8 can again be observed. Again, the rank filters are demonstrably more stable than the subset filters, while HEF and HEF-R1 scored in the middle (i.e., between the rank and subset filters).

Table 5.9: The stability measures of CWrel with the features selected by filters and heuristic ensembles over 10 runs of 10-fold cross-validation

CWrel	*FCBC*	*CFS*	*ReliefF*	*Gain Ratio*	*HEF*	*HEF-R1*
Zoo	***1.0***	*0.94*	*0.9*	*0.9*	*0.94*	*0.95*
Dermatology	*0.83*	*0.92*	*0.93*	***0.98***	*0.85*	*0.97*
Promoters	*0.85*	*0.85*	*0.85*	***0.9***	*0.81*	*0.83*
Splice	*0.81*	*0.81*	*0.94*	***0.96***	*0.82*	*0.85*
M-feat-factor	*0.75*	*0.8*	***0.93***	*0.84*	*0.83*	*0.85*
Arrhythmia	*0.56*	*0.71*	***0.87***	*0.84*	*0.8*	*0.67*
Colon	*0.39*	*0.5*	***0.79***	*0.56*	*0.62*	*0.55*
SRBCT	*0.53*	*0.61*	*0.79*	*0.76*	***0.81***	*0.66*
Leukaemia	*0.34*	*0.41*	***0.75***	*0.71*	*0.65*	*0.52*
Ovarian	*0.43*	*0.49*	***0.86***	*0.82*	*0.66*	*0.66*
Average	*0.65*	*0.70*	***0.86***	*0.83*	*0.78*	*0.75*
St. Dv.	*0.21*	*0.18*	***0.06***	*0.12*	*0.09*	*0.15*

However, in order to comprehend the reasons for the differences in the classification accuracy levels between the PART and ALL methods, in other words, why the classifier results with the PART method are worse than with the ALL method, we measure the similarity of the FS results between the PART and ALL methods by using IATI and ICW, as described in (Somol and Novovicova, 2010). These similarity measures will give us some indication about how far the features selected by the ALL method are different in terms of number and actual features relative to the PART methods in each fold and in each run.

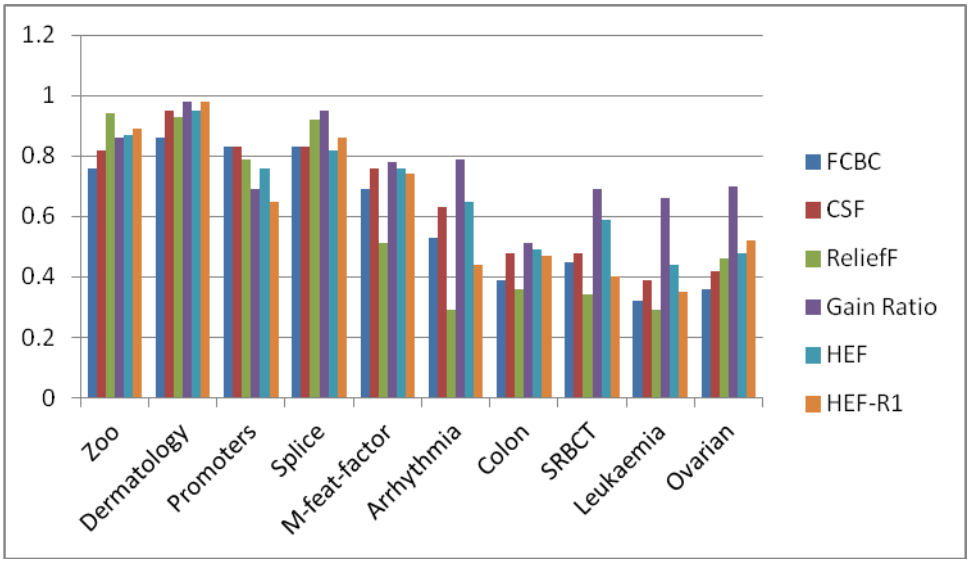


Figure 5.5: The similarity measures of IATI with the features selected by the filters, comparing the PART with the ALL approaches

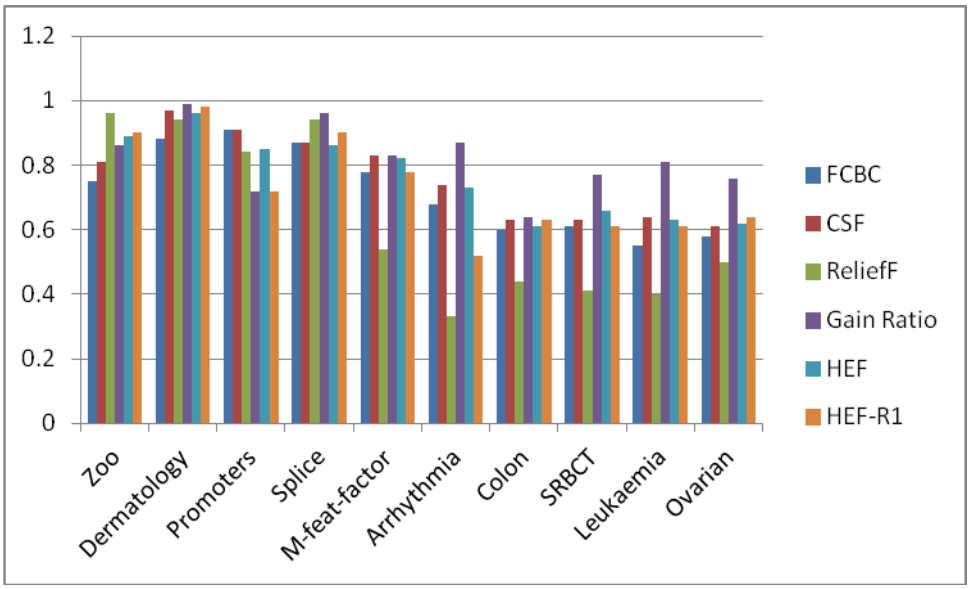


Figure 5.6: The similarity measures of ICW with the features selected by the filters, comparing the PART with the ALL approaches

Figures 5.5 and 5.6 show the similarity measures of IATI and ICW with the features selected by the 4 filters and 2 ensemble, comparing the PART and ALL approaches, which on average scored 0.60, 0.66, 0.58, 0.76, 0.681 and 0.63 respectively with IATI and 0.721, 0.764, 0.63, 0.821, 0.763 and 0.729 respectively with ICW.

In the light of the results shown in these two figures, the best method, according to the similarity comparison between the features selected by the PART method and the features selected by the ALL method, is Gain Ratio, although Gain Ratio and ReliefF always select an equal number based on the heuristic rule described in Chapter 4. This observation indicates that Gain Ratio is more stable than ReliefF to any decrease in the number of samples, while ReliefF is more influenced by sample size than the other FS methods. Furthermore, the number of features selected by the HEF methods has the highest change over 10 runs of 10-fold cross-validation; nevertheless, the level of similarity between the PART and ALL methods is higher than with the other FS methods, except Gain Ratio. Then, the level of similarity between the PART and ALL methods decreases in the following order: CFS, HEF-R1, FCBF and finally ReliefF.

Additionally, the similarity between the PART and ALL approaches is affected by the type of dataset. As we can see, the last 6 datasets have less similarity between the PART and ALL approaches than the first four datasets, on average. This is because they are microarray datasets with a quite high numbers of features and very small sample numbers. Also, the M-feat-factor and Arrhythmia datasets have less similarity than the first four datasets and this may be because the numbers of class labels are high (10 with the M-feat-factor and 13 with Arrhythmia), which is similar to Li's findings: "The study suggests that multi-class classification problems are more difficult than binary ones in general." (Li et al., 2004).

However, the similarity measure with these real-world datasets can only indicate the extent of similarity between the ALL and PART approaches; it cannot tell which one is better when they are dissimilar. Thus, we evaluated how effective they are by measuring their average classification accuracy in Tables 5.5-5.7.

Moreover, we are interested in this section to understanding the relationship between the level of similarity vis-à-vis PART and ALL, and the level of changes in classification accuracies between them.

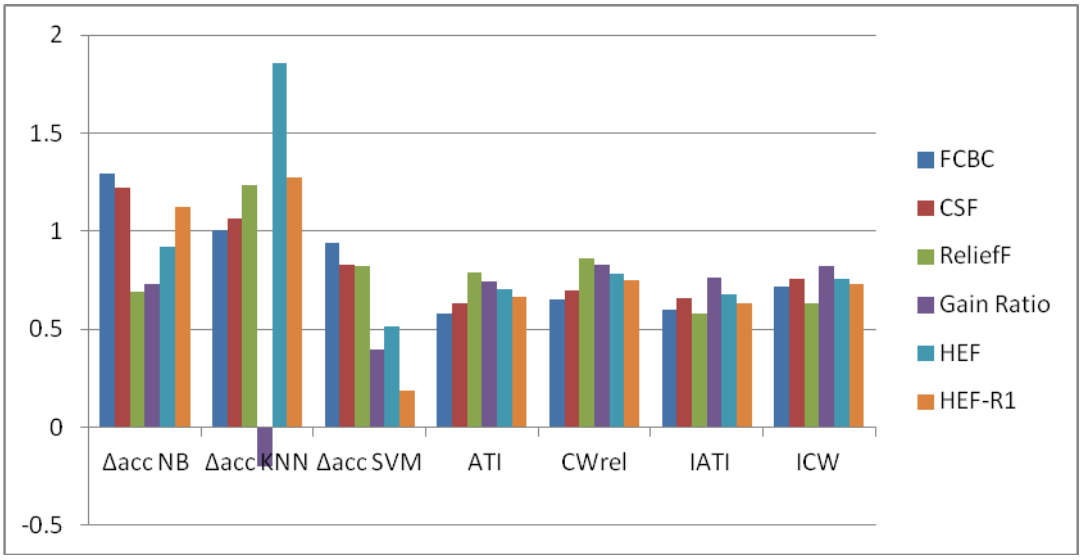


Figure 5.7[4] **:** The difference (Δacc) between the average accuracies of the three classifiers trained by the ALL and PART approaches as well as the averages of similarity measures

In the light of the results shown in Figure 5.7, the highest method according to the level of similarity between the features selected by PART and ALL (IATI& ICW), and the lowest difference in terms of accuracy among the three classifiers relative to PART and ALL, is Gain Ratio, although Gain Ratio has the lowest classification accuracy among the FS methods used in this experiment. ReliefF is the most stable method in terms of changes in sample with the same size as with the PART method (ATI & CWrel), but has the lowest similarity in terms of the selected features when comparing the PART and ALL methods (IATI& ICW). This means that ReliefF is more influenced by changes in sample size than the other FS methods. In contrast, the subset filters (FCBF & CFS) were less stable in the face of changes in the samples (ATI & CWrel) and delivered less similarity in the features selected in comparing PART and ALL (IATI & ICW). HEF and HEF-R1 scored in between the rank and subset filters, however, HEF has the highest classification accuracy among the FS methods used in this experiment. Moreover, in terms of classifiers, SVM is less subject to change in comparing the PART and ALL methods among the other classifiers over all the FS methods.

In the next section, we will apply the experiment on the generated synthetic dataset in which we know the relevant features in advance; this should help us to answer the above questions clearly.

[4] Δacc = acc (ALL) – acc (PART), represents the difference between the average accuracies of the three classifiers trained by the ALL and PART methods.

5.5.2 Results on Synthetic Datasets

In this section, the results after applying four filters and two heuristic ensembles over 21 synthetic datasets and 4 bench mark synthetic datasets will be presented, grouped in different families that deal with various situations. The first group presents different irrelevant features, the second group presents different samples, the third group presents different numbers of relevant features, the fourth group present the different class noise injections, and last group presents the bench mark synthetic datasets. The behaviour of the FS method will be evaluated according to the classification accuracy obtained by the NB classifier, the similarity with the optimal set with the PART and ALL methods, and the stability with the PART method.

5.5.2.1. Accuracy Evaluation

It is important, as a common practice in the literature, to see the average classification accuracy obtained in a 10-fold cross-validation of 10 runs, as described in Section 5.4.2. In order to see whether or not the cross-validation on the PART method has any influence, we can compare the accuracy with the ALL method.

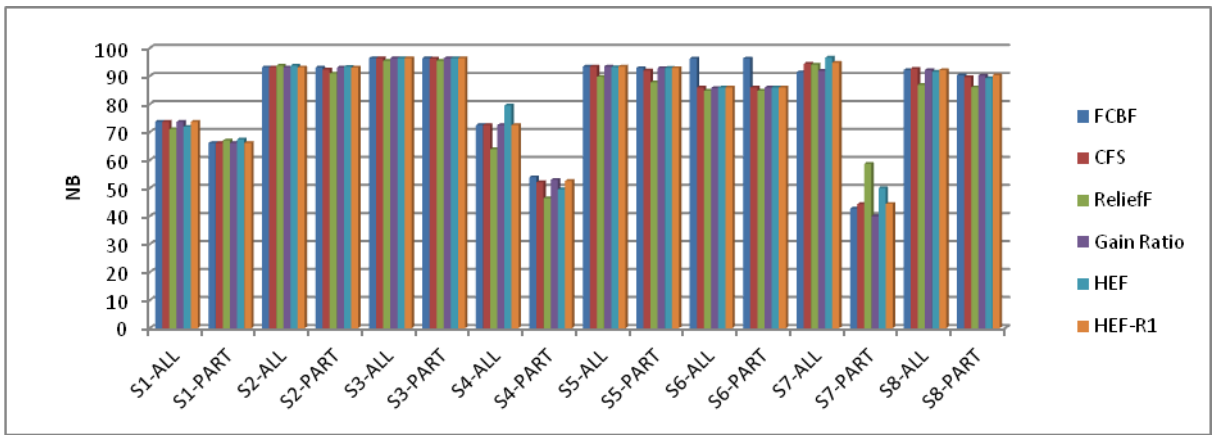


Figure 5.8: Accuracy of NB classifier obtained for S1 to S8 datasets with both methods

Figure 5.8 shows the accuracy of the NB classifier obtained for the 8 datasets with both methods. The best classification accuracy was obtained by S3-PART as well as S3-ALL, which has the highest similarity, as this dataset has the smallest number of irrelevant features (90) and the highest number of samples (10,000) without any difference between the PART and ALL methods. On the other hand, S7-PART has the worst classification accuracy as well as the lowest similarity, as this dataset has the

smallest number of samples (100) and the highest number of irrelevant features (9,990). Among these two datasets, we can see various classification accuracy results, varying based on two factors in general: the number of samples and the number of irrelevant features. In addition to that, the diversity between the PART and ALL methods becomes clear on the datasets with small samples (such as S1, S4 and S7). It is clear that the ALL method has a higher accuracy than the PART method. S7-ALL in particular greatly outperforms S7-PART by (47.2) in terms of accuracy, while both methods give similar similarity; this case simulates the problem of microarray datasets, which have high dimensionality with small numbers of samples. On the other hand, the PART and ALL methods obtained similar accuracy on the remaining datasets, which have medium or high numbers of samples.

Also, we can see that FCBF in S6-ALL and S6-PART outperforms the other filters; this may be explained by the irrelevant features (randomly generated), possibly adding some useful information by chance to the classifier, while the disturbed relevant features are not so informative.

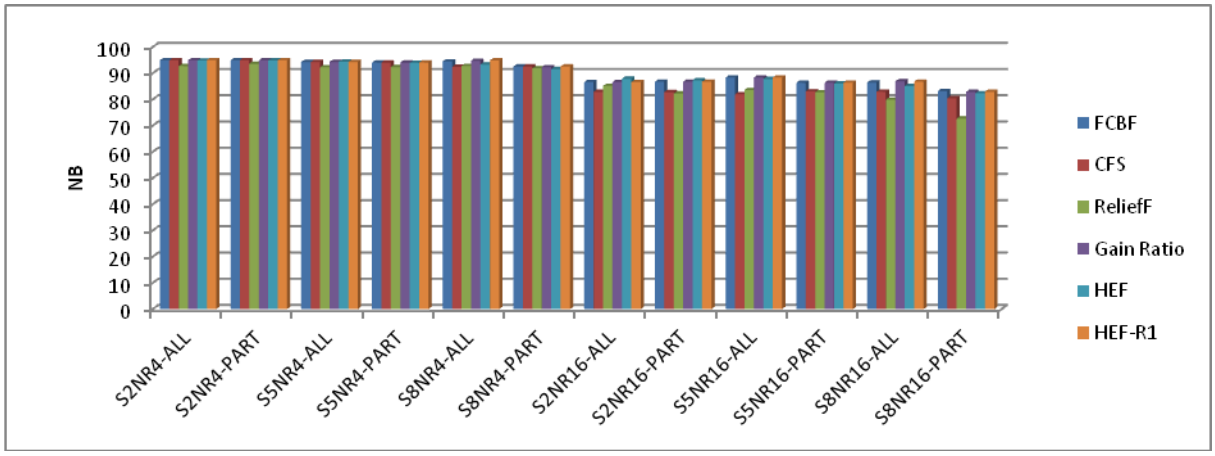


Figure 5.9: Accuracy of NB classifier of the S2NR4 to S8NR16 datasets with both methods

Figure 5.9 shows the accuracy of the NB classifier obtained for the 6 datasets from two groups: the first group consists of 1,000 samples, four relevant features and different numbers of irrelevant features (96, 996 and 9,996, respectively), and the second group consists of the same number of samples but with 16 relevant features and different numbers of irrelevant features (84, 984 and 9,984, respectively). The first group has almost the same accuracy with the PART and ALL methods, except S8NR4-PART which was reduced by 2.2, and which has slightly less accuracy in all FS methods. The

accuracy in the first group (NR4) is between 94.97 and 92.44, while the second group (NR16) has a considerable decrease in accuracy of between 87.32 and 82.37, except ReliefF with S8NR16, which has 79.88 with the ALL method and 72.71 with the PART method. In brief, the first group has higher accuracy than the second group, due to the number of optimal features being small, and more importantly their corresponding coefficient values are higher, which enables the FS method to select these features. It was difficult for the FS method to select all the optimal features in the second group because it has quite a high number of relevant features; also, some of these features have low corresponding coefficient values, making it difficult to determinate the class label. Also, the second group has lower accuracy with the PART method than with the ALL method, especially with S8NR16-PART decreasing by 3.2 in terms of accuracy.

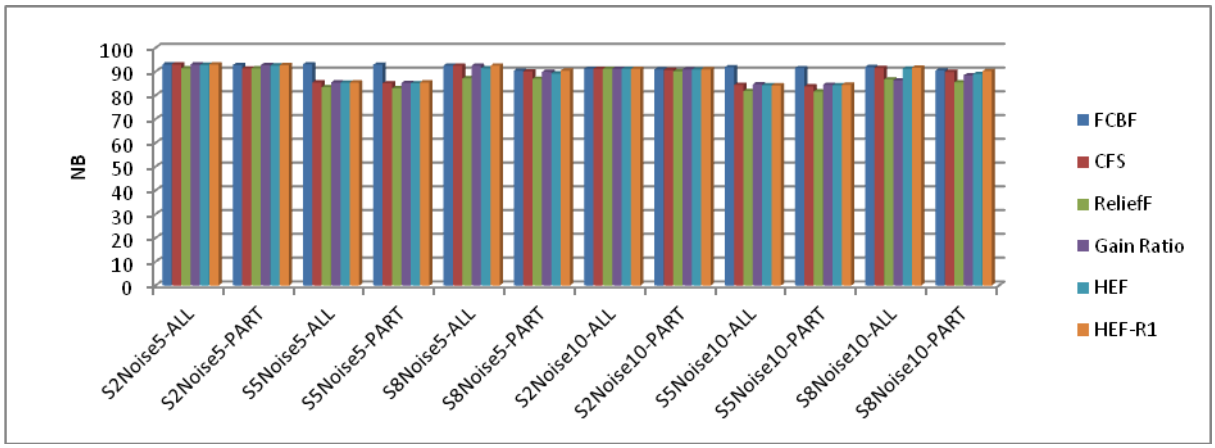


Figure 5.10: Accuracies of NB classifier of the S2Noise5 to S8Noise10 datasets with both methods

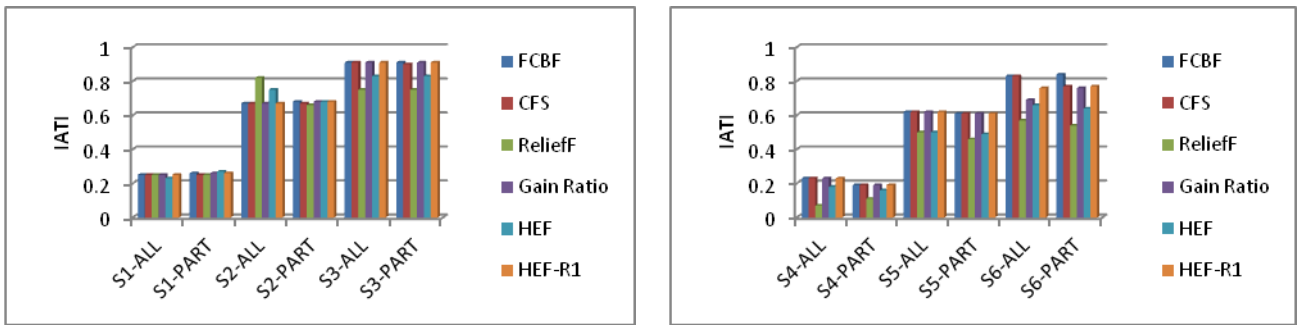
Figure 5.10 shows the accuracies of the NB classifier obtained for the 6 datasets from 2 groups: the first group consists of 1,000 samples, 10 relevant features and different numbers of irrelevant features (96, 996 and 9,996, respectively) and 5% injected class noise, and the second group consists of the same number of samples and relevant features, the only difference being in the degree of noise, which increases to 10%, as in Section 5.4.1.2.

The above figure shows a slight decrease in accuracy when increasing the noise rate. For example, S2Noise5-ALL has 93.04 with most of the filters, while S2Noise10-ALL has 91.19 with all the filters, and all the others in the second group have less accuracy than the first group due to the increase in the noise level. Also, these three datasets (S2, S5, S8), without adding any noise (as we can see in Figure 5.8), have higher accuracy

than with the addition of 5% or 10% noise. Furthermore, the ALL method has slightly higher accuracy than the PART method by about 1% in most of the datasets. In addition, FCBF has very high accuracy relative to the other FS methods; this may be explained by the irrelevant features (randomly generated) adding some information useful to the classifier, while the disturbed relevant features are not so informative.

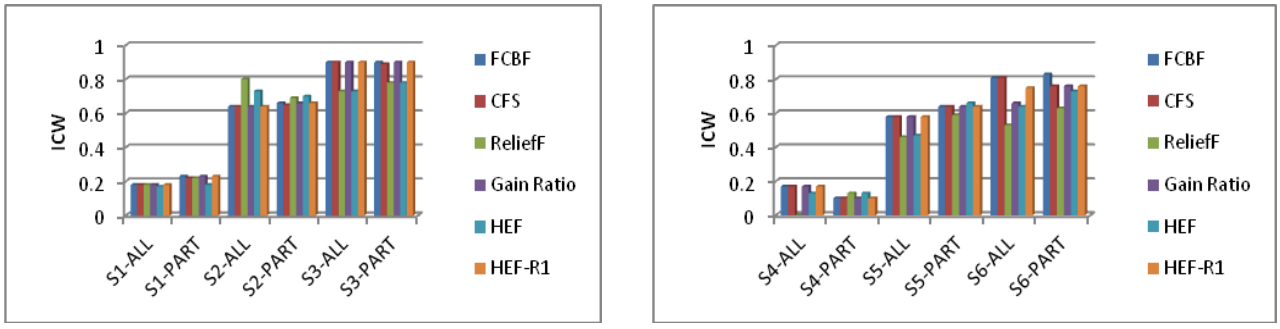
5.5.2.2. Stability Evaluation

A) Dealing with Different Numbers of Samples



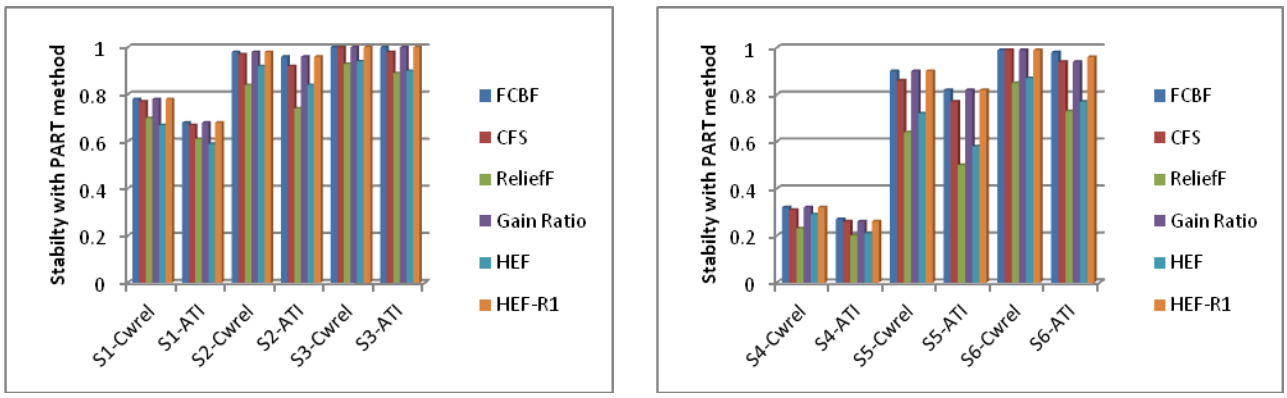
(a) ***(b)***

Figure 5.11: IATI comparison between each filter subset with optimal subset on: (a) S1, S2 and S3 (b) S4, S5 and S6



(a) ***(b)***

Figure 5.12: ICW comparison between each filter subset with optimal subset on: (a) S1, S2 and S3 (b) S4, S5 and S6



(a) ***(b)***

Figure 5.13: Comparing feature selector's stability (CWrel, ATI) with the PART method on: (a) S1, S2 and S3 (b) S4, S5 and S6

Figure 5.11 (a) shows the results of the IATI measure over the datasets consisting of 100 features; 10 of them are relevant while the remaining are irrelevant, and they have different numbers of samples (100, 1,000 and 10,000, respectively). For S1-ALL, none of the FS methods used in this study were able to select high numbers of relevant features; they just selected the most relevant features (X_{10} and X_9) and one irrelevant feature, which led to low similarity (0.25) when compared with the optimal set.

Similarly, for S1-PART, none of the FS methods were able to select high numbers of relevant features; they just selected the most relevant features in addition to one or more irrelevant features, and these subsets can be diverse in each fold.

However, with S3-ALL and S3-PART, the FS methods were able to select (approximately) the optimal set without any irrelevant features; as we can see, there are very high similarity values (0.9), except for ReliefF (0.75) and accordingly HEF (0.83).

With S2-ALL, the results are acceptable; they are better than S1 but worse than S3, and the similarity is 0.67 with FCBF, CFS, Gain Ratio and HEF-R1 when compared with the optimal set. On the other hand, ReliefF selected 8 relevant features without any irrelevant features, so the similarity is 0.82, which is higher than the others. Also, HEF scored 0.75 because it has 8 relevant features with only one irrelevant feature. With S2-PART, the FS methods were able to select on an average 7 of the relevant features, with similarity equal to 0.67.

Figure 5.11 (b) shows the results of the IATI measure over the datasets consisting of 1,000 features; 10 of them are relevant while the remaining are irrelevant, and there are 100, 1,000 and 10,000 samples, respectively. In fact, S4, S5 and S6 are equal to the previous datasets (S1, S2 and S3) in all variables except for the number of irrelevant features, which is increased by 10 times. As we can see from Figure 5.11(b), the experiment produced almost the same patterns as in Figure 5.11 (a) but with decreasing similarity among all the datasets. Also, we can see that S4-PART has less similarity than S4-ALL by 0.04, which was not the case with S1, due to the increase in irrelevant features (leading to an increase in the diversity between each fold).

Also, Figures 5.12(a) and (b) show that the results of the ICW measure produce patterns equivalent to Figures 5.11(a) and (b). Among these four figures, we can note that the number of samples plays the most important role. As we can observe, if the number of samples is high, it helps all the FS models to select a high proportional number of relevant features, while if the number of samples is low, it will be hard for all the FS models to select a high number of relevant features – usually less than 20%. In addition, we notice an increasing tendency to select irrelevant features, and increasing the number of irrelevant features in the dataset plays a significant role in disrupting the process of feature selection, and in increasing the chance of choosing irrelevant features.

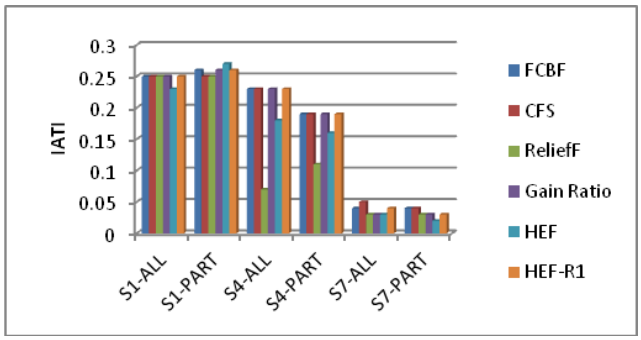
Among the above four figures, we cannot see a large difference between the PART and ALL methods with these results, which are the average of 10 runs; they are relatively similar, except for those datasets with small numbers of samples (like S1 and S4). The PART method with S4 (100 samples and 990 irrelevant features) has to some extent smaller values in some runs (on average) than the ALL method, and we will focus on this case in the following section.

Figures 5.13 (a) and (b) show the results of the CWrel and ATI measures over the PART method on S1, S2, S3 and S4, S5, S6, respectively. We can clearly notice the large difference in stability between Figures 5.13(a) and (b) due to the increase in the number of irrelevant features in Figure 5.13(b). Also, among each figure the level of stability increases with the increasing sample numbers. When we compare the values produced by the feature-focused (CWrel) with the subset-focused (ATI) measures, we can see a relative (steady) increase in CWrel (more so than in ATI) in both figures.

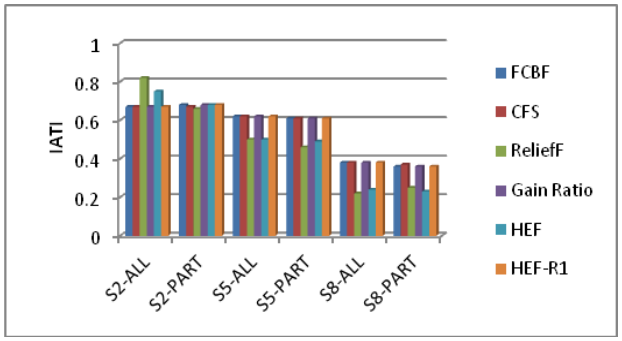
B) Dealing with Different Numbers of Irrelevant Features

Figure 5.14 (a) shows the results of the IATI measure over the datasets consisting of 100 samples, 10 relevant features and different numbers of irrelevant features (90, 990 and 9,990, respectively). All these three datasets produce very low similarity in both methods, especially S7 which selected a high number of irrelevant feature and just two relevant ones, due to the small number of samples compared with the number of irrelevant features. In addition, S4-ALL has a slightly higher value than S4-PART (by about 0.04), except for ReliefF filters, which failed to select any relevant features. S1 and S7 have similar results for both methods.

Figure 5.14 (b) shows the results of the IATI measure over the datasets consisting of 1,000 samples, 10 relevant features and different numbers of irrelevant features (90, 990 and 9,990, respectively). Because the samples are increased in this figure, we can also observe increases in the similarity (up to 0.67). Among these three datasets, S2 has (on an average) acceptable similarity, as does S5 to a lesser extent; S8 has less similarity due to the high number of irrelevant features included in this dataset, which led all the FS models to select high numbers of irrelevant features as well as relevant ones. In addition, there are no considerable differences between PART and ALL in these two figures.

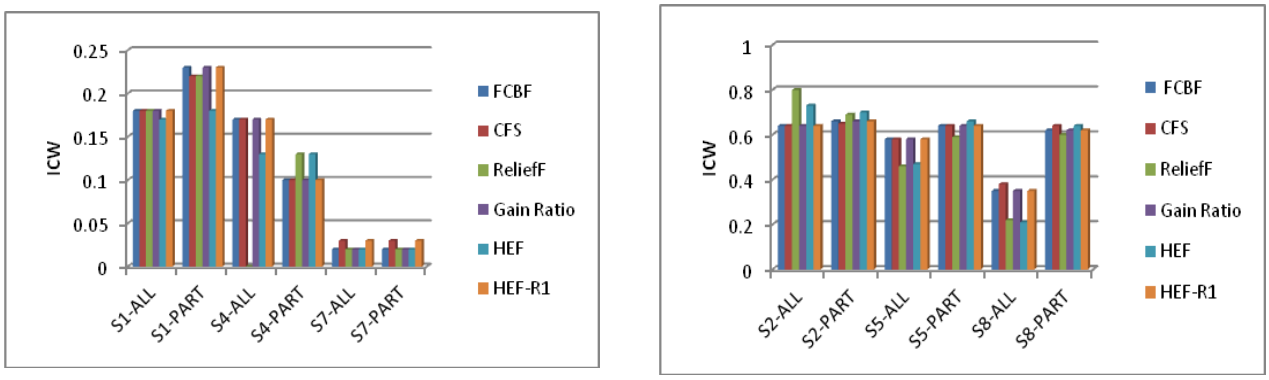


(a)



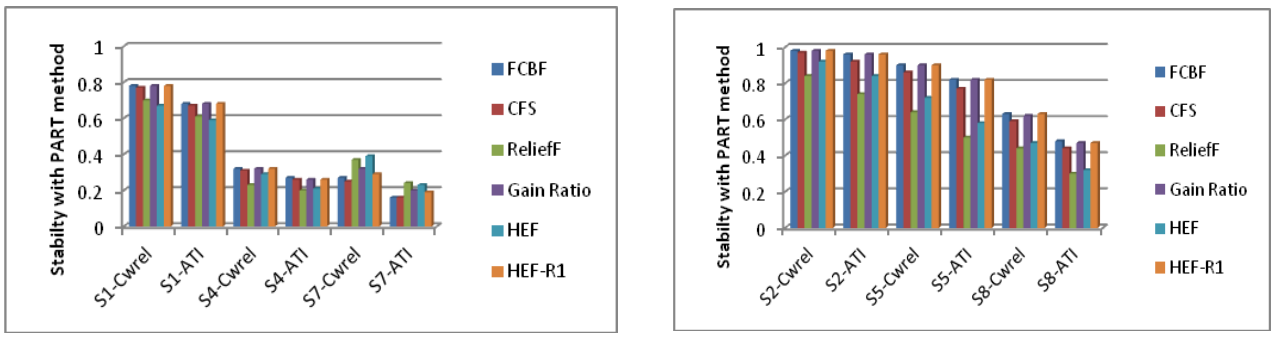
(b)

Figure 5.14: IATI comparison between each filter subset with optimal subset on: (a) S1, S4 and S7 (b) S2, S5 and S8



(a) ***(b)***

Figure 5.15: ICW comparison between each filters subset with optimal subset on: (a) S1, S4 and S7 (b) S2, S5 and S8



(a) ***(b)***

Figure 5.16: Comparing feature selector's stability (CWrel, ATI) with the PART method on: (a) S1, S4 and S7 (b) S2, S5 and S8

Figures 5.15(a) and (b) show that the results of the ICW measure produce patterns that are equivalent to Figures 5.14(a) and (b), except for the ALL method on S5 and S8, which yielded lower ICW values than did the PART method. The possible reason could be that the subset features selected by the PART method changed in some folds and do not change with the ALL, which led to an increase in the frequency to a greater number of features.

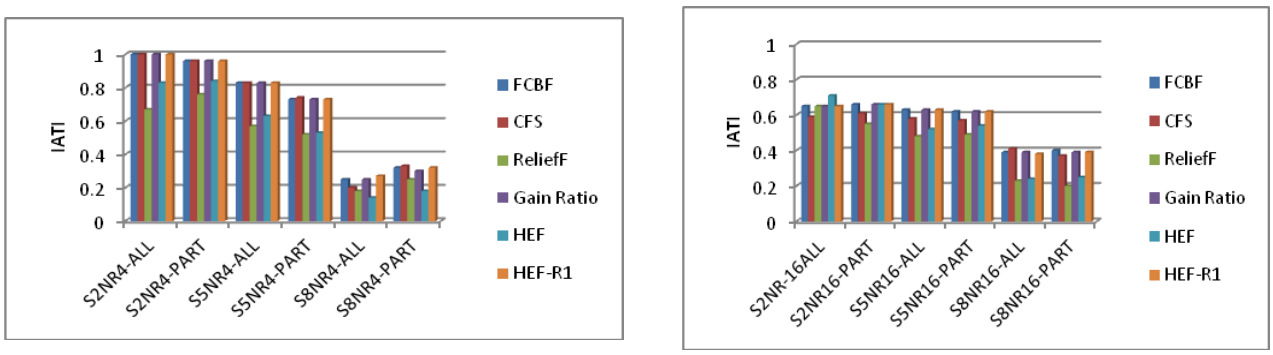
Figure 5.16 (a) shows the results of the CWrel and ATI measures over the PART method on S1, S4 and S7; we can clearly notice that increasing irrelevant features causes a decrease in stability. Conversely, Figure. 5.16(b) illustrates a higher stability on S2, S5 and S8 because of the increase in the samples from 100 in Figure 5.16(a) to 1,000 in Figure 5.16(b). Also, the decline in the stability is not as sharp as in (a) even when the irrelevant feature number increases, as in S8, due to the sample size. Also, we can notice that the ATI value is slightly less than for CWrel in all datasets.

To sum up, there are no clear differences between the PART and ALL methods; however, the PART method with a low number of samples has low stability, especially with increases in the number of irrelevant features, as shown in Figure 5.16(a). In addition, the number of samples is the primary factor playing a role in the performance of FS, and after that, it is the number of irrelevant features. Also, among these figures, we can observe that ReliefF is an unstable filter in both the PART and ALL methods, even when we average the results of 10 runs in both methods.

C) Dealing with Different Numbers and Importance of Relevant Features

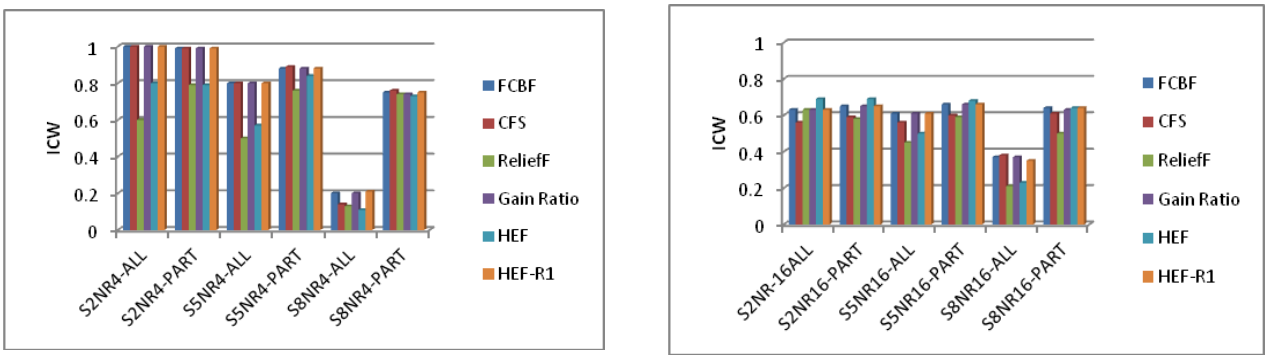
In this section we will investigate the influence of relevant feature numbers and their weights (corresponding coefficient values) on FS and also on the evaluation methods. Therefore, we applied the dataset generated in Section 5.4, which has 4 relevant features with high importance (high corresponding coefficient value) in the first group and 16 relevant features with low importance (low corresponding coefficient value) in the second group, on the PART and ALL methods.

Figure 5.17 (a) shows the results of the IATI measure over the datasets consisting of 1,000 samples, 4 relevant features and different numbers of irrelevant features (96, 996 and 9,996, respectively). Clearly, we can see how much the irrelevant features can decrease the similarity between the optimal features and the features selected in each FS model. S2NR4 has high similarity (up to 1) in almost all the methods, which successfully selected all the relevant features without any irrelevant features, except for ReliefF (and accordingly HEF) because ReliefF missed one relevant feature and included instead one irrelevant feature, which led to a decrease in the similarity.



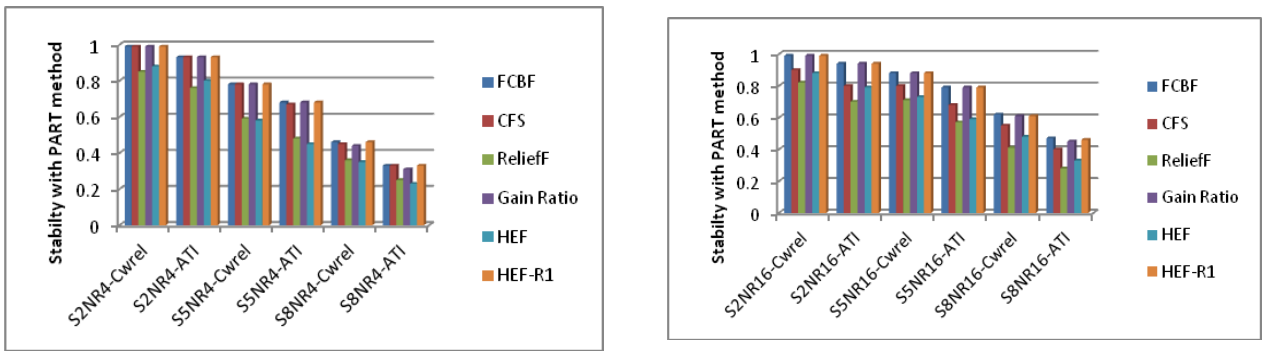
(a) ***(b)***

Figure 5.17: IATI comparison between each filter subset with optimal subset on: (a) S2NR4, S5NR4 and S8NR4 (b) S2NR16, S5NR16 and S8NR16



(a) ***(b)***

Figure 5.18: ICW comparison between each filter subset with optimal subset on: (a) S2NR4, S5NR4 and S8NR4 (b) S2NR16, S5NR16 and S8NR16



(a) ***(b)***

Figure 5. 19: Comparing feature selector stability (CWrel, ATI) with PART method on: (a) S2NR4, S5NR4 and S8NR4 (b) S2NR16, S5NR16 and S8NR16

S5NR4 has lower similarity in all methods due to an increase in the irrelevant features, which led the FS method to include one irrelevant feature in addition to all the relevant features, except for ReliefF (and accordingly HEF) because ReliefF missed one relevant feature and included two irrelevant features instead.

However, S8NR4 has huge dissimilarity because all the methods included relatively high numbers of irrelevant features (between 10 and 24) in addition to 4 relevant features. On the other hand, Figure 5.17(b) shows the results of the IATI measure over the datasets consisting of 1,000 samples, 16 relevant features and different numbers of irrelevant features (84, 984 and 9,984, respectively). All the methods in these datasets failed to select all the relevant features; they selected only the most relevant features (between X_{16} and X_7). FS methods select the highest number of relevant features without any irrelevant ones with S2NR16, also FS selected with S5NR16 same as S2NR16 in addition to two irrelevant features, while FS included a high number of

irrelevant features (between 10 and 32) in addition to the highest number of relevant features with S8NR16.

From these two figures, we can conclude that the FS method is able to select the highest relevant features but it becomes a challenge when the feature is less relevant. Also, we can see small differences between PART and ALL in Figure 5.17(a) but there are no clear differences between PART and ALL in Figure 5.17(b). The small decreases with the PART method relative to ALL in Figure 5.17(a), (which are equal to -0.04, -0.1 and 0.06, respectively) are due to the decreases in stability, as we can see in Figure 5.19(a). Also, the second possible reason could be because any missing relevant features or any addition of irrelevant features in any of the folds in the PART method will affect the similarity, especially because all 4 optimal features are highly relevance and the number of optimal features is small.

Figures 5.18 (a) and (b) show the results of the ICW measure, which produced patterns that are different to Figures 5.17 (a) and (b); this is especially the case with S8NR4-PART and S8NR16-PART due to the difference in the measuring mechanism, as IATI evaluates similarities between selected subset features and optimal features (see equation 8.14), while ICW evaluates feature selection frequencies over all subset features considered together as whole and optimal features (see equation 5.12). So, for this reason, both S8NR4-PART and S8NR16-PART have high total frequencies of irrelevant features, which have a high possibility of changing with the different folds. Accordingly, both S8NR4-PART and S8NR16-PART have high ICW values, while both S8NR4-ALL and S8NR16-ALL have low ICW values because irrelevant features do not change with the different runs, and so the total frequency of selected features was not as high as with the PART method.

Figure 5.19 (a) shows the results of the CWrel and ATI measures over the PART method on S2NR4, S5NR4 and S8NR4; we can clearly notice that increasing the irrelevant features causes a decrease in stability. Also, Figure 5.19(b) illustrates the same pattern with S2NR16, S5NR16 and S8NR16.

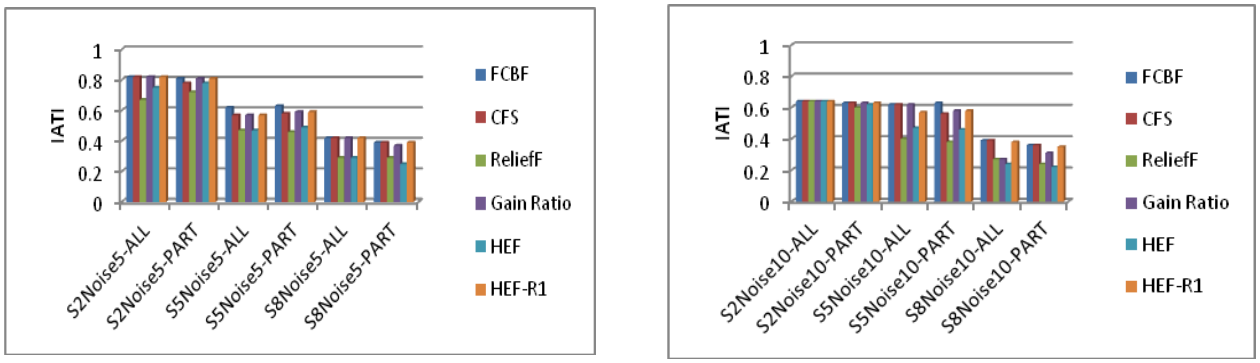
To sum up, the differences between the PART and ALL methods on the datasets with $N_R = 4$ are more apparent than on the datasets with $N_R = 16$ because the number of optimal features is small, so any missing features will affect the similarity. Moreover, the FS methods are able to select the highest number of relevant features but it becomes

a challenge when the features have low or very weak relevance, as in the datasets with N_R =16. Also, increasing the irrelevant features led to decreasing the similarity between the optimal features and the features selected in each FS method; it also decreased the stability with the PART method. In addition, among these figures, we can observe that CFS has less similarity in the datasets with 16 optimal results because CFS missed more relevant features than FCBF and Gain Ratio. ReliefF has less similarity and stability in all the figures, with both the PART and ALL methods (and accordingly HEF).

D) Dealing with Different Class Noise Injection

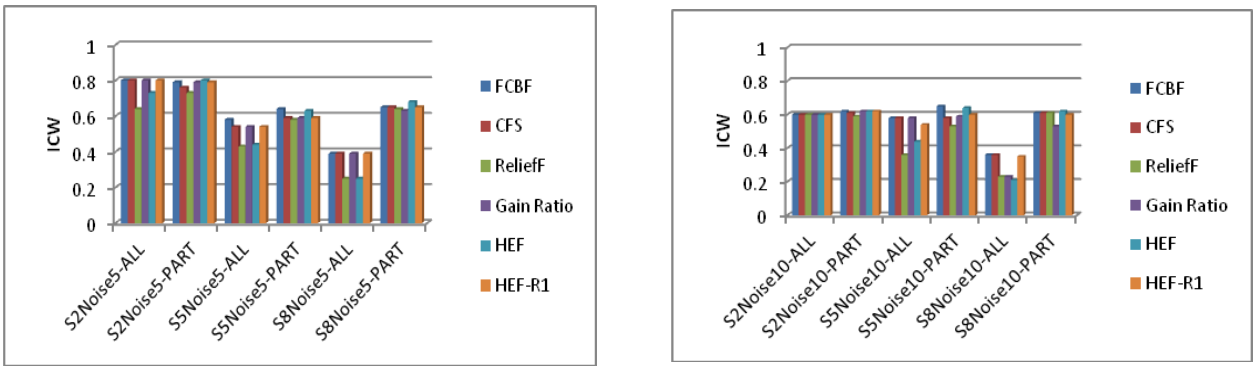
In this section we will investigate the influence of class noise injection on FS and also on the PART and ALL methods. Therefore, we applied the dataset generated in Section 5.4, which has 10 relevant features, with 5% class noise injected in the first group and 10% noise injected in the second group.

Figure 5.20 (a) shows the results of the IATI measure over the datasets consisting of 1,000 samples, 10 relevant features and different numbers of irrelevant features (90, 990 and 9,990, respectively) with 5% class noise injected. We cannot see clear differences between the PART and ALL methods in this figure, which averages over 10 runs; they are relatively similar. Figure 5.20 (b) shows the results after raising the noise level to 10%, which slightly decreases the similarity of the PART method (more so than the ALL method). In addition, it is worth noting how the similarity decreases in S2Noise5 and S2Noise10 with PART and ALL from 0.81 to 0.64, respectively, while there is almost no difference between S5 and S8 with 10% noise than with 5% noise. Therefore, we can say that datasets with small numbers of irrelevant features (as S2) can easily be affected by noise (more so than datasets with high numbers of irrelevant features s). Also, we can see in S8Noise10 that the Gain Ratio filter is affected by noise more than the other filters.



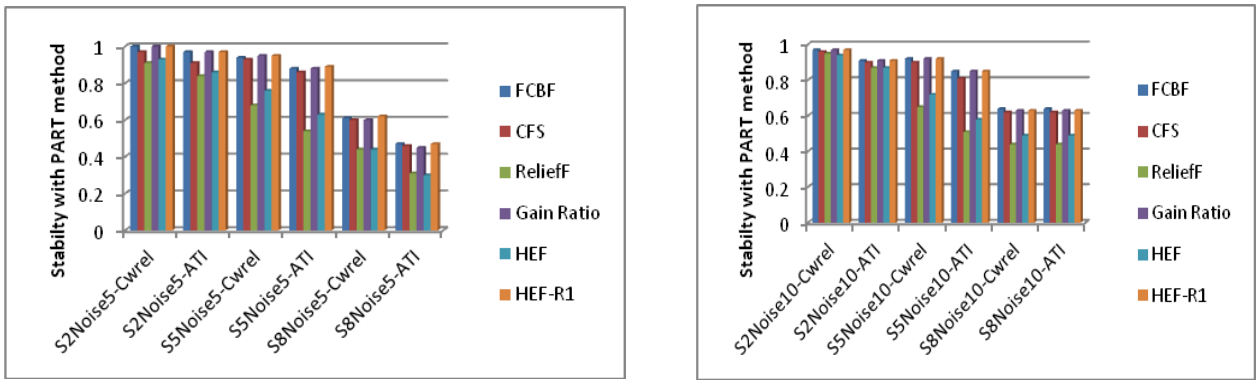
(a) ***(b)***

Figure 5.20: IATI comparison between each filter subset with optimal subset on: (a) S2Noise5, S5Noise5 and S8Noise5 (b) S2Noise10, S5Noise10 and S8Noise10



(a) ***(b)***

Figure 5.21: ICW comparison between each filter subset with optimal subset on: (a) S2Noise5, S5Noise5 and S8Noise5 (b) S2Noise10, S5Noise10 and S8Noise10



(a) ***(b)***

Figure 5.22: Comparing feature selector's stability (CWrel, ATI) with the PART method on: (a) S2Noise5, S5Noise5 and S8Noise5 (b) S2Noise10, S5Noise10 and S8Noise10

Figures 5.21(a) and (b) show that the results of the ICW measure produce patterns equivalent to Figures 5.20(a) and (b), except with the PART method on S5 and S8, which yielded higher ICW values than the ALL methods did. The possible reason could

be that the subset features selected by the PART method changed in some folds and do not change with ALL, which led to increasing the frequency to higher numbers of features.

Figure 5.22 (a) shows the results of the CWrel and ATI measures over the PART method with 5% class noise on S2, S5 and S8; we can clearly notice that increasing the number of irrelevant features causes a decrease in stability. Figure 5.22(b) illustrates a greater decrease in stability because of the increase in noise from 5% to 10%.

In brief, IATI produced a tiny difference between PART and ALL (around 0.02), which is not considered, while ICW has a notable increase with PART (relative to ALL) with S5 and S8. Also, datasets with a small number of irrelevant features (such as S2) can easily be affected by noise (more so than datasets with a high number of irrelevant features). Finally, increasing the noise led to decreasing the stability. Also, the Gain Ratio filter was affected by the increasing noise level to a greater extent than the other filters.

5.5.3 Experiment with Benchmark Synthetic Data

In order to generalise our experiments, we also used other synthetic datasets which are commonly used. The first dataset is Corral (John et al., 1994) which has 32 samples and 6 binary features (A_0 , A_1 , B_0, B_1 , I, R) and the class value is Y= ($A_0 \wedge A_1$) v ($B_0 \wedge B$). Features (A_0 , A_1 , B_0, B_1) are independent of each other, feature I is irrelevant to Y and feature R is correlated to class label by 75% and is redundant. The correct behaviour for a given FS method is to select the four relevant features and discard the irrelevant and correlated ones. The other three synthetic datasets are Monk1, Monk2 and Monk3 (Thrun et al., 1991) which have 6 binary features (A_1, A_2, A_3, A_4, A_5, A_6). Monk1: (A_1 = A_2) v (A_5=1) which has 124 samples, Monk2: exactly two of A_1=1, A_2=1, A_3=1, A_4=1, A_5=1, A_6=1 which has 169 samples and Monk3: (A_5=3 $\wedge$ A_4=1) v ($A_5 \neq 4 \wedge A_2 \neq 3$) which has 122 samples and 5% noise in the target.

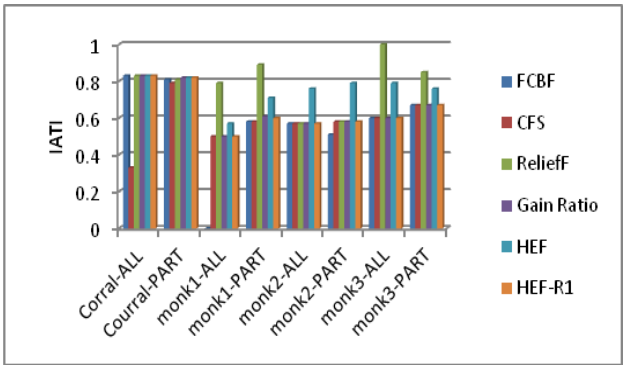


Figure 5.23: IATI comparison between each filter subset with optimal subset over synthetic data which were widely used.

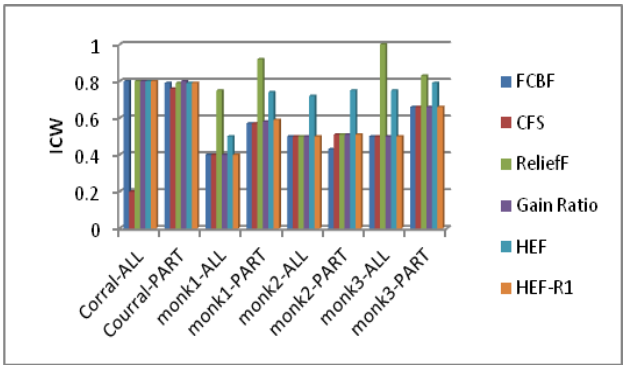


Figure 5.24: ICW comparison between each filter subset with optimal subset over synthetic data which were widely used**.**

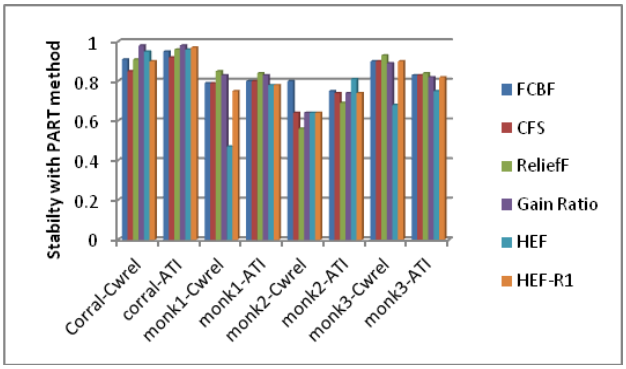


Figure 5.25: Comparing feature selection stability (CWrel, ATI) with the PART method.

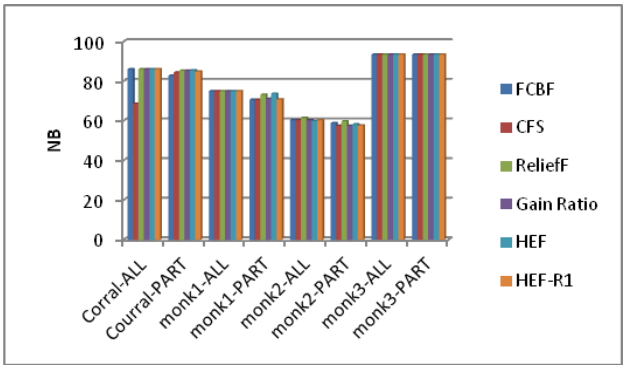


Figure 5.26: Accuracy of NB classifier over synthetic data widely used on both methods

Figure 5.23 shows the results of the IATI measure over these four datasets, we can see the PART method has slightly higher similarity than the ALL method except for the Corral dataset, but the clear difference was between FS methods; ReliefF and HEF outperform other filters. Also, Figure 5.24 shows that the results of the ICW measure produce patterns that are equivalent to Figure 5.23. The interesting thing about these datasets is the performance of ReliefF and HEF, which are able to select more relevant features than other filters, which is a different result to that of our synthetic data; ReliefF may work better than these filters on other problems such as interaction, noisy or non-linear problems. In addition to that, Figure 5.25 shows different stability levels between FS methods with different datasets. Finally, Figure 5.26 shows the accuracy of the NB classifier obtained for the 4 datasets with both methods. The best classification accuracy was obtained by Monk3-PART as well as Monk3-ALL, while the ALL method obtained higher accuracy on Corral, Monk1 and Monk2 than the PART method.

5.6 Discussion

5.6.1 Real-world Benchmark Datasets

The following summaries are about the differences between the PART and ALL methods in terms of classification accuracy and stability on 10 real datasets:

1- The number of selected features in the PART method is clearly changed in the different folds and runs, compared to the ALL methods as illustrated in Figures 5.3 and 5.4 and in Appendix A.

2- The level of accuracy achieved by using the PART method decreases more than with the ALL method on the three classifiers (on average) by -1.07, -1.03, -0.91, -0.31, -1.09 and -0.86, respectively. The degree of change in the accuracy between the PART and ALL methods differs from one classifier to another as well as from one FS to another.

3- The level of stability achieved by using the PART method differs from one FS to another in the same dataset. Rank filters are more stable than subset filters, while HEF and HEF-R1 scored in between the rank and subset filters, which proves that the ensemble method improves the level of stability, even if some of the members are relatively unstable.

4- The FS methods are more stable on some datasets than on others, based on certain factors such as number of samples, number of features and number of class labels.

5- The level of similarity between the PART and ALL methods differed from one FS to another on the same dataset. Gain Ratio with the PART method selected more similar features to the ALL method than the other FS methods did. Then, the level of similarity between the PART and ALL methods decreases in the following order: HEF, CFS, HEF-R1, FCBF and finally ReliefF.

6- The high level of stability and similarity in addition to the low level of difference in classification accuracy between the PART and ALL methods are not a strong indication of gaining highly accurate results, as we observed in Figure 5.7.

Although the above results demonstrate that the accuracy achieved by using the PART method is lower than by using the ALL method, and that the level of similarity between the PART and ALL methods differed from one FS to another, these results on the real-

world datasets do not give us a clear picture to determine which method provides less bias and is more reliable to use. Also, we do not know which method assists in selecting the more relevant features, as we applied the experiment on real data without knowing the most relevant features.

5.6.2 Synthetic Datasets

The following section sums up the differences between the PART and ALL methods in terms of classification accuracy and stability on 21 synthetic datasets generated:

1- When datasets have a high number of samples such as S3, S6 and S9 (S=10,000), the results demonstrate no difference between the PART and ALL methods in accuracy and similarity, and also both methods have high stability.

2- Datasets with a medium number of samples such as S2, S5 and S8 (S=1,000), demonstrate no difference in accuracy and similarity with the IATI measure, except if the dataset has a low number of relevant features such as S5NR4 ($N_R = 4$). Similarity with ICW is higher with the PART method than with the ALL when it is compared with optimal features, especially with increasing irrelevant features as S5 and S8.

3- Datasets with a low number of samples such as S1, S4 and S7 (S=100) show clear differences in accuracy; which illustrates that the ALL method achieves higher accuracy than the PART method while the similarity and stability are still low in both methods.

Additionally, all the filter methods used in this investigation show relatively similar behaviour in the similarity measures, except for ReliefF. Also, our datasets are generated from a linear problem, which are ideal for correlation-based methods such as CFS, FCBF and Gain Ratio (Tuv et al., 2009). ReliefF may work better than these filters on other problems such as noisy or non-linear problems, as shown in Section 5.5.3. On the other hand, these filters have a number of weaknesses; for example FCBF often fails with a multiple non-linear interaction dataset because FCBF needs the MDL discretisation step which only works well when the number of categories is small and the response is categorical with a small number of categories (Tuv et al., 2009). Gain Ratio is often sensitive to noise and CFS is highly sensitive to outliers as it uses correlations between features (Tuv et al., 2009). The motivation for using linear

synthetic datasets is just to simplify the problem and to focus more precisely on our investigation, without the need to include more complicated problems, which may affect our results. Accordingly, HEF failed in some cases (depending on the results selected by these four filters), which is not the case for HEF-R1. Therefore, HEF-RI gives better generalisation results than HEF, as it removes some outliers and irrelevant features, which were selected by only one filter.

5.7 Conclusion

In this chapter, the differences between the PART and ALL methods have been investigated in terms of classification accuracy and stability on 10 real benchmark datasets and 21 generated synthetic datasets, as well as on 4 benchmark synthetic datasets.

The results could be summarised as follows:

1-The PART and ALL approaches produce no obvious difference in terms of accuracy and similarity on the real-world and synthetic datasets with high numbers of samples, such as S3, S6 and Splice, and also both methods have high stability.

2- They also demonstrate no obvious differences in terms of accuracy and similarity with the IATI measures on those datasets with a medium number of samples, such as S2, S5 (S = 1,000) and Dermatology, unless the datasets have a high number of irrelevant features, such as S8 and M-feat-factors.

3- These two approaches are demonstrated to have only small differences in accuracy and similarity, and also have high stability on those datasets with low numbers of samples and very low numbers of features, such as Zoo ($N_T = 17$) and Promoters ($N_T = 57$).

4- They show clear differences in accuracy on the datasets with low numbers of samples, such as S1, S4, S7 (S = 100), Colon and Leukaemia, which indicates that the ALL approach achieves higher accuracy than the PART approach, although the similarity and stability results are still low in both the methods.

In conclusion, when the dataset contains a large number of samples (S ≥ 10,000), there is no noticeable difference between these two approaches in terms of stability and accuracy. When the dataset is small, the ALL and PART approaches have almost similar stability. However, there is a clear difference in terms of their accuracy, that is, the ALL approach achieves a higher accuracy than the PART approach, which indicates that the accuracy estimate is possibly overstated and that bias has occurred. Therefore, the PART approach can prevent bias to some extent, although its superiority decreases with increasing sample sizes.

In addition, the experimental results on synthetic datasets present some general conclusions as follow:

1- The number of samples plays a major role in the performance of FS. Whenever the number of samples increases, this leads to the FS method selecting more relevant features and discarding irrelevant ones. Also, it leads to increasing the similarity and stability in addition to the classification accuracy.

2- The number of irrelevant features is an important factor in the performance of FS, as increasing the number of irrelevant features in the dataset disrupts the FS process and increases the possibility of choosing irrelevant features; in addition, it reduces the similarity, stability and classification accuracy.

3- The number and the importance of relevant features also play an important role in the performance of FS. Usually, the FS method is able to select the most highly relevant features but it becomes a challenge when the features have less relevance, as with dataset with $N_R = 16$.

4- Finally, the level of noise is another important factor influencing the FS process in which increases the chances of choosing irrelevant features as well as decreasing the similarity, stability and classification accuracy.

Since the main aim of this thesis is to develop a feature selection ensemble that can improve the reliability and accuracy of feature selection, there are some important issues that need to be investigated in the remaining chapters of this research. Firstly, we should consider how to extend the HEF by applying different wrappers after analysing the results obtained by HEF, aiming to reduce the number of features selected, while preserving the same accuracy and stability. Secondly, we should consider the types of

filters and number of filters that should be included in the proposed ensemble, in addition to choosing a suitable aggregation method, which is an important decision to make. Finally, we will investigate whether weighting the filter members in an ensemble differently may lead to any further improvement of the performance of the HEF.

Chapter 6

Improving the Heuristic Ensemble of Filters

6.1 Introduction

In the earlier chapter, we investigated the stability and accuracy of ALL and PART strategies systematically and then determined their suitability when dealing with datasets with different characteristics. The experiments were carried out by using ten real world benchmark datasets, in addition to twenty one synthetic datasets generated. The results indicate that the PART approach is more effective in reducing the bias when the sample size is small but starts to lose its advantage as the sample size increases. Hence, we chose the PART approach in the remaining chapters.

At this stage, after it had been decided which approach we would use, we went back and focused on the main aim of this research, which is to develop a feature selection ensemble that can improve the reliability by measuring the stability in conjunction with improving the performance by measuring the classification accuracy of feature selection. In this chapter, we attempted to improve the HEF through 3 procedures. Firstly, we extended the HEF by applying different wrappers after the results obtained by HEF, aiming to reduce the number of features selected while preserving the same accuracy and stability. Secondly, we added more filters as members in the HEF. Thirdly, we changed the aggregation method from simply counting the frequency of each feature selected to mean rank aggregation by sorting the selected features based on the means of their ranks in all the ranking filters. In addition, we discussed the partial rank and ways to deal with this situation.

The rest of this chapter is organised as follows: Section 2 provides the description and the frameworks of adding wrapper after HEF. Section 3 discusses adding more filters as a member in the HEF. Section 4 changed the aggregation method from counting the frequency of each feature selected to mean rank aggregation. Section 5 explains the experimental design and procedure, while Section 6 illustrates the results and evaluates the three approaches. Finally, Section 5 concludes our work.

6.2 Adding Wrapper after HEF

6.2.1 Proposed Hybrid Heuristic Ensemble of Filters (HHEF)

Most algorithms for supervised feature selection can be classified as filter or wrapper methods. The wrapper method evaluates the quality of a set of features based on the performance of a learning algorithm. It searches through the space of feature subsets using a specific classifier to guide the search. It tends to lead to better accuracy, but requires high computational effort, compared to filter methods (Kabir and Islam, 2010). Over the last decade, wrapper-based feature subset selection has been an active research. Different search methods have been used to guide the search process, for instance, greedy sequential (Kittler, 1978), floating (Pudil et al., 1994), best-first search (Ginsberg, 1993), and branch and bound (Somol et al., 2004), etc. However, a wrapper is intractable in high-dimensional data, thus, hybrid filter-wrapper methods have been the focus of attention in the last few years (Gutlein et al., 2009, Bermejo et al., 2011, Bermejo et al., 2009, Ruiz et al., 2006, Min and Fangfang, 2010). The idea is to guide the wrapper by the output of the filter which intends to retain the advantage of wrappers while the number of features reduces. However, the hybrid approach, which consists of a single filter and single wrapper, is dependent on the choice of specific filter and wrapper (Leung and Hung, 2010).

In this section we aimed to identify the most important features while preserving the same accuracy and stability. To do this, we applied some wrappers after HEF to make the wrappers capable of focusing on the remaining relevant features after the removal of most of the irrelevant features by HEF.

Figure 6.1 illustrates the proposed hybrid ensemble which operates in three stages. The first stage runs two types of filters individually – Subset Filters (SF) which are CFS and FCBF, and Ranking filters (RFs) which are ReliefF and Gain Ratio. Then, the highest number of features that was selected by the SF was used as a cut-off point to select the top ranked features from the rankings of RF. The second stage aggregates the results of the individuals using a heuristic algorithm based on the frequency and rankings of the selected features. The third stage, which is the novel part of this section, runs different wrappers after HEF and HEF-R1 to identify the most important features, while preserving the same accuracy and stability.

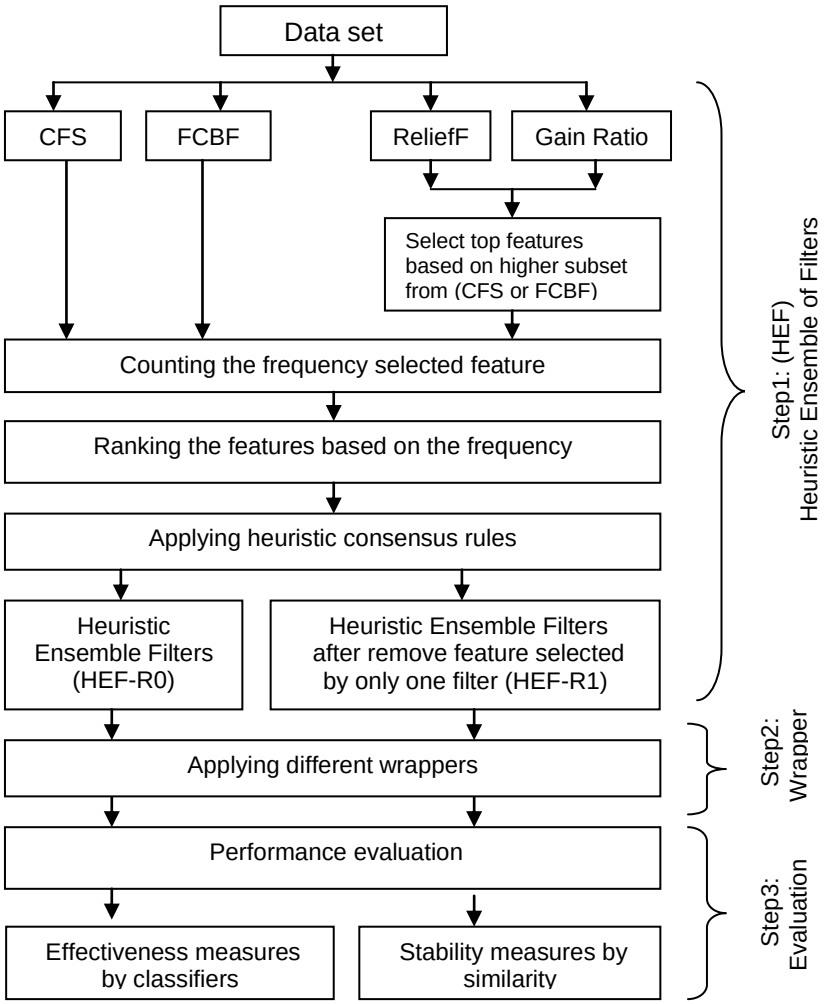


Figure 6.1: Framework of hybrid ensemble of FS

6.2.2 Choice of Wrappers

In principle, any wrapper of any type can be used after our HEF in our hybrid ensemble. However, some factors should be considered when choosing the wrapper, including speed and ability of avoiding over-fitting. Earlier research indicated that extensive search using the wrapper suffers from over-fitting and high computational cost (Kohavi and Sommerfield, 1995, Loughrey and Cunningham, 2005a, Loughrey and Cunningham, 2005b). So, learning from these research studies, our experiment selects wrappers working incrementally at the feature level as greedy forward search and also working incrementally at the block or set of features level as linear forward selection and re-ranking search. We chose three wrappers which had been considered fast, and these are briefly described below to gain an idea of how they work.

1) Greedy Forward search

The search starts by performing a greedy forward search through the space of feature subsets. It starts with no features. It stops when the addition of any remaining features results in a decrease in evaluation. Also, it can produce a ranked list of features by traversing the space from one side to the other and recording the order in which the features are selected (Kittler, 1978).

2) Linear Forward Selection

The search starts by ranking all features then selecting the top-K ranked features as an input to forward selection. The fixed set selects a fixed K number of features. The search direction can be forward, or floating forward selection. Only the K best features are employed in the subsequent forward selection and the rest are discarded (Gutlein et al., 2009).

3) Re-ranking Search

The search starts by ranking all the features; then, the ranking is split into blocks of size G, and an incremental filter-wrapper algorithm is applied but only on the first block. Let X_i be the subset of features selected from this first block. Then the rest of the ranking is re-ranked again but the previously selected subset X_i is taken into account. Then an incremental filter-wrapper algorithm is run again over the first block in this new ranking, but X_i subset is selected for initialisation instead of an empty set and so on. The search stops when no feature is selected in the current block. This search leads to a reduced number of re-ranks, which means that only a few blocks and features are required to be analysed in this method (Bermejo et al., 2011).

6.3 Adding More Filters in HEF

6.3.1 Choice of Filters

This section focuses on adding more filter methods to HEF, as filters are simpler to implement, faster, and more independent of the machine learning model. However, filters are designed with different evaluation criteria which may work well with some datasets, but may not work well with others. Therefore, in order to improve HEF to

select more reliable and stable feature selection, we categorised these evaluation criteria into groups broadly based on the following study (Fahad et al., 2014): distance, information, dependency, statistical and consistency. After that, we studied the popular filters under each of these categories in order to be able to choose the appropriate filters from each category. Then, we chose Chi-x^2 to add it to our ensemble as it is based on statistical measures which were not considered in the earliest experiments (Chapter 4). A statistical criterion is described as a criterion which uses statistical measures and is initially sorted by placing each significant value of the features into its own interval. Chi-x^2 (Liu and Setiono, 1995) is able to perform feature selection and to discretise numeric and ordinal features at the same time (Liu and Setiono, 1997). Also, by running an initial investigation, we found that Chi-x^2 is more stable than the other chosen filters. So we choose this filter as an additional member in HEF aiming to increase the stability of HEF.

6.3.2 Choice of Number of Filters

In terms of determining the number of member filters, we followed the guidelines given in (Wang et al., 2010b), which is that an ensemble of a very few carefully selected filters is similar to or better than ensembles of many filters. So, in this concept demonstration study, we initially chose a total of four filters in the previous chapters – two rank filters, namely ReliefF (Kononenko, 1994) and Gain Ratio (Quinlan, 1993), and two subset filters, namely Correlation-based Feature Selection (CFS) (Hall, 1999) and Fast Correlation-Based Filter (FCBF) (Yu and Liu, 2004). All of these filters were described in Chapter 2. However, in this chapter we added Chi-x^2 as an additional member in HEF because it represents other evaluation criteria which are statistical based and we found that Chi-x^2 is more stable than the other filter members in HEF. Consequently, five filters are chosen as members in the HEF.

It should be noted that each algorithm uses a different criterion in evaluating the relevance of the candidate features in datasets. When combined, candidate features are assessed from many different aspects.

6.4 Changing the Aggregation Method for Combining Feature Subsets

The aggregation method is an essential part of determining the HEF result. However, there are two issues that needed to be addressed before changing our aggregation method from the frequency (which we had been using) to another aggregation method. Firstly, SF had produced subset features without ranking these features, which forced us to use the frequency and limit our options of using other rank aggregation methods. Secondly, each filter member produced subset features, even for RF, because we had selected the top features based on the highest subset from SF. Therefore, we need to consider the partial rank and how to deal with this situation while modifying the aggregation methods. The following sections discuss how we can solve these two issues.

6.4.1 Converting Feature Subset to Ranking Subset

There are some studies that have investigated aggregation methods, but most of them use ranking features, which is the outcome of RF (Wang et al., 2010b, Wang et al., 2011). They rarely used the subset feature by counting the frequency of each feature (Abeel et al., 2010); only if they cut the ranking feature early before the aggregation step (Altidor et al., 2011).

In this section, we are going to discuss methods to solve the first issue by converting the subset filters (FCBF and CFS) to ranked subset filters, with suitable ranking evaluation criteria.

FCBF (Yu and Liu, 2003): is a subset filter that works based on correlation measure, relevance and redundancy analysis, used in conjunction with a Symmetrical Uncertainty feature evaluation. This filter has an option in WEKA to generate ranking subset features – "generate Ranking". FCBF is capable of generating attribute rankings (Witten and Frank, 2000).

CFS (Hall, 2000): is also a subset filter that prefers a subset of features that are highly correlated with the class while having low inter-correlation with each other. As we

mentioned above, in order to avoid the high computational cost of CFS, such as SF, we used linear forward selection (LFS) as a search method, together with CFS instead of using best-first search (Gutlein et al., 2009). In order to rank the output of CFS, we followed the suggestions of the developer in Hall and Holmes (2003), and the forward selection search method was used with the CFS method to produce rank lists of attributes. Also, Hall (2014) stated that "Subset Evaluators, such as CFS and the wrapper, do not produce a ranking. They just return a single best subset of features found during the search. There is no significance to the order of attributes produced by the attribute selection filter and the output from the evaluation in "weka.attributeSelection" in this case. One can derive a ranking from these methods if "GreedyStepwise" is used as the search and the option to produce a ranked list is turned on" [5].

The issue was solved by ranking the subset feature from FCBF and CFS. Now we will look forward to solving the procedure to deal with the partial rank.

6.4.2 Dealing with Partial List or (Top-K List)

In our experiment, each filter member produced a ranking of subset features, even RF, since we had selected the top features based on the highest subset from SF. Therefore, we need to consider the partial rank (top-K list) and methods to deal with this situation, while modifying the aggregation method from counting the frequency of each feature that handles the partial list to other aggregations that take into account the position or the score of each feature.

It is assumed that a special case of a top-K list is a "full list" (Fagin et al., 2003), that is, a combination of all of the features in a dataset. Top-K lists are only the few most encouraging features that can be further examined in follow-up studies. And this is the reason for taking top-K lists into account. Over 10,000 features may be present in a typical list of a genetic dataset, but K will typically range between 25 and 100

[5] Also, with my communication on November 14, 2014, Hall confirmed that the output of CFS can be ranked by using forward selection as the search and the option can be turned on to produce a ranked list together with CFS as evaluation.

(DeConde et al., 2006). Moreover, a full ranked list is often unavailable in the case of many conditions, such as biological problems etc. Instead, only a top-K list would attract the interest of many on such occasions (Lin, 2010).

There are several standard methods for comparing or aggregating a full list with different rankings. However, we cannot simply use these methods because most of them deal only with comparing one list against another over the same features (full list) (Fagin et al., 2003).

Problems arose in recent years while comparing top-K list and rank aggregation (Fagin et al., 2003, Dash and Liu, 2003, DeConde et al., 2006, Lin, 2010). For example, in order to handle the top-K list using mean aggregation, several studies (Fagin et al., 2003, Prati, 2012) assumed that all the features that had not been ranked would appear at the bottom of the ranked list. Accordingly, they give the position of each feature not appearing in the list equal to K+1, where K is the maximum number of features in the partial list. Based on these studies, we make each feature not appear in the list position equal to K+1 to solve the partial list issue.

6.4.3 Ranking Aggregation Methods

Aggregating the diverse outputs from different FS methods into a single result is a key component in feature selection ensembles. Hence, choosing a suitable aggregation method is an important decision to make. In the previous chapters, we applied the method of counting the frequency of each feature, because we had relied on the subset features without considering the ranking of these features. However, in this section, after solving the two issues by ranking the SF and dealing with the top-K list, we were able to use other techniques for aggregating the rank features. There are a number of techniques to aggregate rank features, such as mean, median, lowest rank, highest rank, robust rank aggregation (Kolde et al., 2012), stability selection (Haury et al., 2011), exponential weighting (Haury et al., 2011), enhance Borda (Wald et al., 2012) and round robin (Neumayer et al., 2011). Wald and his colleagues (2012) made an extensive comparison of nine rank aggregation methods in term of similarity and they found a number of groups with similar rank aggregation techniques, as follows: the first group are mean, median, stability selection, exponential weighting, enhance Borda and robust

rank aggregation; and the second group are highest rank and round robin. The lowest rank aggregation is not similar to any ranking techniques. These groups guided the researchers and gave them an idea of the techniques to focus on when attempting to study a large range of aggregation techniques. So, if there are two aggregation techniques and they produce similar results, there is no need to apply the one that requires more computation (Wald et al., 2012). However, it is found by Wang et al (2011) that the mean method performs better than the median in terms of accuracy. Also, we can note that two of the well-known ensemble types, mean and median, are each mathematically equivalent to more complex methods, as long as all the lists being aggregated are full lists. Mean aggregation is equivalent to the Borda and median is equivalent to the Spearman footrule (Wald et al., 2012). In addition to that, more sophisticated methods have been developed, such as Condorcet, Schulze and Markov Chain (Prati, 2012). However, Condorcet, Schulze and Markov Chain are computationally expensive and not suited to the case of extremely large search spaces (Wald et al., 2012).

Accordingly, we decided to use the most commonly used rank list aggregation technique of mean aggregation. But only such mean aggregation that deals with partial lists.

On the other hand, there are two methods towards aggregation: rank-based aggregation and score-based aggregation. Rank-based aggregation only takes into account the order (position) of the features, while score-based aggregation is the combination of the features based on their score for each feature produced by each FS method (Dittman et al., 2013). Rank-based aggregation has a number of advantages. Firstly, it is computationally cheap and requires few or no parameters to set up. Secondly, it is naturally calibrated and scale insensitive, while for score-based aggregation, first of all, we need to rescale the value within the same range. Nevertheless, the values might be in the same absolute scale, or they may represent different relative scales (Prati, 2012). Recently, it has been found by Khoshgoftaar et al. (2013) that the rank-based aggregation method outperforms the score-based aggregation method for the majority of the datasets. As a result, we selected the rank-based aggregation method in our experiment. The mathematical formulation is shown below:

Let $\{x_1, \ldots, x_i, \ldots, x_N\}$ be a set of N features (full features) in a dataset D, l representing the number of ranks generated from l number of filters and r_i representing

a ranking position of x_i in ranking j (R_j), Where $1 \leq r_i \leq N$. The mean rank aggregation of full ranking $\bar{r}(x_i)$ is given by:

$$\bar{r}(x_i) = \frac{1}{l}\sum_{j=1}^{l} r_{ij} \quad (6.1)$$

While in our research we need to consider the partial rank. So, the mean rank aggregation of partial ranking $\bar{r_p}(x_i)$ of K features is given by:

$$\bar{r_p}(x_i) = \frac{1}{l}\sum_{j=1}^{l} r_{ij} \quad (6.2)$$

Where K is number of features selected from N (K < N) and $r_{ij} = K + 1$, if $x_i \notin R_{pj}$. Also, in some cases, K has different length, then $r_{ij} = K_j + 1$.

Additionally, let $f(x_i)$ represent the frequency of x_i appearing in the selected subsets, where m_j is a subset of R_{pj}:

$$f(x_i) = \sum_{j=1}^{l} \delta(m_j, x_i) \quad (6.3)$$

$$\delta(m_j, x_i) = \begin{cases} 1, & x_i \in m_j \\ 0, & x_i \notin m_j \end{cases} \quad (6.4)$$

In this section, three schemes of mean rank aggregation with partial list have been proposed:

1. The first one ranks the features based on the frequency $f(x_i)$, but the chance of the features having the same frequency is high. To resolve this issue, we had ranked them by means of these features $\bar{r_p}(X_i)$ and we made the position of each feature not appear in the lists equal to K+1, where k is the maximum number of features in the partial list.
2. The second one ranks the features based on the mean and we made each feature not appear in the list; the position is equal to K+1.
3. The third one ranks the features based on the mean and we made each feature not appear in the list; the position is equal to K+1. Then we divided the mean of each feature by the frequency of this feature.

$$\bar{r_p}(x_i) = \frac{\bar{r_p}(x_i)}{f(x_i)} \quad (6.5)$$

The reason for dividing the mean of each feature by the frequency is to make the rank order of the features selected by most of the filters smaller, which leads to these features rising to the top of the ranking, while the features selected by a few filters still remain at the bottom of the ranking.

6.5 Experiments

6.5.1 Experiment Design and Procedure

To verify the consistency in our experiments, we used the same 10 real datasets and the same classifiers: NB (John and Langley, 1995), KNN (Aha et al., 1991) and SVM (Platt, 1999). The 10-fold cross-validation strategy was used in the FS and classification stages; moreover, each experiment was repeated 10 times with different shuffling of the data. In total, 51,000 models – 17 (5 FS + 12 ensemble) × 10 (datasets) × 3 (classifiers) × 10 (run) × 10 (folds) – were built for the experiments.

The statistical significance of the results of the multiple runs for each experiment was calculated, and the comparisons between accuracies were done with the Friedman test with a significance level of 0.05 (Demšar, 2006); this is a non-parametric test. It ranks the algorithms for each dataset independently. The algorithm with best performance gets the rank of 1, the second best one gets the rank 2 and so on. In case of ties, average ranks are assigned. Then, if the null hypothesis is rejected, the Nemenyi test can proceed. It is used when all algorithms are compared to each other on multiple testing datasets. The performances of two algorithms are significantly different if the corresponding average ranks differ by at least the critical difference:

$$\text{CD}=q_{\alpha}\sqrt{\frac{A(A+1)}{6D}}$$

where A is number of algorithms, D is number of datasets used and q_{α} is the critical value. These are all based on the Studentized range statistic divided by $\sqrt{2}$ (Demšar, 2006).

Moreover, in addition to accuracy, we will measure the stability of FS, as in each fold, the FS method may produce different feature subsets. Measuring stability requires a similarity measure for the FS results. The stability measures used in our investigation are Relative Weighted Consistency (CWrel) and Average Tanimoto Index (ATI) (Somol

and Novovicova, 2010), as the subset cardinality is not equal in our research. ATI evaluates pair-wise similarities between subsets in the system (10 folds), while CW_{rel} evaluates the overall occurrence of the features in the system (10 folds) as a whole. $CW_{rel}$and ATI may produce different results in each run, so the average of 10 runs will be used.

Furthermore, aggregating the outputs of all filter members by mean may produce a higher number of selected features, including features with low frequency levels selected by only a couple of filters (or even a single one). In order to address this issue and also to obtain more important features, we have selected the top 75%, 50% and 25% of features from our final ranked list.

6.6 Results

In this section, the classification accuracy and stability results obtained after applying the different proposed methods were shown. To sum up, three wrappers after HEF and HEF-R1 were applied. Also, we added Chi-x^2 as a member in HEF and compared the results with the previous version of HEF. Moreover, we changed the aggregation method from simply counting the frequency of each feature selected to mean rank aggregation by sorting the selected features based on the means of their ranks in all of the ranking filters.

6.6.1 Hybrid Heuristic Ensemble of Filters (HHEF)

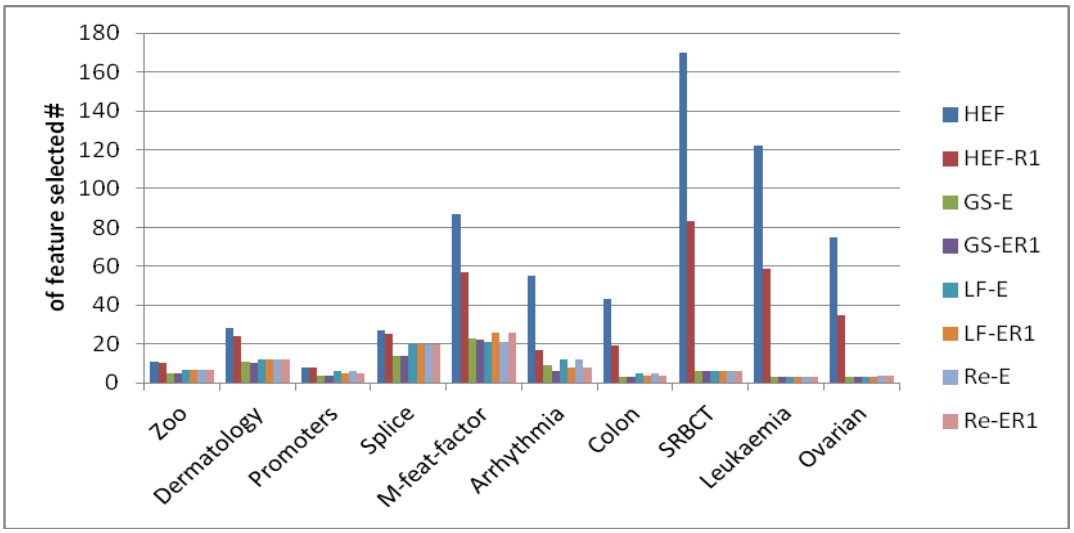


Figure 6.2: Average number of features selected by HHEF

Figure 6.2 shows the average number of selected features by each hybrid ensemble approach HHEF, where GS, LF and Re represent the wrappers after HEF or HEF-RI by using Greedy search, Linear Forward Selection and Re-ranking Search, respectively, in addition to simple HEF and HEF-R1. We observed from the figure that when the wrappers had been applied after the HEF output, they helped to reduce the number of selected features, as many as three times, especially for microarray datasets, to reveal the most important features.

6.6.1.1 Accuracy Evaluation

Figures 6.3 – 6.5 show the average test accuracy of NB, KNN and SVM classifiers, respectively, which used the features selected by HEF, HEF-R1 in addition to three wrappers applied after the HEF and HEF-R1. The results in these three figures reveal similar patterns as follows: firstly, HEF and HEF-R1 have a higher average accuracy, which indicates that HEF and HEF-R1 are more accurate than applying wrappers after them. Secondly, adding wrapper after the results of HEF has a lower average accuracy, especially on microarray datasets, which indicates that using a wrapper after HEF and HEF-R1 may help to identify the most important features (as seen in Figure 6.2), but it leaves out some less important features, decreasing the classification accuracy.

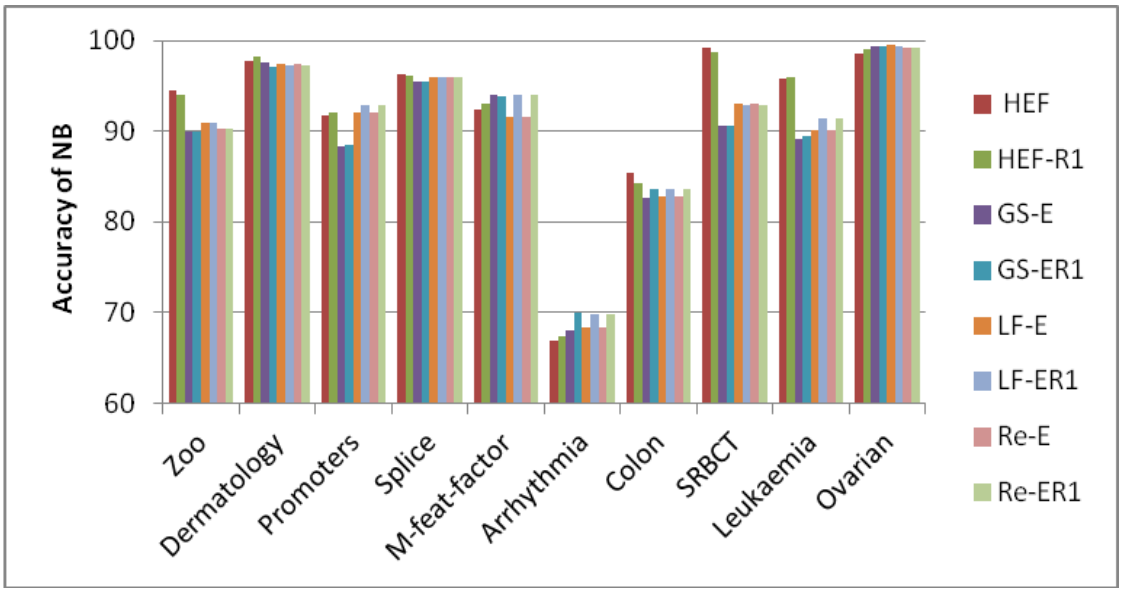


Figure 6.3: The average test accuracy of NB classifiers trained with 2 HEF and 6 hybrid HEF

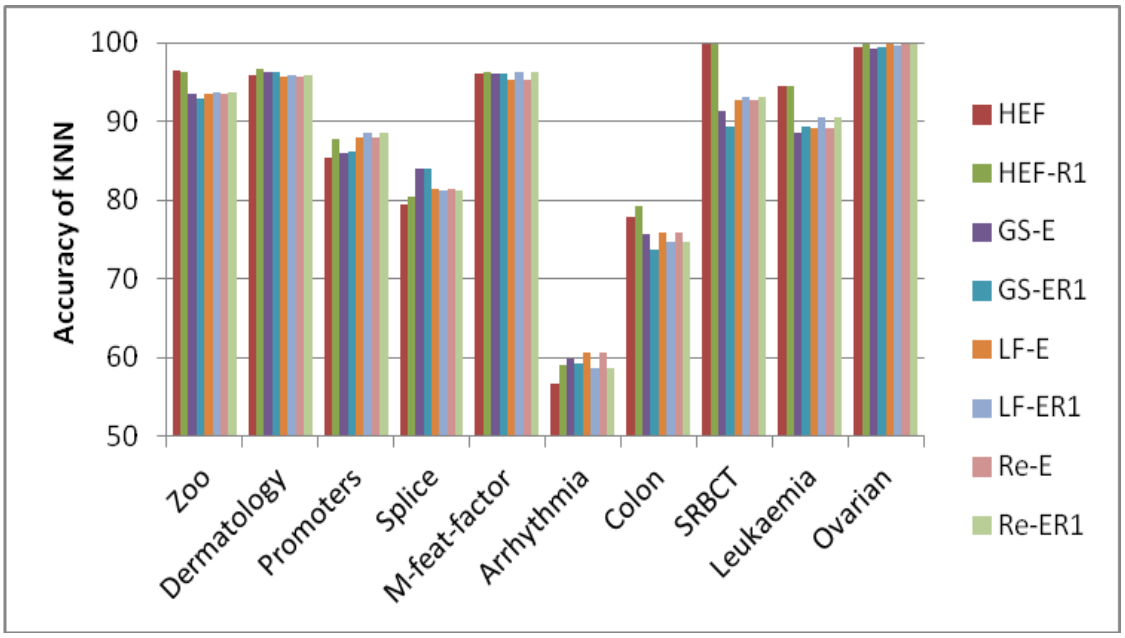


Figure 6.4: The average test accuracy of KNN classifiers trained with 2 HEF and 6 hybrid HEF

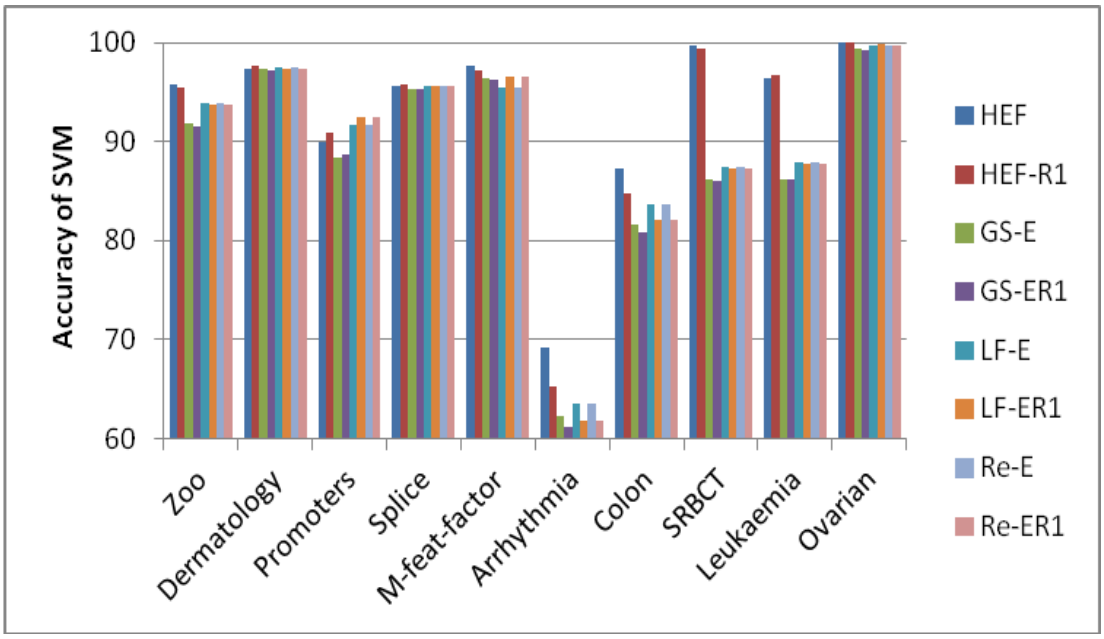


Figure 6.5: The average test accuracy of SVM trained with 2 HEF and 6 hybrid HEF

The Nemenyi test shows that there is an insignificant difference in accuracy results using NB and KNN of HEF and all hybrid ensemble approaches against each other. On the other hand, the Nemenyi test presents a significant accuracy improvement by using SVM. Accordingly, we can identify three groups in the accuracy of SVM: the accuracy of GS-ER1 is significantly worse than those of HEF-R1, HEF, LF-E and Re-E. Also, accuracy of GS-E is significantly worse than those of HEF-R1, HEF. LF-E, Re-E, LF-ER1and Re-ER1 belong to all the groups, as we can see in Figure 6.6. at p <0.05.

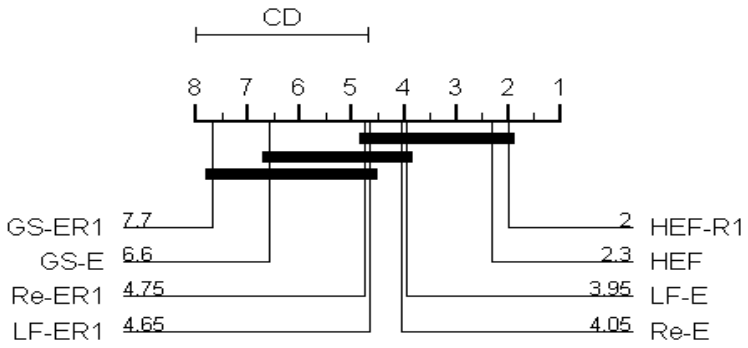


Figure 6.6: Accuracy comparison using SVM of HEF and all hybrid ensemble approaches against each other with Nemenyi test.

6.6.1.2 Similarity Evaluation

Figure 6.7 shows the average stability of ATI using 10 datasets focusing on different hybrid ensemble approaches. It is clearly seen that HEF and HEF- R1 had the highest stability and outperformed the hybrid ensemble approaches. In contrast, LF-E, GS-E and Re-E had the lowest stability.

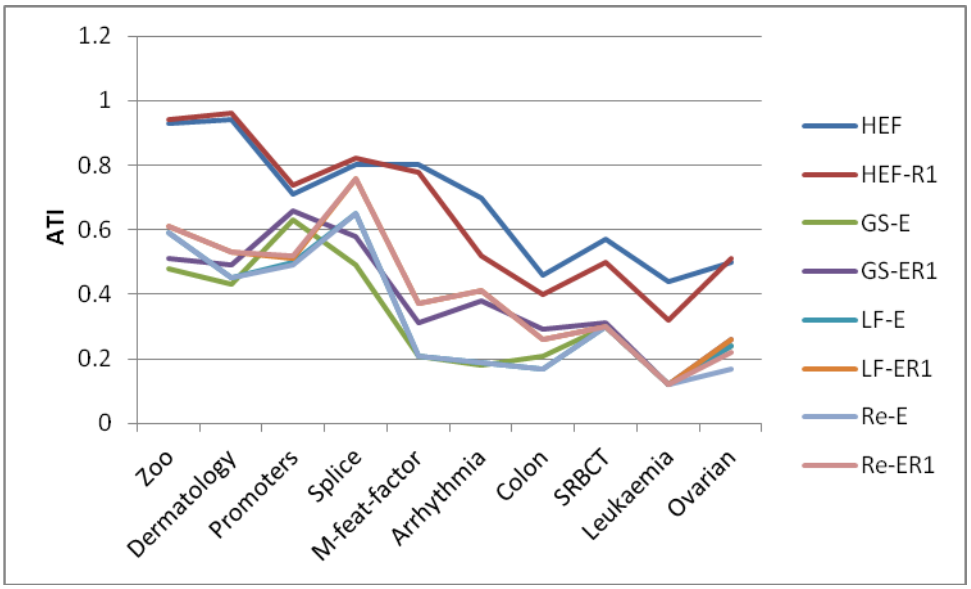


Figure 6.7: The stability measures of ATI with the features selected by 2 HEF and 6 hybrid HEF

The Nemenyi test showed that the accuracy of HEF and HEF-R1 had been significantly better than LF-E, GS-E and Re-E, as we can see in Figure 6.8. at $p < 0.05$. We can identify two groups of ensemble approaches: the accuracy of HEF and HEF-R1 are significantly better than that of LF-E, GS-E and Re-E, while LF-ER1, GS-ER1 and Re-ER1 belongs to both groups.

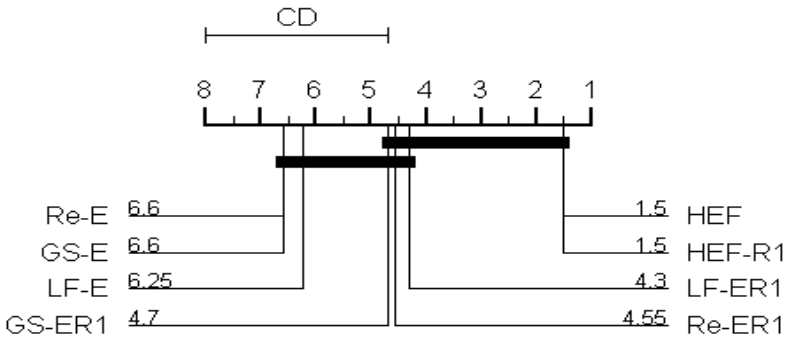


Figure 6.8: Stability comparison using ATI of HEF and all hybrid ensemble approaches against each other with the Nemenyi test

In summary, the experimental results demonstrated that the HEF is more reliable, consistent and effective than hybrid HEF as the features selected by the HEF achieved better accuracy and more stable results. Furthermore, when the wrappers had been applied after the HEF output, they helped to reduce the number of selected features, as many as three times especially with microarray datasets, to reveal the most important features, by sacrificing some overall classification accuracy and stability.

So, based on the above results, we shall not work more on HHEF but rather extend the investigation by adding more filters as members, aiming to achieve further improvement in the HEF.

6.6.2 Adding More Filters in HEF

In this section we investigated the benefits of adding more filter members in the ensemble results. The four filters (FCBF, CFS, ReliefF and GR) which had been used in the previous Chapters as members in HEF produced good results. However, we had aimed to improve the HEF's result by splitting filter methods to different evaluation categories as it could be seen in Section 6.3.1. Then we selected filters as members from each category. So based on our discussion in Section 6.3.1, we added Chi-x^2 filter as a fifth member in HEF in the hope of gaining an improvement in terms of accuracy or stability.

6.6.2.1 Accuracy Evaluation

Tables 6.1-6.3 show the average accuracy of NB, KNN and SVM classifiers on the 10 datasets; each value presented in the tables is the average over 10 runs of 10-fold cross-validation outcomes. For each classifier, the accuracies of classification on the datasets with all the original features are given in the "All features" column as a comparison, and "HEF+5F, HEF-R1+5F" represents the ensemble of five filters including Chi-x^2 while "HEF+4F, HEF-R1+4F" represents the ensemble with four filters without Chi-x^2. It should be noted that in comparison when we state that filter A is better or worse than filter B for simplicity, it means that the models trained with the features selected by filter A are better or worse than the models trained with the features selected by filter B, under the same experimental set-ups.

Tables 6.1-6.3 show what we expected, which is that each single filter performed well in some datasets (in bold) but poorly in others. This confirms the perception that the accuracy of individual filters is inconsistent and that there is no meaningful pattern that can be extracted to indicate when they do better and when they do not. Nevertheless, the NB and KNN classifiers trained with the features selected by HEF-R1+5F have a higher average accuracy for all the datasets, which indicates that HEF-R1+5F are more accurate than the individual filters in feature selection. On the other hand, the SVM classifier trained with the features selected by HEF+5F is the overall winner as it has a marginally higher average accuracy than all the others. One different phenomenon observed is that SVM models trained with the full feature set performed not as bad as for the other two types of models (NB and KNN), and even gave the highest accuracy on three datasets (Multi-Feature Factor, Arrhythmia and Leukaemia).

Table 6.1: The accuracies of NB models trained with all the features and the features selected by filters and heuristic ensembles

NB	All	FCBF	CFS	ReliefF	GR	Chi-x^2	HEF+5F	HEF-R1+5F	HEF+4F	HEF-R1+4F
Zoo	93.96	93.45	93.3	94.28	93.59	**96.15**	94.46	94.27	94.46	94.07
Dermatology	97.43	97.49	98.09	95.91	85.45	87.15	97.43	98	97.79	**98.31**
Promoters	90.19	**92.48**	**92.48**	90.39	92.19	92.1	91.6	92.11	91.7	92.01
Splice	95.41	95.84	95.84	**96.32**	95.98	96.01	96.2	96.2	96.21	96.18
M-feat-factor	92.47	93.93	**93.96**	87.82	89.82	89.84	92.17	92.41	92.45	93.01
Arrhythmia	62.39	68.1	**68.72**	63.48	54.71	58.15	66.07	67.32	66.89	67.32
Colon	55.81	80.22	82.21	84.33	79.12	82.19	84	**84.88**	85.4	84.29
SRBCT	99.04	95.56	97.21	99.06	99.17	98.54	**99.64**	98.64	99.28	98.67
Leukaemia	**98.75**	95.68	96.09	95.18	95.8	95.82	95.43	96.95	95.82	95.96
Ovarian	92.41	**99.72**	99.45	97.7	97.81	97.7	98.29	98.42	98.49	98.97
Average	87.78	91.24	91.73	90.447	88.36	89.36	91.53	**91.92**	91.84	91.87
St. Dv.	14.67	9.17	8.91	9.99	12.60	11.50	9.47	9.10	9.17	9.16

Table 6.2: The accuracies of KNN models trained with all the features and the features selected by filters and heuristic ensembles

KNN	All	FCBF	CFS	ReliefF	GR	Chi-x^2	HEF+5F	HEF-R1+5F	HEF+4F	HEF-R1+4F
Zoo	96.14	95.13	95.63	96.35	96.22	96.13	**96.54**	96.43	96.44	96.23
Dermatology	94.64	95.0	**96.64**	93.55	86.47	86.55	94.37	95.92	95.8	96.61
Promoters	79.71	87.61	87.61	84.67	**90.11**	90.02	85.97	89.03	85.47	87.75
Splice	74.43	80.9	80.9	81.22	**82.37**	82.28	79.4	81.55	79.4	80.36
M-feat-factor	96.03	96.29	**96.42**	94.11	95.24	95.11	96.21	95.97	96.15	96.17
Arrhythmia	53.2	60.94	**61.46**	55.84	45.93	52.49	54.51	57.82	56.61	59.01
Colon	76.83	79.17	79.38	78.57	80.0	**80.45**	78.79	**80.43**	77.79	79.21
SRBCT	82.39	98.21	99.65	**100.0**	99.65	99.26	99.63	99.89	99.75	99.76
Leukaemia	88.39	**94.88**	94.2	93.45	92.66	92.36	94.07	94.84	94.48	94.55
Ovarian	94.86	99.76	99.68	98.97	98.86	98.3	99.52	99.84	99.52	**99.84**
Average	83.66	88.78	89.15	87.67	86.75	87.29	87.901	**89.172**	88.14	88.94
St. Dv.	12.86	11.45	11.53	12.69	15.00	13.11	13.30	12.33	12.96	12.64

Table 6.3: The accuracies of SVM models trained with all the features and the features selected by filters and heuristic ensembles

SVM	All	FCBF	CFS	ReliefF	GR	Chi-x^2	HEF+5F	HEF-R1+5F	HEF+4F	HEF-R1+4F
Zoo	96.24	95.13	95.84	94.85	95.73	**96.83**	95.74	95.64	95.74	95.44
Dermatology	96.04	97.03	97.51	95.6	88.16	88.22	97.21	**97.54**	97.29	**97.71**
Promoters	91.03	**92.25**	92.15	88.89	91.65	91.67	90.89	91.19	90.02	90.89
Splice	93.13	95.48	95.48	**96.14**	95.9	95.91	95.79	95.72	95.68	95.79
M-feat-fact	**97.7**	97.25	97.42	96.13	96.51	96.19	97.69	97.2	97.68	97.17
Arrhythmia	**71.06**	60.45	66.24	67.46	59.16	60.56	68.94	64.87	69.18	65.29
Colon	84.52	83.79	85.43	85.19	82.0	84.26	**87.29**	85.12	87.26	84.79
SRBCT	99.63	98.57	99.04	99.18	99.29	**99.64**	**99.64**	99.51	**99.63**	99.4
Leukaemia	**98.04**	96.52	96.21	96.53	95.52	95.84	96.25	96.8	96.39	96.64
Ovarian	99.96	99.96	**100.0**	99.33	99.17	98.57	**100**	**100**	**100.0**	99.96
Average	92.73	91.64	92.53	91.93	90.309	90.76	**92.944**	92.35	92.88	92.30
St. Dv.	**8.46**	11.23	9.60	9.16	11.53	11.04	8.80	10.05	8.76	9.95

As we can see in Figure 6.9, the Nemenyi test identifies two groups by evaluating the accuracy of KNN: the accuracy of HEF-R1+5F, HEF-R1+4F and CFS are significantly better than "All features", whereas, FCBC, HEF+4F, HEF+5F, Gain Ratio, ReliefF and Chi-x^2 belong to both groups. The accuracy of NB and SVM are not significant and the Friedman test cannot reject the null hypotheses.

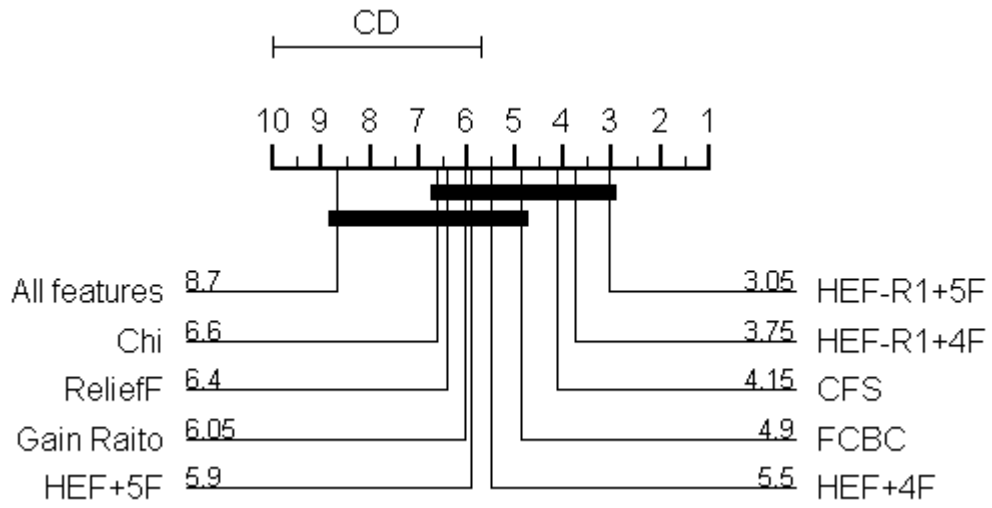


Figure 6.9: Results of the Nemenyi test used to evaluate the accuracy of KNN of each filter and ensemble approaches against each other

Table 6.4: The number of best and worst accuracies summarisation of three classifiers

		FCBC	**CFS**	**ReliefF**	**Gain Raito**	**Chi-x^2**	**HEF +5F**	**HEF-R1+5 F**	**HEF+ 4F**	**HEF-R1+4F**
NB	Best	2	3	1	0	1	1	1	0	1
	Worst	-2	-2	-4	-2	-2	0	0	0	0
KNN	best	1	3	1	2	1	1	1	0	0
	worst	-2	0	-2	-2	-2	0	0	0	0
SVM	best	1	1	1	0	2	3	2	2	1
	worst	-2	-1	-3	-4	-1	0	0	0	0
All classifiers	**best**	**4**	**6**	**3**	**2**	**4**	**5**	**4**	**2**	**2**
	worst	**-6**	**-3**	**-9**	**-8**	**-5**	**0**	**0**	**0**	**0**

Table 6.4 summarises the number of best cases (positive number) and worst cases (negative numbers) for all the filters and ensemble on each individual classifier, as well as for all classifiers together. From the table above, we can observe the following results:

1- HEF+5F and HEF-R1+5F achieved the best accuracy result but never delivered the worst. So they have the most frequency in the best case and less frequency in the worst case in total.

2- Among five filters, CFS showed the best case for more frequency and less frequency, and the worst case in total than the other filters in terms of accuracy. The Chi-x^2 is in second place with four best cases and five worst cases which is better than the remaining filters. ReliefF and then Gain Ratio showed the highest number of worst cases, even though they showed a number of best cases.

6.6.1.2 Similarity evaluation

In addition to accuracy, we measured the stability of each filter and ensemble with and without the Chi-x^2 filter in order to know if adding Chi-x^2 will improve the stability of the ensemble result or not.

Tables 6.5 and 6.6 show how each filter, as well as the ensemble method, has different stability in the same dataset; thus, it is apparent that some filters are more stable than

others when the sample changes. As we can see, Chi-x^2 has a higher average stability for all the datasets, and after that, ReliefF and then Gain Ratio, which indicates that RF is more stable when changing datasets than other methods. In contrast, FCBF and CFS were unstable in the face of changes in the samples, while HEF and HEF-R1 scored in between the rank and subset filters. This proves that the ensemble method improves the level of stability even if some of the members are relatively unstable. Also, we can observe the improvement in the stability on HEF+5F and HEF-R1+5F after adding Chi-x^2 as a member, compared to HEF+4F and HEF-R1+4F.

Table 6.5: The stability measures of ATI with the features selected by 5 filters and 4 heuristic ensembles

Dataset	FCBC	CFS	ReliefF	Gain Raito	Chi-x^2	HEF+ 5F	HEF-R1+ 5F	HEF +4F	HEF-R1+ 4F
Zoo	**0.96**	0.94	0.91	0.91	**0.96**	0.94	0.95	0.93	0.94
Dermatology	0.81	0.92	0.93	**0.97**	0.95	0.94	0.96	0.94	0.96
Promoters	0.75	0.75	0.75	**0.81**	**0.81**	0.71	0.74	0.71	0.74
Splice	0.76	0.76	0.91	**0.94**	**0.94**	0.8	0.81	0.8	0.82
M-feat-factor	0.64	0.7	0.89	0.75	**0.91**	0.77	0.74	0.8	0.78
Arrhythmia	0.43	0.56	0.77	0.72	**0.78**	0.69	0.64	0.7	0.52
Colon	0.28	0.36	**0.66**	0.41	0.61	0.47	0.39	0.46	0.4
SRBCT	0.36	0.44	**0.66**	0.61	0.64	0.56	0.52	0.57	0.5
Leukaemia	0.22	0.26	**0.61**	0.55	0.6	0.43	0.36	0.44	0.32
Ovarian	0.29	0.34	**0.76**	0.7	0.9	0.52	0.49	0.5	0.51
Average	0.55	0.60	0.78	0.73	**0.81**	0.68	0.66	0.68	0.64
St. Dv.	0.250	0.23	0.11	0.17	0.13	0.17	0.20	0.17	0.21

Table 6.6: The stability measures of CWrel with the features selected by 5 filters and 4 heuristic ensembles

Dataset	FCBC	CFS	ReliefF	Gain Raito	Chi-x^2	HEF+ 5F	HEF-R1+ 5F	HEF+ 4F	HEF-R1+ 4F
Zoo	**1.0**	0.94	0.9	0.9	0.97	0.95	0.96	0.94	0.95
Dermatology	0.83	0.92	0.93	**0.98**	0.95	0.85	0.97	0.85	0.97
Promoters	0.85	0.85	0.85	**0.9**	**0.9**	0.81	0.83	0.81	0.83
Splice	0.81	0.81	0.94	**0.96**	**0.96**	0.82	0.84	0.82	0.85
M-feat-factor	0.75	0.8	0.93	0.84	**0.95**	0.83	0.85	0.83	0.85
Arrhythmia	0.56	0.71	0.87	0.84	**0.88**	0.8	0.77	0.8	0.67
Colon	0.39	0.5	**0.79**	0.56	0.75	0.66	0.54	0.62	0.55
SRBCT	0.53	0.61	0.79	0.76	0.78	**0.81**	0.7	**0.81**	0.66
Leukaemia	0.34	0.41	**0.75**	0.71	**0.75**	0.65	0.56	0.65	0.52
Ovarian	0.43	0.49	0.86	0.82	**0.95**	0.75	0.65	0.66	0.66
Average	0.649	0.704	0.86	0.82	**0.884**	0.793	0.767	0.779	0.751
St. Dv.	0.21	0.18	0.06	0.1	0.08	0.08	0.143	0.09	0.15

As we can see in Figure 6.10, the Nemenyi test identifies four groups when comparing the stability using ATI, while in Figure 6.11, the Nemenyi test identifies three groups when comparing the stability using CWrel. These two figures show similar results by ranking Chi-$\mathbf{x^2}$ as the first stable filter, and after that, ReliefF then Gain Ratio. This indicates that RF is more stable in the case of changing samples than other methods. In contrast, FCBF and CFS are ranked as the least stable filters, which means that these are unstable in the face of changes in the samples, while HEF+5F is ranked before HEF+4F, which means HEF+5F is more stable than HEF+4F. So, we can declare that adding Chi-$\mathbf{x^2}$ as a member contributes to the stability of HEF.

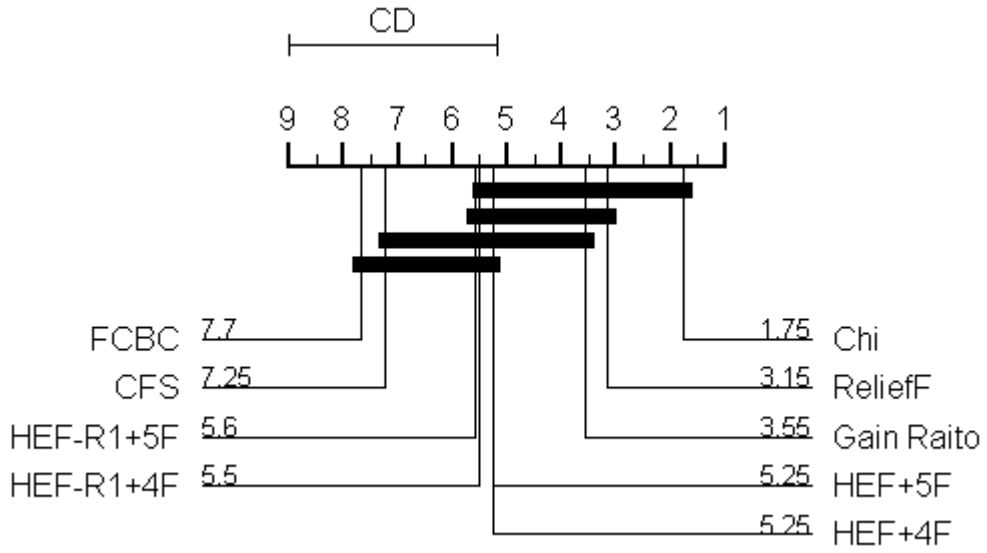


Figure 6.10: Stability comparison using ATI of each filters and ensemble approaches against each other with Nemenyi test

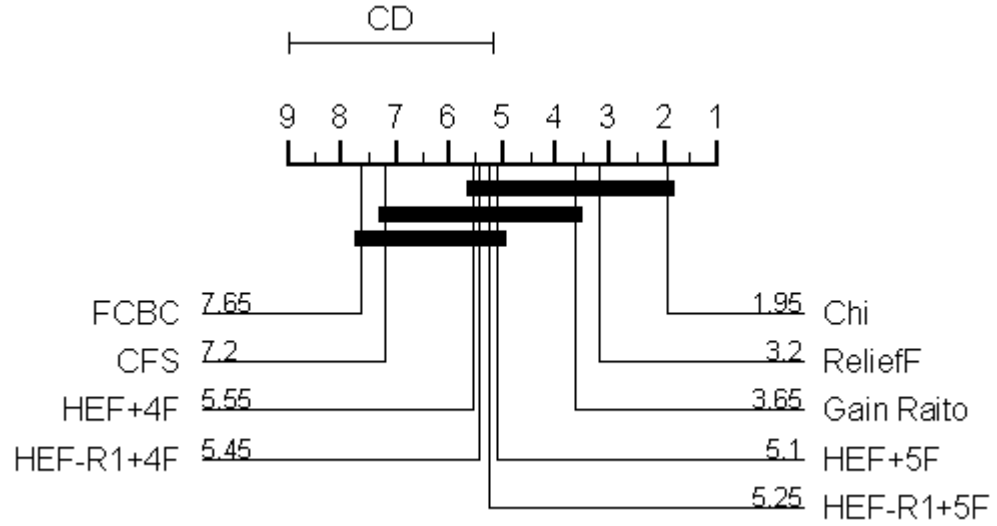


Figure 6.11: Stability comparison using CWrel of each filters and ensemble approaches against each other with Nemenyi test

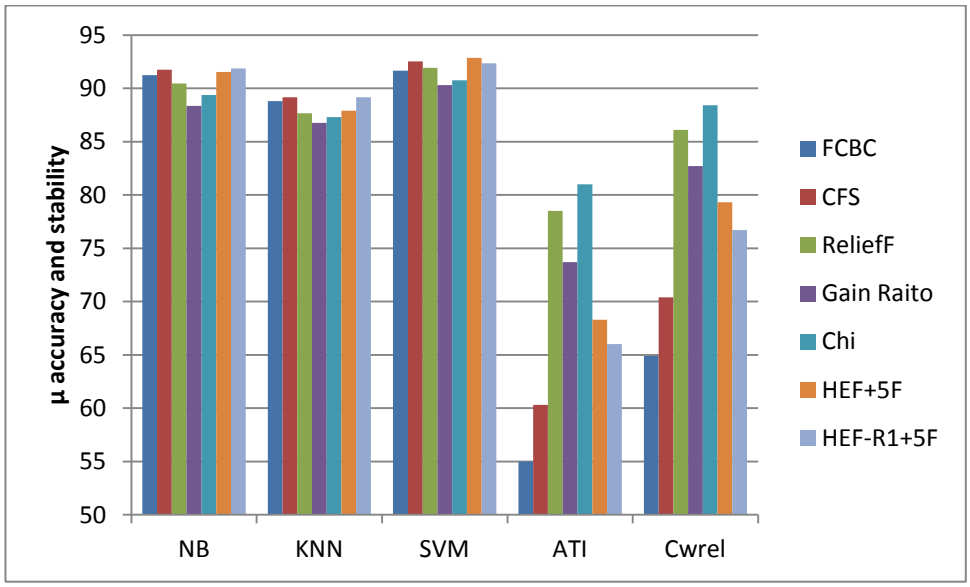


Figure 6.12: Average accuracy and stability of HEF+5F and 5 filter members on 10 real datasets, focusing on each evaluation measure

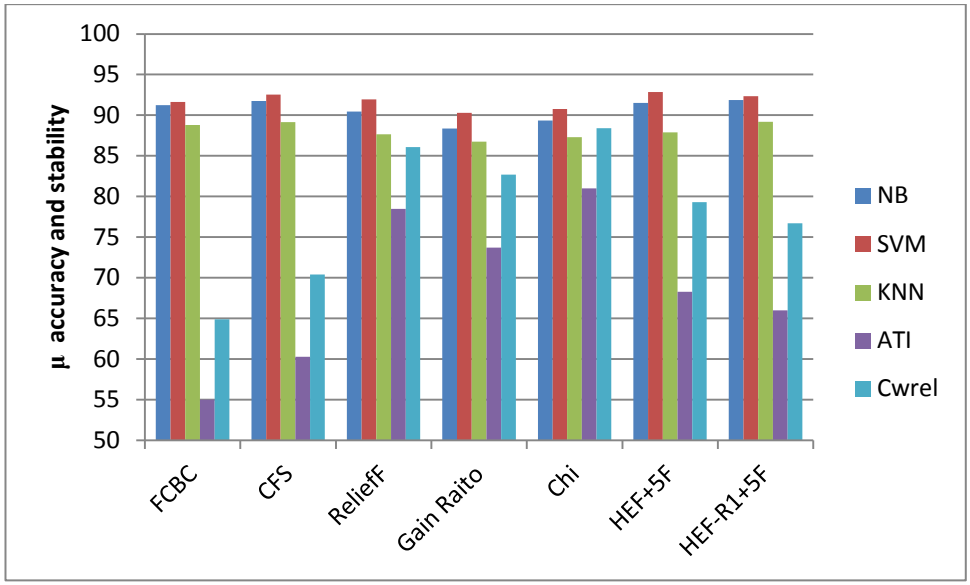


Figure 6. 13: Average accuracy and stability of HEF+5F and 5 filter members on 10 real dataset, focusing on each FS technique

Figures 6.12-6.13 show the comparison between the HEF+5F and their filter members in terms of the average accuracy and stability (μ performance) on 10 real datasets. We should note that the stability measures have scores between 0 and 1, but we multiply these score values by 100 in order to make the comparison clearer between the accuracy and the stability, and to be able to present them in one figure. It can be seen that in most cases there is no clear winner among the filter members. As a result, there is no filter method that satisfies both accuracy and stability. For example, SF (FCBF and CFS)

performs well regarding the accuracy but performs very poorly regarding the stability. On the other hand, RF (ReliefF, Gain Ratio and Chi-x^2) performs well regarding the stability but performs less well regarding the accuracy.

Therefore, we can conclude that each of these filters has its own strong and weak points and neither one can satisfy both accuracy and stability. However, our ensemble approach (HEF+5F and HEF-R1+5F) was able to identify important features which helped the classifiers to perform well and produce to trade-offs regarding stability between SF and RF. So, it produced more accurate and reliable results and gave more confidence in the final results.

6.6.2.3 Time Complexity Analysis

In this section, we have presented the time complexity of our experiments theoretically and experimentally by using the big O notation, in addition to measuring the execution time needed to run each filter and then to build the classifier. This is an important consideration in order to compare the computational performances of each filter member in the ensemble and the model building phase.

The complexity of our ensemble algorithms proposed in this research can be divided into two phases: run time of the filter members in the ensemble and the run time of the aggregation step.

The run time of the filter depends on the filters used as members in the ensemble, and we considered this issue from the initial framework by selecting fast filters, especially with SF. Let N be the number of features in the dataset and S the number of samples.

In terms of SF, we selected FCBF which has a best case complexity O(N), when only one feature is selected and a worse case complexity O(N^2), when all features are selected, which are comparable to subset evaluation by greedy sequential search. But in general cases when K (1<K<N) features are selected, the number of evaluations performed will be much less than with a greedy sequential search, because the features removed in each round are not considered in the next round (Yu and Liu, 2004). The second SF used is CFS, which uses linear forward selection (LFS) search instead of best-first search which runs much faster to produce similar results in the initial

experiments. In the classical greedy sequential search, the number of evaluations grows quadratically with the number of features N. Thus the upper bound on the number of evaluations is $\frac{1}{2} \times N(N+1)$. Using LFS reduces the upper bound on the number of evaluations to $\frac{1}{2} \times K(K+1)$, regardless of the original number of features (Gutlein et al., 2009). Whereas, in terms of RF, the time complexity is O(N), except ReliefF which is O(N.S). So the complexity of running 5 filters will be equal to O(N^2) in a worse case and equal to O(N.S) in a best case.

The time complexity of the aggregation step will be O(K^2), where K is the number of features selected from the original dataset. K<<<N can be considered negligible compared to N and it can be said that the complexity of the ensemble O(K^2) is smaller than O(N) in a dataset with high dimensionality.

In the following three tables (6.7-6.9) the running time for each filter has been recorded using three classifiers. This test was repeated 10 times to give the average execution time required to run each filter and to build the classifier. The running time includes the filter's time (t_F) and the classification model's generation time (t_C).

Table 6.7: Running time (seconds) for each filter with NB classifier

Runtime NB	**FCBF**	**CFS**	**ReliefF**	**Gain Ratio**	**Chi-x²**
Zoo	0.01	0.01	0.01	0.01	0.01
Dermatology	0.02	0.03	0.07	0.02	0.02
Promoters	0.01	0.01	0.02	0.01	0.01
Splice	0.14	0.13	5.96	0.08	0.08
M-feat-fact	0.94	1.07	10.21	0.58	0.56
Arrhythmia	0.15	0.2	0.82	0.14	0.15
Colon	2.3	2.46	2.37	2.29	2.29
SRBCT	3.33	3.39	3.22	3.0	3.05
Leukaemia	25.39	26.5	25.77	25.33	25.39
Ovarian	135.25	140.6	143.05	131.76	131.55
Average	16.754	17.44	**19.15**	16.322	16.311
Average- Ovarian[6]	3.58	3.75	**5.38**	3.49	3.50

[6] Average running time of all datasets used except Ovarian.

Table 6.8: Running time (seconds) for each filter with KNN classifier

Runtime KNN	FCBF	CFS	ReliefF	Gain Ratio	Chi-x^2
Zoo	0.01	0.01	0.01	0.01	0.01
Dermatology	0.02	0.03	0.07	0.02	0.02
Promoters	0.01	0.01	0.02	0.01	0.01
Splice	1.03	1.03	6.85	0.95	0.95
M-feat-fact	1.13	1.28	10.42	0.75	0.74
Arrhythmia	0.14	0.19	0.8	0.13	0.14
Colon	2.35	2.51	2.41	2.35	2.35
SRBCT	3.45	3.69	3.61	3.39	3.39
Leukaemia	30.52	31.25	30.91	30.47	30.51
Ovarian	119.6	124.8	134.7	115.28	117.8
Average	15.82	16.48	**18.98**	15.33	15.59
Average- Ovarian	4.295	4.44	**6.12**	4.23	4.23

Table 6.9: Running time (seconds) for each filter with SVM classifier

Runtime SVM	FCBF	CFS	ReliefF	Gain Ratio	Chi-x^2
Zoo	0.04	0.04	0.04	0.03	0.04
Dermatology	0.06	0.05	0.09	0.04	0.04
Promoters	0.01	0.01	0.02	0.01	0.01
Splice	1.66	1.6	6.9	8.48	8.57
M-feat-fact	0.97	1.08	10.26	0.58	0.57
Arrhythmia	0.26	0.29	0.92	0.25	0.24
Colon	2.71	2.76	2.48	2.42	2.4
SRBCT	3.4	3.64	3.57	3.34	3.35
Leukaemia	30.83	31.67	31.2	30.79	30.83
Ovarian	144.6	146.8	151.7	139.28	139.8
Average	18.454	18.794	**20.718**	18.522	18.585
Average- Ovarian	4.437	4.571	**6.164**	5.104	5.116

Also, Figure 6.14 shows the average runtime performances of 9 real datasets (excluding Ovarian) using three classifiers. We can observe that ReliefF has the highest runtime in seconds with the three classifiers, while Gain Ratio and Chi-x^2 have less run time on average than FCBF and CFS by using NB and KNN. In contrast, FCBF and CFS have less run time on average than Gain Ratio and Chi-x^2 using SVM. This figure clearly shows that ReliefF demonstrates an unexpectedly slow performance although its time complexity is linear to dimensionality. The reason is that searching for nearest

neighbours in ReliefF involves a distance calculation which is more costly than the calculation of Gain Ratio and Chi-x^2(Yu and Liu, 2004). Based on these results, we can say that SF spends a similar time to RF and is some times faster than some of them, such as ReliefF. This means that we can gain the advantage of using the SF and at the same time we can overcome the time complexity issues.

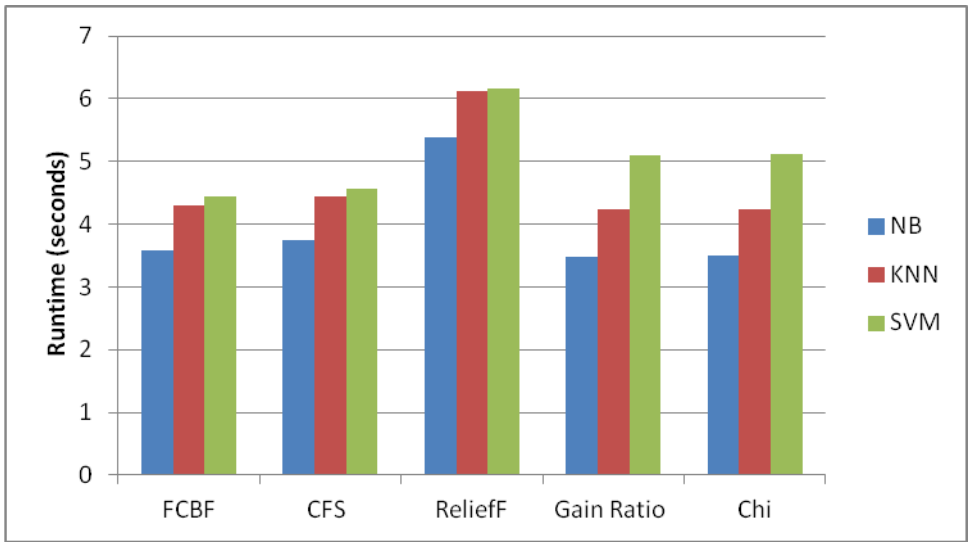


Figure 6.14: Average runtime performances of 9 real datasets (excluding Ovarian) using three classifiers

6.6.3 Changing the Aggregation Method for Combining Feature Subsets:

In this section, we have made a comparison between three different schemes of mean rank aggregation with partial list. The first one ranks the features based on the frequency. Then if there are some features having equal frequencies, we ranked them by means of these features and we made each feature not appear in the list; the position was equal to K+1, where K is the maximum number of features in the partial list. We represented this scheme as HEF-a. The second one ranks the features based on the mean and we made each feature not appear in the list; the position was equal to K+1. We represented this scheme as HEF-b. The third one had ranked the features based on mean and we made each feature not appear in the list; the position was equal to K+1. Then we divided the mean of each feature by the frequency of this feature, and we represented this scheme as HEF-c.

6.6.3.1 Accuracy Evaluation

Tables 6.10-6.12 show the average test accuracies of NB, KNN and SVM classifiers, respectively. The results in these three tables reveal similar patterns as follows: firstly, HEFb-75% has the highest average accuracy among other schemes with all classifiers including using all the selected features. Secondly, there are small differences between the three schemes within the same number of features. Thirdly, in general, the classification accuracy by selecting top 75% of the features produced higher values than selecting the top 50% ,and the top 50% of the features produced higher values than selecting the top 25% of the features.

Table 6.10: The accuracies of NB models trained with three different schemes of mean rank aggregation

Dataset NB	HEF	HEFa-75%	HEFb-75%	HEFc-75%	HEFa-50%	HEFb-50%	HEFc-50%	HEFa-25%	HEFb-25%	HEcF-25%
Zoo	94.46	93.67	94.93	**93.95**	89.69	91.58	90.08	82.98	88.41	88.41
Dermatology	97.79	97.57	**98.14**	97.65	89.22	90.6	89.41	84.14	83.66	84.31
Promoters	91.6	92.08	92.64	92.08	93.78	**94.56**	94.36	83.32	83.32	83.32
Splice	96.2	96.18	96.18	**96.22**	95.49	95.43	95.48	93.82	93.74	93.74
M-feat-fact	92.59	92.69	92.7	92.69	92.28	**92.77**	92.28	91.32	91.58	91.32
Arrhythmia	66.97	**66.69**	66.66	**66.69**	65.51	66.11	66.51	62.92	63.23	62.92
Colon	84.05	84.38	84.24	84.38	85.09	**85.4**	85.09	84.71	85.33	84.71
SRBCT	**99.53**	98.79	99.04	99.04	98.43	98.45	98.67	96.61	96.74	96.39
Leukaemia	95.82	96.09	**96.21**	**96.21**	**96.21**	96.09	**96.21**	95.52	95.52	95.39
Ovarian	98.33	98.21	98.34	98.29	**98.38**	98.53	98.3	97.98	98.13	97.98
Average	91.73	91.63	**91.91**	91.72	90.41	90.95	90.64	87.33	87.97	87.849

Table 6.11: The accuracies of KNN models trained with three different schemes of mean rank aggregation

Dataset KNN	HEF	HEFa-75%	HEFb-75%	HEFc-75%	HEFa-50%	HEFb-50%	HEFc-50%	HEFa-25%	HEFb-25%	HEFc-25%
Zoo	**96.44**	95.93	95.93	95.73	95.63	93.45	95.83	84.85	90.39	90.39
Dermatology	95.8	95.87	**96.14**	95.71	89.16	90.26	89.44	82.83	83.41	83.6
Promoters	85.97	88.34	88.44	88.34	89.01	88.43	**89.2**	84.09	84.09	84.09
Splice	79.4	82.12	81.82	82.03	84.29	84.39	84.29	89.53	**89.61**	**89.61**
M-feat-fact	**96.24**	96.1	96.08	96.1	95.74	95.94	95.74	95.21	95.23	95.19
Arrhythmia	56.17	56.44	56.82	56.44	56.49	56	56.49	**58.12**	57.86	**58.12**
Colon	78.29	78.69	78.52	78.69	77.88	77.5	77.88	**80.93**	80.26	**80.93**
SRBCT	99.5	99.51	99.4	99.51	**99.76**	99.64	99.64	99.56	99.67	99.45
Leukaemia	94.75	95.02	**95.57**	95.3	95.84	95.7	94.59	91.73	92	90.89
Ovarian	99.52	99.56	99.48	99.56	**99.64**	99.56	99.6	98.98	99.37	98.98
Average	88.21	88.76	**88.82**	88.74	88.34	88.09	88.27	86.58	87.19	87.13

Table 6.12: The accuracies of SVM models trained with three different schemes of mean rank aggregation

Dataset SVM	HEF	HEFa-75%	HEFb-75%	HEFc-75%	HEFa-50%	HEFb-50%	HEFc-50%	HEFa-25%	HEFb-25%	HEFc-25%
Zoo	95.74	**96.83**	**96.83**	**96.83**	95.34	93.45	95.34	91.09	91.09	91.09
Dermatology	97.29	97.16	**97.57**	97.05	88.48	90.35	88.46	83.27	84.07	84.07
Promoters	90.12	91.56	91.54	91.54	93.86	**94.25**	**94.25**	81.13	81.13	81.13
Splice	95.68	**95.89**	95.79	95.88	95.72	95.69	95.73	94.56	94.43	94.43
M-feat-fact	**97.76**	97.52	97.5	97.52	96.98	97.16	96.98	96.08	96.07	96.08
Arrhythmia	**69.23**	67.5	67.48	67.5	65	64.94	65	61.88	61.84	61.88
Colon	**87.29**	86.83	86.83	86.83	85.26	85.83	85.26	83.93	84.1	83.93
SRBCT	99.75	**100**	99.89	**100**	99.42	99.78	99.54	99.53	99.53	99.53
Leukaemia	96.39	96.68	96.55	96.68	**97.07**	96.5	**97.07**	95.54	95.5	95.27
Ovarian	**100**	**100**	**100**	**100**	**100**	**100**	**100**	99.44	99.68	99.44
Average	92.93	92.997	**92.998**	92.98	91.71	91.79	91.76	88.65	88.74	88.69

As we can see in Figure 6.15, the Nemenyi test identified three groups by evaluating the accuracy of NB, and the accuracy of HEFb-75% is significantly better than HEFa-25% and HEFc-25%. HEFc-75%, HEFb-50%, HEF, HEFa-75%, HEFc-50%, HEFa-50% and HEFb-25% belong to all the groups. On the other hand, the difference in accuracy of KNN and SVM was not significant and the Friedman test could not reject the null hypotheses.

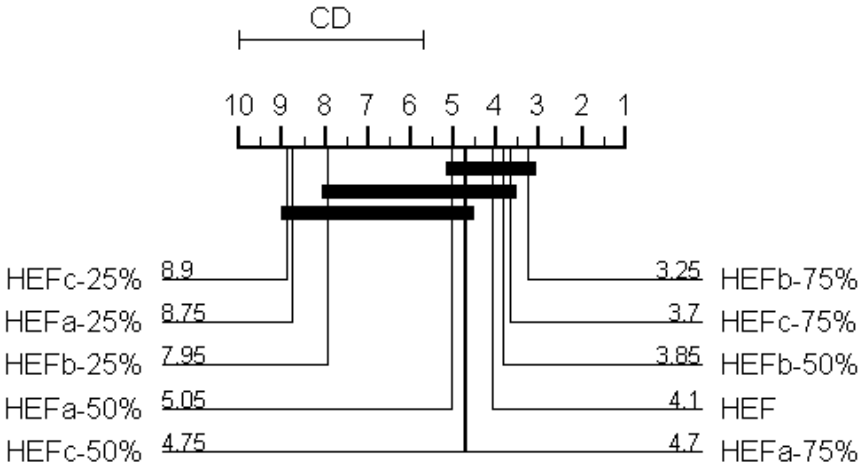


Figure 6. 15: Results of the Nemenyi test was used to evaluate the accuracy of NB of three different schemes of mean rank aggregation against each other

Table 6.13 and Figure 6.16 show the averages of test accuracy for each scheme and classifier, independent of the dataset. We can see that the highest accuracy in the three classifiers was achieved by HEFb-75%. Also, HEFb-50% and HEFb-25% achieved the highest accuracy in three classifiers, except in one case when HEFa-50% obtained the highest accuracy by using the KNN classifier.

Table 6.13: Average test accuracy over 10 real datasets with three different schemes of mean rank aggregation focusing on the three classifiers

	NB	**KNN**	**SVM**
HEF	91.73	88.21	92.93
HEFa-75%	91.63	88.76	92.997
HEFb-75%	**91.91**	**88.82**	**92.998**
HEFc-75%	91.72	88.74	92.98
HEFa-50%	90.41	**88.34**	91.71
HEFb-50%	**90.95**	88.09	**91.79**
HEFc-50%	90.64	88.27	91.76
HEFa-25%	87.33	86.58	88.65
HEFb-25%	**87.97**	**87.19**	**88.74**
HEFc-25%	87.849	87.13	88.69

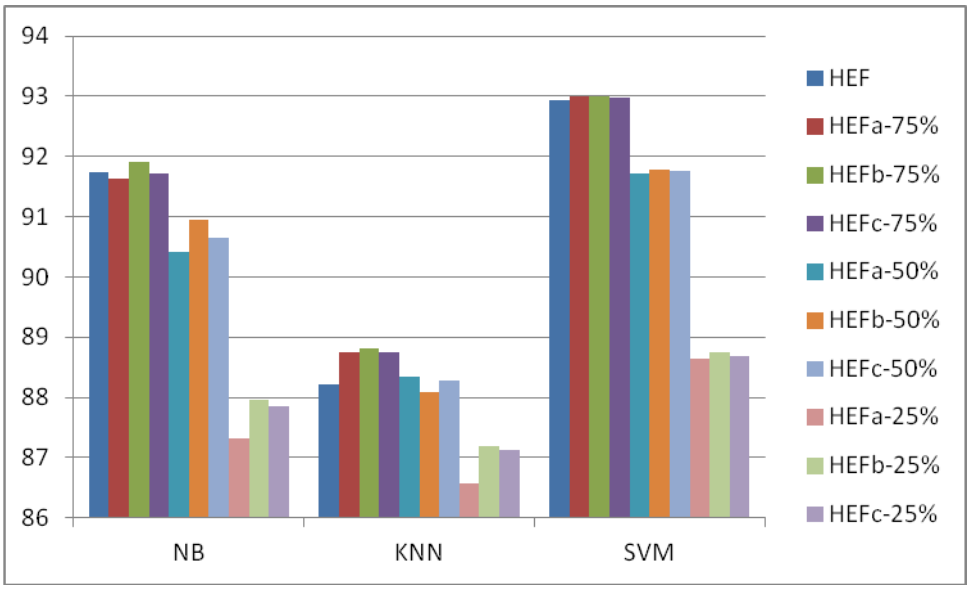


Figure 6. 16: Average test accuracy over 10 real datasets with three different schemes of mean rank aggregation focusing on the three classifiers

6.6.3.2 Stability Evaluation

In this section, we discussed the stability of the three different schemes of mean rank aggregation and compared the different numbers of selected features between them.

Table 6.14: The stability measures of ATI with three different schemes of mean rank aggregation

ATI	HEF	HEFa -75%	HEFb -75%	HEFc -75%	HEFa -50%	HEFb -50%	HEFc -50%	HEFa -25%	HEFb -25%	HEFc -25%
Zoo	0.93	0.9	**0.97**	0.91	**0.97**	**0.97**	**0.97**	0.84	0.81	0.81
Dermatology	0.94	0.94	0.92	0.92	0.96	0.92	0.95	0.65	0.76	0.72
Promoters	0.71	0.78	0.78	0.78	0.83	0.86	0.84	0.8	0.8	0.8
Splice	0.8	0.89	0.88	0.9	0.9	0.9	0.91	0.92	0.91	0.91
M-feat-fact	0.82	0.78	0.78	0.78	0.76	0.74	0.76	0.67	0.7	0.67
Arrhythmia	0.71	0.67	0.67	0.67	0.66	0.65	0.66	0.53	0.55	0.53
Colon	0.49	0.49	0.49	0.49	0.46	0.48	0.46	0.57	0.58	0.57
SRBCT	0.6	0.57	0.576	0.57	0.52	0.54	0.52	0.41	0.43	0.41
Leukaemia	0.47	0.44	0.45	0.44	0.38	0.38	0.38	0.3	0.31	0.31
Ovarian	0.55	0.52	0.53	0.53	0.55	0.54	0.55	0.76	0.75	0.76
Average	0.702	0.698	**0.7046**	0.699	0.699	0.698	0.7	0.645	0.66	0.649

Table 6.15: The stability measures of CWrel with three different schemes of mean rank aggregation

CWrel	HEF	HEFa -75%	HEFb -75%	HEFc -75%	HEFa -50%	HEFb -50%	HEFc -50%	HEFa -25%	HEFb -25%	HEFc -25%
Zoo	0.94	0.9	1	0.91	1	1	1	0.85	0.83	0.83
Dermatology	0.85	0.93	0.91	0.91	0.99	0.94	0.98	0.71	0.82	0.78
Promoters	0.81	0.86	0.86	0.86	0.92	0.94	0.93	0.87	0.87	0.87
Splice	0.82	0.92	0.91	0.93	0.94	0.94	0.94	0.97	0.97	0.97
M-feat-fact	0.84	0.83	0.84	0.83	0.85	0.82	0.85	0.8	0.82	0.8
Arrhythmia	0.81	0.78	0.79	0.78	0.79	0.78	0.79	0.68	0.69	0.68
Colon	0.65	0.64	0.79	0.64	0.61	0.78	0.61	0.7	0.69	0.7
SRBCT	0.78	0.79	0.65	0.79	0.68	0.63	0.68	0.71	0.72	0.71
Leukaemia	0.66	0.61	0.79	0.61	0.63	0.67	0.63	0.68	0.72	0.69
Ovarian	0.71	0.68	0.61	0.69	0.71	0.63	0.7	0.86	0.72	0.86
Average	0.787	0.794	**0.815**	0.795	0.812	0.813	0.811	0.783	0.785	0.789

Table 6.14 and 6.15 showed how each scheme of mean rank aggregation had different stability within the same dataset, thus, it is apparent that some schemes were more stable than others when the samples had been changed. As we can see, HEFb-75% showed a higher average stability for all the datasets. However, the stability results were not significantly different from each other and the Friedman test could not reject the null hypotheses.

6.7. Conclusion

In this chapter, we aimed to improve the results of HEF in terms of accuracy and stability. Therefore, we tried three ideas as follows:

- Firstly, we applied three types of wrapper after the HEF in order to select the most important features without sacrificing stability or accuracy. The result showed that adding the wrapper after HEF led to the selection of a very few number of features, but reduced the accuracy and stability. Therefore on balance this idea was demonstrated to be ineffective.

- Secondly, we added Chi-x^2 filter as a new member in the HEF, in addition to the four filters used previously. We had added Chi-x^2 filter after dividing the filter methods into five groups in order to gain a deeper understanding, then

selected appropriate filters from each group. The result showed that HEF+5F and HEF-R1+5F achieved the best result but never delivered the worst. Also, in terms of stability, HEF+5F and HEF-R1+5F showed that they improved the stability more than HEF+4F and HEF-R1+4F, because Chi-x^2 had shown a higher average in stability for all the datasets. This indicates that adding more stability members leads to an increase in the stability of the ensemble. So HEF+5F and HEF-R1+5F are the more reliable, as well as more accurate.

- Thirdly, we had investigated other aggregation methods, specifically three schemes of mean rank aggregation which dealt with the top-K list. The comparison result confirmed that ranking the feature based on mean and making each feature not appeared in the list, with position equal to K+1 (HEF-b) is the best scheme in most cases in terms of accuracy and stability.

In conclusion, the second and the third ideas will be used in the following chapters because they have improved the results of HEF in terms of accuracy and reliability. On the other hand, we will exclude the first idea which adds the wrapper after HEF because it did not improve the results of HEF in terms of accuracy and stability.

Chapter 7

Weighted Heuristic Ensemble of Filters

7.1 Introduction

In the previous chapter, we attempted to improve the HEF by adding more filters as members in the HEF and extending the HEF with a wrapper, aiming to reduce the number of features selected while preserving the same accuracy and stability. Furthermore, we changed the aggregation method from counting the frequency of each feature selected to mean rank aggregation, by sorting the selected features based on the means of their ranks in all the ranking filters.

Intuitively speaking, it is reasonable to assume that the filters should be treated differently in accordance with their performance, as in reality, there are some differences in the performances of filters. Thus, the use of different weights for calculating the total scores of the selected features may improve the performance. In this chapter, we will investigate the effect of changing the weight for each filter in an ensemble. Our hypothesis is that weighting the members in an FSE differently based on their performance should lead to some improvement of the performance of the FSE. To the best of my knowledge, so far this is the first study that gives weight to filter methods based on a validation set, or by using prior knowledge when aggregating the output of the filters in the ensemble. The work in this chapter has been published at the Intelligent Systems Conference 2015.

The rest of this chapter is organised as follows: Section 2 presents related work which roughly considers three main topics, namely illustrating the application using the idea of supervised rank aggregation, applying weight to some rankers by analysing some researches, and the limitations of these researches. Section 3 describes the frameworks of adding fixed weight, variable weight and selective filters. Section 4 gives the results and evaluates the three proposed approaches. Finally, Section 5 and 6 evaluate and conclude our work.

7.2 Related Work

The rank aggregation technique has been investigated and used in some application areas, such as metasearch, image fusion and others. It usually determines the weights of each ranking list by learning an aggregation function using training data (Liu et al.,

2007, Lillis et al., 2006). For example, in a meta-analytic bioinformatics study, some labs are more efficient in the data collection and analysing procedure than other labs; also, in a metasearch study, more capacity and accuracy could be found while using some search engines than with others. Moreover, some judges are found to be more experienced and impartial than others in a competition and some base rankers could be found to be incomprehensible or even misleading in some extreme cases.

Aslam and Montague (2001) proposed two algorithms based on Borda Count for metasearch, namely Borda-fuse and Weighted Borda-fuse. Borda-fuse gives the same weight to all engines, whereas Weighted Borda-fuse uses different weights. This is an earlier study that gives different base rankers different weights by using labelled training data. For instance, the weights can be determined by using the MAP (Mean Average Precision) of the base rankers. So, in order to determine the precision value of each engine, training data is required by Weighted Borda-fuse. Training details not required by Borda-fuse, as the rank results can be directly unified by the base rankers' score. It has been observed from experimental results that Weighted Borda-fuse is indeed superior to Borda-fuse. However, Weighted Borda-fuse has the problem of calculating the weights of the ranking list independently, using heuristics. It is also unclear whether the same concept can be applied to other methods (Liu et al., 2007). The authors themselves pointed out that it may not always be optimal to use precision values as weight. The ideal condition would be to fine tune the weight vector used by the Borda Count by means of certain techniques. The results will reveal the potency of using precision values as weights. Also, another limitation of the Borda Count and the Weighted Borda-fuse model is that there is no clear way of handling missing documents (De et al., 2012).

Lin and Ding's (2009) method appears to be one of the few available methods to consider the different quality of base rankers. However, one obvious limitation of this approach is that no systematic and principled strategies are available for designing a proper weighting scheme when facing a practical problem. A good weighting scheme may be learned by supervised rank aggregation.

Liu et al. (2007) deal with supervised rank aggregation (SRA). In their procedure, the training data is provided in the form of the true relative ranks of some entities, and the weights are optimised with the support of the training data as well as the aggregated list. Instead of pre-specified constants, the weights are generally treated as parameters in

these models. The unavailability of any training data in many applications is a problem of SRA.

In the biomedical applications of computational biology, Abeel et al. (2010) discussed the robustness of ensemble feature selection by using the embedded method, support vector machine-recursive feature elimination (SVM-RFE), then obtaining different rankings by bootstrapping the training data. They used two aggregation methods: complete linear aggregation and complete weight linear aggregation. The complete linear aggregation uses the complete ranking of all the features to produce the ensemble result by summing the ranks, over all bootstrap samples and setting all weights equal to one. On the other hand, complete weight linear aggregation measures the weights of the scores of each bootstrap ranking using AUC. AUC is obtained by linear SVM, trained on the bootstrap samples and evaluated on the out-of-bag (OO) samples, and the amount of the weight is measured as w_i=OO - AUC_i.

Although greater accuracy can be achieved by supervised aggregation, the labelled data are not always available in practice (Wang and Li, 2012). Also, a prudent way of handling the quality difference is assigning weights to base rankers; in practice, designing a proper weight specification scheme can be rather difficult, especially when the availability of prior knowledge on the base rankers is poor (Deng et al., 2014).

7.3 Weighted Heuristic Ensemble Filters (WHEF)

In this section, three methods are proposed. The first one assigns a fixed weight to some filters, and the second one assigns variable weights to some filters in order to investigate the impact of weighted filters on the final result of the ensemble aggregation. The third one assigns a weight equal to 1 to some filters and assigns a weight equal to zero to other filters, which means in other words that it selects some filters and discards others based on the validation set.

We will start the experiment by adding more weight to the subset filters and less weight to RF; the justification for that is in Section 7.3.1. Then, in the second method we change the strategy by adding more weight to some filter members in the HEF based on a higher classification accuracy of individual filters using the validation set. Finally, we

select the two top filters only to aggregate their feature selected results and to disregard the results of the three remaining filters.

We first give some definitions and notations. Given a set of features X, let X_i be a subset of X and assume that there is a ranking order among the features in X_i. Consider an ensemble consisting of l filters, then we assume each filter F_i provides a feature ranking $f_i = \{f_i^1, f_i^2 \ldots . f_i^{X_i}\}$, all the rankings are aggregated into a consensus feature ranking f_E by a weighted voting function.

$$f_E = E\left(\sum_{i=1}^{l}(w_i\ f_i)\right) \qquad (7.1)$$

Where, E( ) is the aggregating function of an ensemble, w_i denotes a weight coefficient. If we assume that all of the filters are equally important then set $w_i = 1$ for i=1, ..., l, thus $f_i = (w_i\ f_i)$ will be the same as in our previous chapters.

By assigning different weight values to different filters, filter F_i with a larger weight should play a more important role in generating the consensus feature ranks.

7.3.1 Fixed Weight Methods (FWHEF)

In this section, we give more weight to subset filters (SF) and less weight to rank filters (RFs) in order to allow SF to play a more important role in generating the consensus feature ranks. The reason for adding more weight to SF is that many SF methods have been demonstrated to be efficient in removing both irrelevant and redundant features. In such SF methods, the existence and effect of redundant features are also taken into account to approximate the optimal subset (Yu and Liu, 2004, Hall, 2000, Koller and Sahami, 1996). RF methods are not designed for removing redundant features because they evaluate each feature individually. As a result, a similar ranking is likely to be found for redundant features. For instance, a large number of redundant features can be found for high-dimensional data which is far from the optimal (Yu and Liu, 2004)

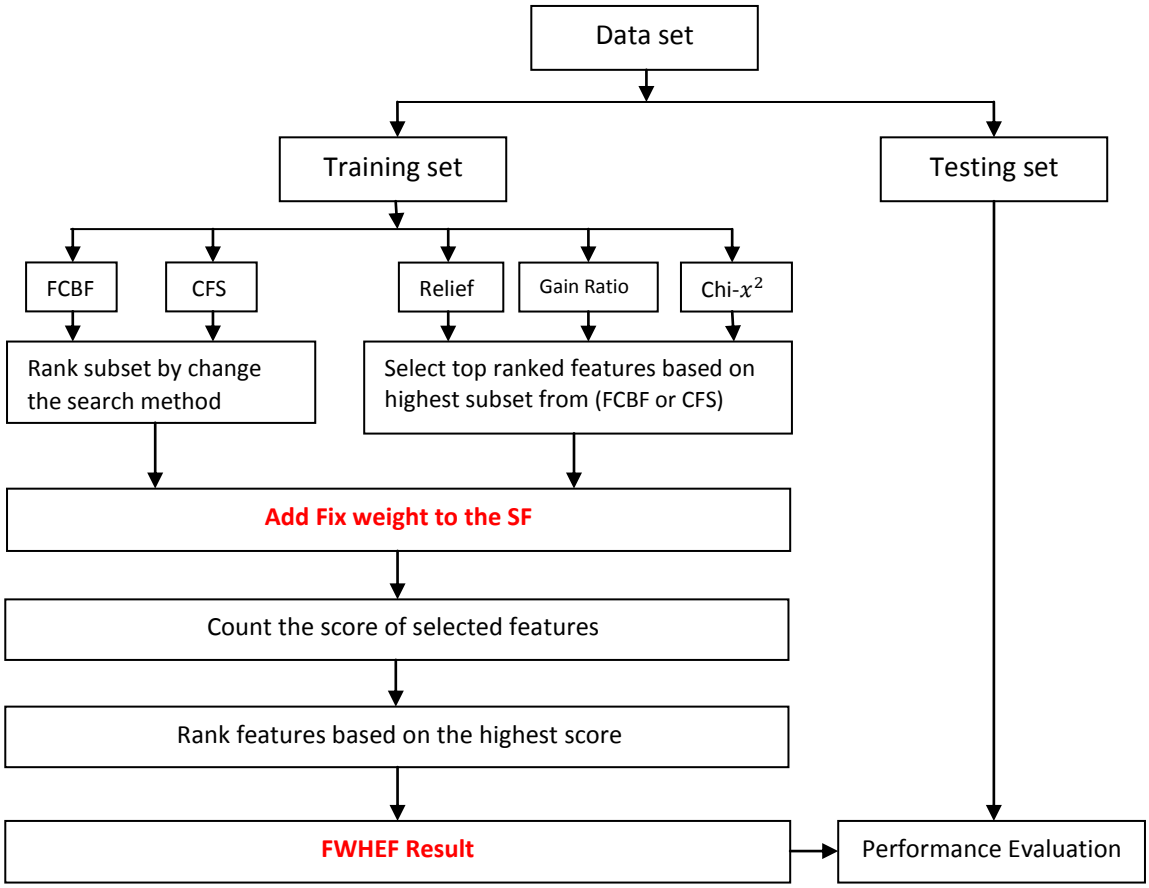


Figure 7.1: Framework of FWHEF

The framework of Fix Weight HEF (FWHEF) illustrated in Figure 7.1. However, how to decide the appropriate weights for SF and RF is not an easy task, as no prior knowledge on filters is available, no training sets can be used, and so we select different values as a weight in the following systematic manner:

$$f_{E1} = E\ (\sum_{i=1}^{l}(\ w_i\ f_i\ )) \qquad (7.2)$$

$$\text{S.T.}\ \sum_{i=1}^{l} w_i = 1 \qquad (7.3)$$

where E1 is the aggregating function of FWHEF and each filter F_i is assigned a weight w_i, where f_i is the same as that in (7.1).

$$w_i = \begin{cases} \beta_i\ , & if\ \ F_i \rightarrow SF \\ \lambda_i\ , & if\ \ F_i \rightarrow RF \end{cases} \qquad (7.4)$$

where β is a coefficient generated to give more weight to the feature selected by SF, and λ is another coefficient generated to give less weight to the feature selected by RF, and the sum of these two coefficients is equal to one. We start with β_1, then add each β_{i+1} by $\Delta\beta$ and so on, and also start λ with λ_1, then add each λ_{i+1} add by $\Delta\lambda$ and so on, as follows:

$$\beta_{i+1} = \beta_i + \Delta\beta \quad (7.5)$$

$$\lambda_{i+1} = \lambda_i - \Delta\lambda \quad (7.6)$$

The first case starts by $w_i = \beta_1 = 0.2$,if $F_i \rightarrow SF$ and $w_i = \lambda_1 = 0.2$ if $F_i \rightarrow RF$, which means there has been equal accuracy, and this will give us the same results as HEF which is adding an identical weight equal to one to each filter in the HEF. But, from the second case, we start to increase β_i by $\Delta\beta = 0.075$ to give more weight to the feature selected by SF, and to decrease λ_i by $\Delta\lambda = 0.05$ to give less weight to the feature selected by RF. Then, we carry on the experiments in a systematic manner by increasing β_i by $\Delta\beta = 0.075$ and decreasing λ_i by $\Delta\lambda = 0.05$. Appendix B shows the results of different cases, however the statistical test shows that there are no significant differences between them, therefore, we select the middle case, in which $w_i = \beta_3 = 0.1$,if $F_i \rightarrow SF$ and $w_i = \lambda_3 = 0.35$ if $F_i \rightarrow RF$ as a fixed weight value of FWHEF to compare it with other weighted approaches in this chapter.

7.3.2 Variable Weight Based on Validation Set (VWHEF)

In this section, we discuss how to apply variable weight on some filters based on the classification accuracy, by assuming that if a filter produces a high accuracy it means that it can select more relevant and important features and vice versa, using the same classifier. Variable Weighted HEF (VWHEF) uses the classification accuracy values to compute the weights of each filter, so a training set is required. Figure 7.2 illustrates how the training data was split into training and validation sets in order to evaluate the accuracy for each of the individual filters. The experiments were performed through 10-fold cross-validation. We split the training set into 10 subsets, used 9-folds for training and 1-fold for validation, then rotated this process 10 times to create 10 datasets. We then took the average classification accuracy over the 10 validation sets as the final results of each filter. This process is repeated in each fold of the external 10-fold cross

validations which evaluate the VWHEF by using a test set after adding different weights to some filters, as seen in Figure 7.3.

Since we should not use the test set to determine which filters have the higher accuracy to give them more weight, the reason for that is to avoid bias, we use the validation set to estimate the accuracy on the test set. Also, we take the average accuracy of 10 validation sets to produce more reliable results than using just one validation set.

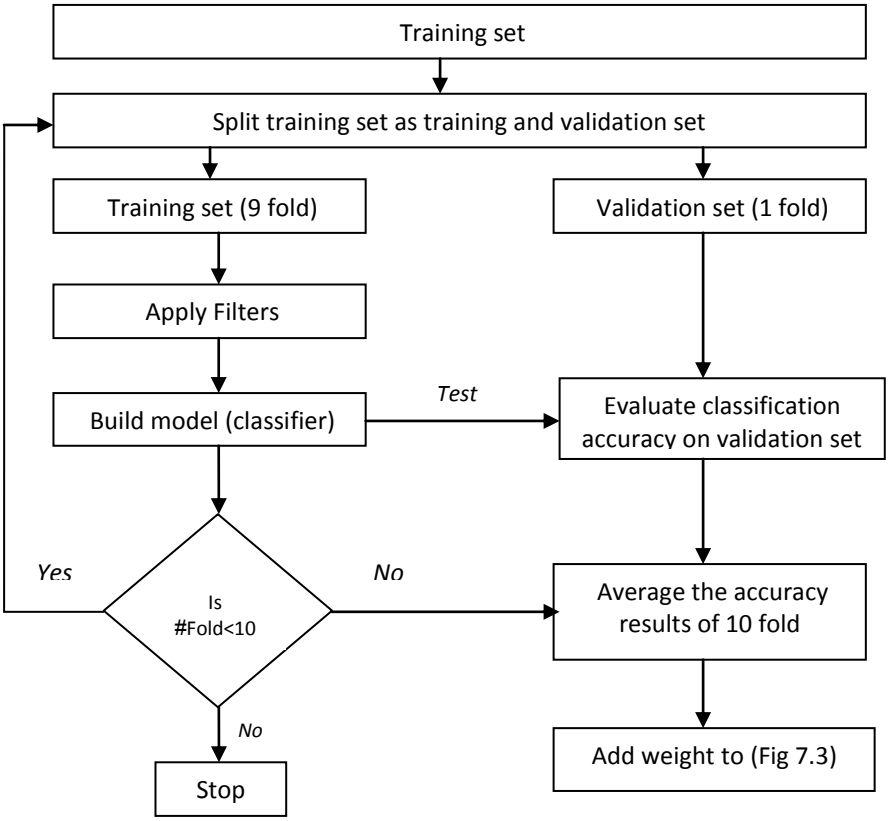


Figure 7.2: Determining the weight by classification accuracy on the validation dataset

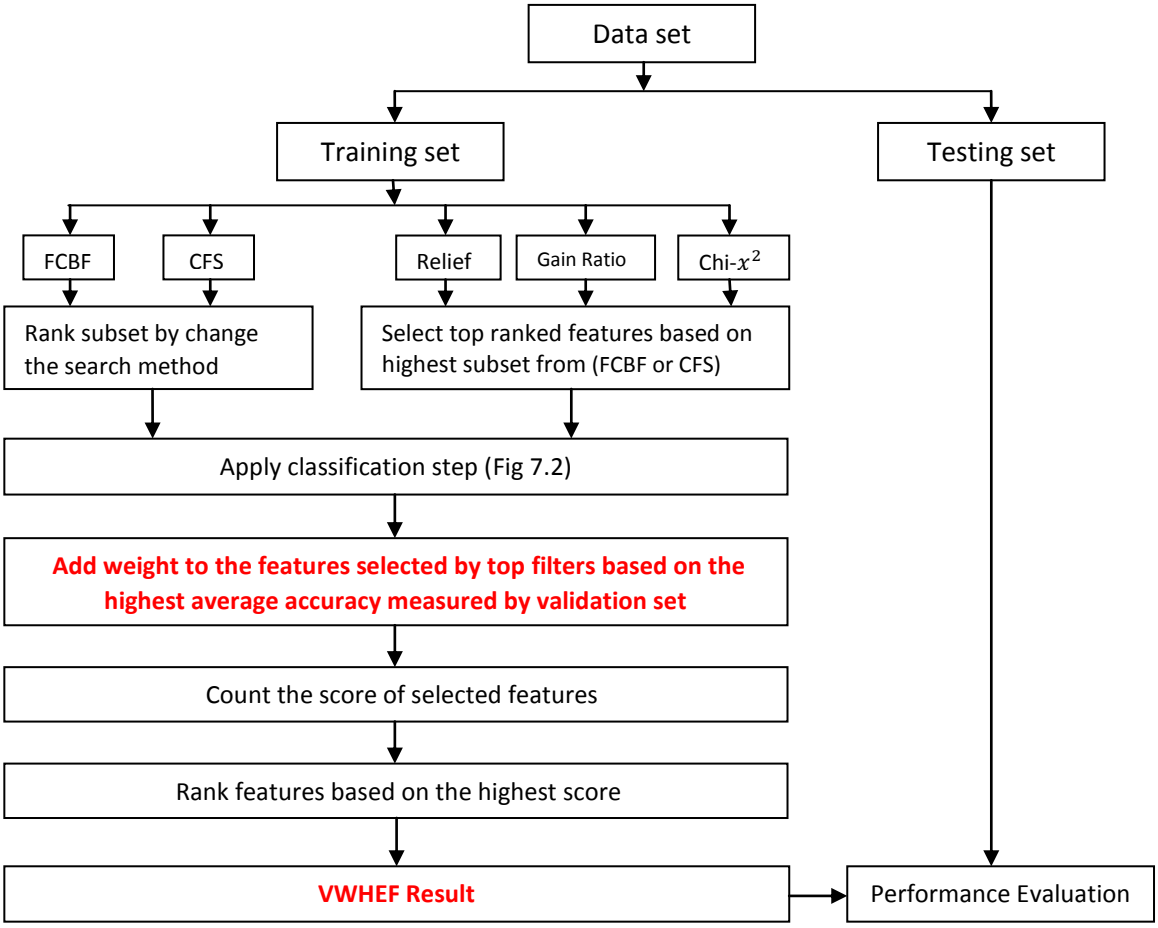


Figure 7.3: Framework of VWHEF

Variance-based Weight Estimation

We design a heuristic method to compute the weight based on the classification accuracy and variance on the validation set, because there are no standard methods to compute the weight. Aslam and Montague (2001) mentioned that it may not always be optimal to use classification precision values as weight (Tongchim et al., 2007). Accordingly, in order to calculate the weight of each filter in VWHEF, we need to find values which have a relation with the accuracy from each filter, giving more weight to filters with high accuracy and low weight to filters with low accuracy. Note that the weights based on classification accuracy range between 100 and 0, which is not the perfect way to use this accuracy directly as weight. Thus, we use standard deviation σ between the average accuracy of each filter as a measure to evaluate how far the accuracy of these filters differs. If σ is high this means that there are big accuracy differences between the filters, which is a motivation to give high weight to the filter

with highest accuracies and vice versa. If σ is low, this means that there is a small accuracy difference, or in other words all filters produce similar results and there is no need to give high weight to the highest filter accuracy. So, based on this justification we use σ as a weight value to the highest filter accuracy. With the same idea, we compute the weight of the second highest accuracy filter, but this time we want the second weight to become smaller than the first one. Therefore, we first measure the difference between the highest filter accuracy and the second one, and then take off this difference from the σ, but if the second weight becomes less than 1 then the weight will be 1. The remaining filters determine the weight of the second filter in a similar way. The framework to compute the weight is illustrated in Procedure 7.1 which can be described as follows: Firstly, all filters are ranked based on the final accuracy of the validation set. Secondly, the standard deviation σ between the final accuracy of the all filters is computed using the validation set. Thirdly, the first weight w_1 is set equal to the σ, but if σ < 1 then w_1 = σ+1. Fourthly, the order position of each feature selected by the highest filter is multiplied by w_1. Fifthly, all the remaining filters from the second to the last use the same weighting formula: subtracting the current "filter accuracy" from the highest "filter accuracy" : Δacc = acc (F_{i-1}) - acc (F_i). Then, the weight w_i is assigned a value as follows: $w_i = w_{i-1} - \Delta acc$, if $w_i < 1$ then $w_i = 1$. Finally, the order position of each feature selected by the current filter is multiplied by w_i.

Procedure 7.1: Compute the weight for VWHEF

1. Rank all filters (F_l) based on the final average accuracy of the validation set
2. Compute σ between the final accuracy of each filters (F_l)
3. $w_1 = \sigma$, If σ < 1 then $w_1 = \sigma+1$
4. $F_1 = \sum_{m=1}^{X^m} (w_1 . f_m)$
5. For i=2 to l
6. Compute diff Δacc = acc (F_{i-1}) - acc (F_i).
7. $w_i = w_{i-1} - \Delta acc$, if $w_i < 1$ then $w_i = 1$
8. $F_i = \sum_{j=1}^{X^j} (w_i . f_j)$
9. i=i+1
10. Go back to the loop

Procedure 7.1 is general for any number of filters, although our experiment uses five filters so we need to determine five weights. However, we give a weight equal to one for any filter that has a weight lower than one, which through experience we know often starts to happen after the second filter. The reason for giving a weight equal to one for any filter that has a weight lower than one is that some features were selected by all of the filters or some of these filters. Accordingly, if these features were selected by the highest accuracy filter or the second highest accuracy filter, this would mean that we were going to give them more weight. At the same time, if these features were selected by filters with lower accuracy, this would mean that we were going to give them less weight. As a result, the majority score of these features did not make it into the top ranking, because the lower weight of the lower accuracy filters affected their score and dragged them into the middle of the ranking.

7.3.3 Selective Filters Based on Validation Set (SFHEF)

When we assume that a filter is able to select more relevant and important features, this should lead to a highly accurate result; on the other hand, if a filter is unable to select relevant and important features, this should lead to less accurate results using the same classifier. This assumption motivates us to ignore the features selected by the worst performing filters and just to focus on the features selected by the best filters by aggregating their features.

In this section, as our experiment was carried out with an ensemble of five filters, we selected the top two filters only, based on their accuracy, to aggregate their results selected by their features and we disregarded the results of the three remaining filters, see Figure 7.4. In this case, SFHEF can be a special case of VWHEF as we can set $w_1 = w_2 = 1$ and $w_3 = w_4 = w_5 = 0$. Using this method, we still need to use a training set to rank the filters based on their accuracy, then we aggregate the features selected by the top two filters. Thus, we use the same framework as in Figure 7.2 but with a weight equal to 1 for the first two filters and a weight equal to zero for the remaining filters. The aims of using this method are to improve the feature selected results by SFHEF and to decrease the number of features aggregated by SFHEF, in addition to improving the accuracy and stability.

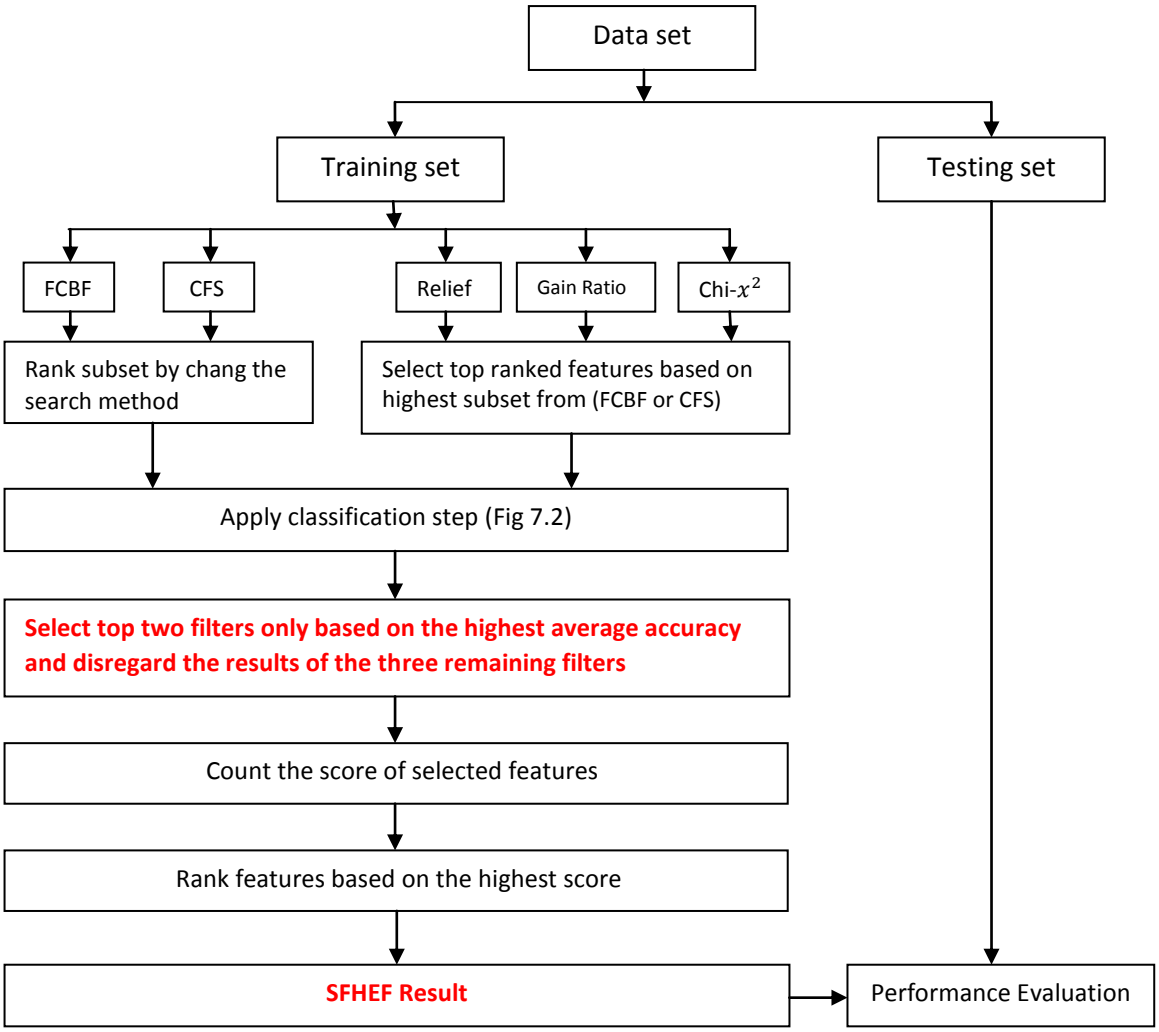


Figure 7.4: Framework of SFHEF

7.4 Experiments

7.4.1 Experimental Design Procedure and Evaluation methods

In this section we conducted several series of experiments with a variety of datasets to empirically evaluate the performances of three proposed ensemble techniques. We compared them with an improved version of HEF in Chapter 6, using 5 filters as members and the mean rank aggregation method with partial rank (HEF-b).

To verify the consistency in our experiments, we used the same datasets and stability measures as in previous chapters and the same classifiers: NB (John and Langley, 1995), KNN (Aha et al., 1991) and SVM (Platt, 1999). Also, we applied the same filters

as members in the ensemble: 2 SF (FCBF and CFS) and 3 RF (ReliefF, GR and Chi-x^2).

For each dataset, the experiments were carried out with the procedures as illustrated in Figures 7.1, 7.3 and 7.4, based on different techniques. In each fold, we firstly ran all FS methods (FCBF, CFS, ReliefF, Gain Ratio and Chi-x^2) by using 90% of all the instances (9 folds); after that the subsets produced by each FS were weighed based on each technique used (FWHEF, VWHEF and SFHEF) to generate the ensemble results and to produce a subset of ranked features. Then we used these rank subsets as input to the classifier with the same 90% of instances (9 folds). Following this, the accuracy of this subset was estimated over the unseen 10% of the data (1 fold). This was performed 10 times, each time proposing a possible different feature subset. In this way, we estimated accuracies and selected attribute numbers, which were the results of a mean over 10 'cross-validation samples'. Each experiment was then repeated 10 times with differently shuffled random seeds in order to assess the consistency of the results. In total, 51,000 models – 17 (5 FS + 12 ensemble) × 10 (datasets) × 3 (classifiers) × 10 (run) × 10 (folds) – were built for the experiments.

The statistical significance of the results of the multiple runs for each experiment was calculated, and the comparisons between accuracies were done with the Friedman test with a significance level of 0.05 (Demšar, 2006).

Moreover, in addition to accuracy, we will measure the stability of FS, as in each fold, the FS method may produce different feature subsets. Measuring stability requires a similarity measure for the FS results. The stability measures used in our investigation are: Relative Weighted Consistency (CWrel) and Average Tanimoto Index (ATI) (Somol and Novovicova, 2010), as the subset cardinality is not equal in our research. ATI evaluates pair-wise similarities between subsets in the system (10 folds), while CW_{rel} evaluates the overall occurrence of the features in the system (10 folds) as a whole. CW_{rel} and ATI may produce different results in each run, so the average of 10 runs will be used.

7.5 Results

In this section, the classification accuracy and stability results obtained after applying the different proposed ensembles were shown. To sum up, three ensemble approaches were tested: FWHEF, VWHEF and SFHEF. Also, we compared these three ensemble approaches with HEFb to demonstrate the capability of the proposed ensemble approaches to improve the results.

7.5.1 Accuracy Evaluation with Different Classifiers

Tables 7.1, 7.2 and 7.3 showed the accuracy of the results obtained with NB, KNN and SVM. Simple HEFb and three proposed ensembles were used over 10 datasets with all the features selected by 5 filters and the top 75% of the selected features. The remaining tables using top 50% and top 25% of the selected features were presented in the Appendix B.

Table 7.1 : The average test accuracy of NB classifiers trained with the features selected by HEFb, FWHEF, VWHEF and SFHEF, with 75% of these features being selected

Dataset	HEFb	HEFb-75%	FWHEF	FWHEF-75%	VWHEF	VWHEF-75%	SFHEF	SFHEF-75%
Zoo	94.46	**94.93**	94.46	93.15	94.46	**94.93**	94.47	93.44
Dermatology	97.79	98.14	97.79	**98.22**	97.79	98.20	98.20	98.20
Promoters	91.60	92.64	91.60	92.79	91.60	92.18	91.91	**93.79**
Splice	96.20	96.18	96.20	95.72	96.20	**96.21**	96.09	95.85
M-feat-fact	92.59	92.70	92.59	93.04	92.59	92.63	93.73	**93.90**
Arrhythmia	66.97	66.66	66.97	67.30	66.97	67.07	68.61	**68.23**
Colon	84.05	84. 24	84.05	84.69	84.05	85.43	85.43	**85.50**
SRBCT	**99.53**	99.04	**99.53**	99.03	**99.53**	98.93	99.07	98.68
Leukaemia	95.82	96.21	95.82	**96.35**	95.82	95.96	95.96	96.10
Ovarian	98.33	98.34	98.33	98.61	98.33	**99.61**	99.60	**99.61**
Average	91.734	91.908	91.734	91.89	91.734	92.115	92.307	**92.33**

Table 7.2: The average test accuracy of KNN classifiers trained with the features selected by HEFb, FWHEF, VWHEF and SFHEF, with 75% of these features being selected

Dataset	**HEFb**	**HEFb-75%**	**FWHEF**	**FWHEF-75%**	**VWHEF**	**VWHEF-75%**	**SFHEF**	**SFHEF-75%**
Zoo	**96.44**	95.93	**96.44**	95.43	**96.44**	95.93	95.45	95.63
Dermatology	95.8	96.14	95.80	**96.72**	95.8	96.58	96.6	95.54
Promoters	85.97	88.44	85.97	87.61	85.97	**88.62**	87.26	87.94
Splice	79.40	81.82	79.40	81.01	79.4	82.07	81.2	**83.02**
M-feat-fact	96.24	96.08	96.24	**96.55**	96.24	95.95	96.38	96.33
Arrhythmia	56.17	56.82	56.17	56.53	56.17	56.66	55.9	55.2
Colon	78.29	78.52	78.29	78.50	78.29	77.14	79.86	**80.41**
SRBCT	99.50	99.40	99.50	99.51	99.5	99.76	**99.89**	99.54
Leukaemia	94.75	**95.57**	94.75	95.16	94.75	95.03	94.28	94.3
Ovarian	99.52	99.48	99.52	99.48	99.52	99.48	99.65	**99.84**
Average	88.208	**88.82**	88.208	88.65	88.208	88.722	88.647	88.775

Table 7. 3: The average test accuracy of SVM classifiers trained with the features selected by HEFb, FWHEF, VWHEF and SFHEF, with 75% of these features being selected

Dataset	HEFb	HEFb-75%	FWHEF	FWHEF-75%	VWHEF	VWHEF-75%	SFHEF	SFHEF-75%
Zoo	95.74	96.83	95.74	95.74	95.74	**96.93**	95.55	95.93
Dermatology	97.29	**97.57**	97.29	97.54	97.29	**97.57**	97.54	97.29
Promoters	90.12	91.54	90.12	92.31	90.12	91.86	91.17	**92.4**
Splice	95.68	95.79	95.68	95.52	95.68	95.88	95.87	**95.96**
M-feat-fact	**97.76**	97.5	**97.76**	97.75	**97.76**	97.38	97.48	97.36
Arrhythmia	69.23	67.48	69.23	68.32	69.23	68.79	**69.52**	68.01
Colon	**87.29**	86.83	**87.29**	86.81	**87.29**	86.67	86.62	85.76
SRBCT	99.75	**99.89**	99.75	**99.89**	99.75	99.75	99.17	99.07
Leukaemia	96.39	**96.55**	96.39	96.41	96.39	96.41	**96.55**	96.26
Ovarian	**100.00**	**100.00**	**100.00**	**100.00**	**100.0**	**100.0**	**100.00**	**100.00**
Average	92.925	92.998	92.925	93.029	92.925	**93.124**	92.947	92.804

The value in bold letters points out the highest accuracy among other ensemble approaches. It should be noticed that the features selected by HEFb, FWHEF and VWHEF are the union of the features selected by each one of the filters, but with a different ranking. Therefore, we found that the accuracy of HEFb, FWHEF and VWHEF with all the features selected had the same accuracy because the same features

had been selected for them. On the other hand, SFHEF had a different accuracy because the features that were selected had been aggregated from only two filters with high accuracy, therefore the selected features were different. The reason for illustrating these three ensembles with full features was to compare them with SFHEF. After that, it was seen how each proposed ensemble produced different results with different rankings by removing 25% of the features from the bottom.

These results were not simple to analyse since the classifier plays an essential role and provides a very different classification accuracy, even with the same set of features. There were several cases found in the above tables that confirmed this fact, for example: HEFb, FWHEF and VWHEF over the Promoters dataset achieved an accuracy of 91.6% by NB, but this dropped to 85.97% with KNN, and with the same ensemble approaches but over the Splice dataset they increased their accuracy from 79.4% to 95.68% using KNN and SVM, respectively.

In general, the classification accuracy when selecting the top 75% of the selected features produced higher values than when selecting all the features in the three classifiers. As we can see, SFHEF-75% with NB shows 92.33% accuracy which was the highest among the other ensemble approaches, while HEFb-75% with KNN shows 88.82% accuracy, which was the highest among the other ensemble approaches. On the other hand, VWHEF-75% with SVM shows 93.124% accuracy which was the highest among the other ensemble approaches. The reason behind this improvement in the accuracy is that the irrelevant and redundant features in the bottom of the ranking were removed, due to obtaining low scores.

In detail, it was hard to determine which ensemble approach was producing the best improvement in terms of accuracy among these four approaches. It mainly depended on the datasets and the classifiers we used. As we can see, each approach had produced a few values in bold letters which meant it had the highest accuracy. However, the difference between the four ensemble approaches with all the classifiers is not significant.

Furthermore, it is worth mentioning that the results of the FWHEF approach depend on the results of SF. So, if SF succeeds in producing high accuracy, then the FWHEF approach will produce high results, but if it fails to produce high accuracy, then the FWHEF approach will produce lower results. Accordingly, as we mentioned in Section

7.3.3, we assumed that SF would often produce better results than RF based on some studies such as (Yu and Liu, 2004, Hall, 2000). Therefore, giving SF more weight should lead to improvement of the overall FWHEF ensemble. However, this is not a realistic assumption and the results produced by VWHEF and SFHEF based on the accuracy of the validation set are more general.

Table 7.4 lists the averages of test accuracy for each approach and classifier, independent of the datasets. We can see that the highest accuracy in the three classifiers was achieved by SFHEF. However, with 75% of the top selected features, the highest accuracies are different: SFHEF-75% has the highest accuracy with NB, and HEFb-75% has the highest accuracy with KNN, whereas VWHEF-75% has the highest accuracy with SVM. With the NB and SVM classifiers, the highest accuracy was achieved for VWHEF-50%, and with the KNN classifiers, the highest accuracy was achieved for FWHEF-50%. Finally, the highest accuracy was achieved for FWHEF-25% by using only the top 25% of the selected features.

Table 7.4: Average test accuracy over 10 real datasets focusing on the classifier

	NB	KNN	SVM
HEFb	91.734	88.208	92.925
FWHEF	91.734	88.208	92.925
VWHEF	91.734	88.208	92.925
SFHEF	**92.307**	**88.647**	**92.947**
HEFb-75%	91.908	**88.820**	92.998
FWHEF-75%	91.890	88.650	93.029
VWHEF-75%	92.115	88.722	**93.124**
SFHEF-75%	**92.330**	88.775	92.804
HEFb-50%	90.952	88.087	91.795
FWHEF-50%	91.561	**88.965**	92.268
VWHEF-50%	**91.707**	88.629	**92.289**
SFHEF-50%	91.414	88.717	91.751
HEFb-25%	87.966	87.189	88.744
FWHEF-25%	**88.474**	**87.653**	**88.904**
VWHEF-25%	88.305	87.205	88.581
SFHEF-25%	86.600	84.976	86.427

Figures 7.5, 7.6 and 7.7 show the average test accuracy of the NB, KNN and SVM classifiers respectively, using 10 datasets focusing on different methods. It is clearly seen that the classification accuracy from using the top 75% of the selected features produced the highest accuracy in the three classifiers, because the irrelevant and redundant features which could have lowered the score had been removed. In contrast, the classification accuracy from using only the top 25% of the selected features produced the lowest accuracy, because some relevant and important features which had median scores were removed and only the top 25% of the features were used. As a result, heuristically using the top 75% of the selected features was the best choice to select and concentrate on.

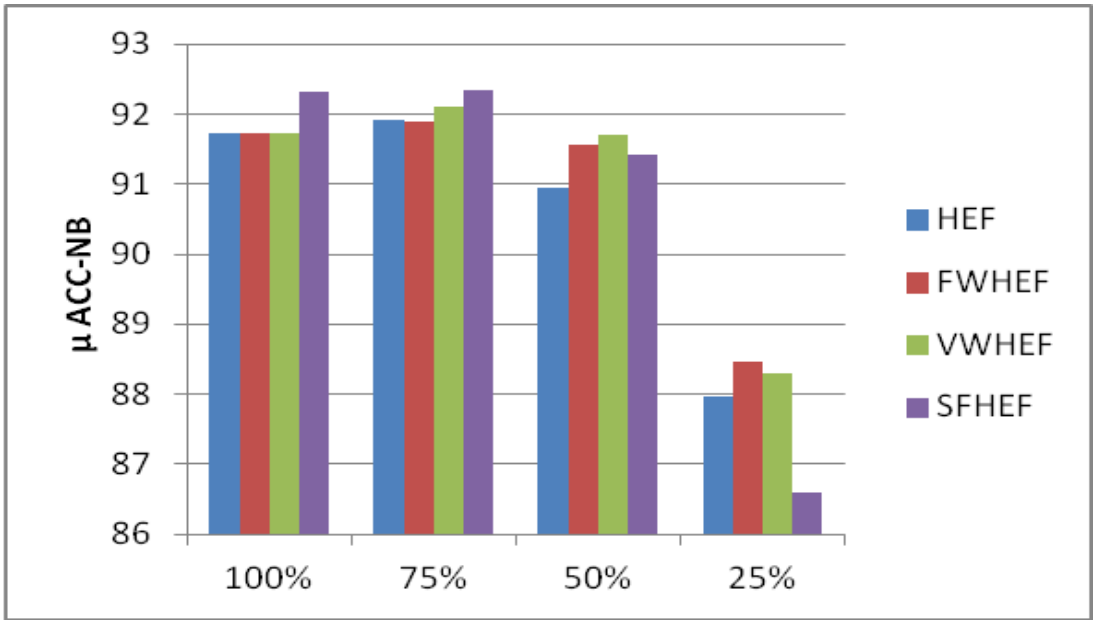


Figure 7.5: The average test accuracy of NB using 10 datasets focusing on different methods

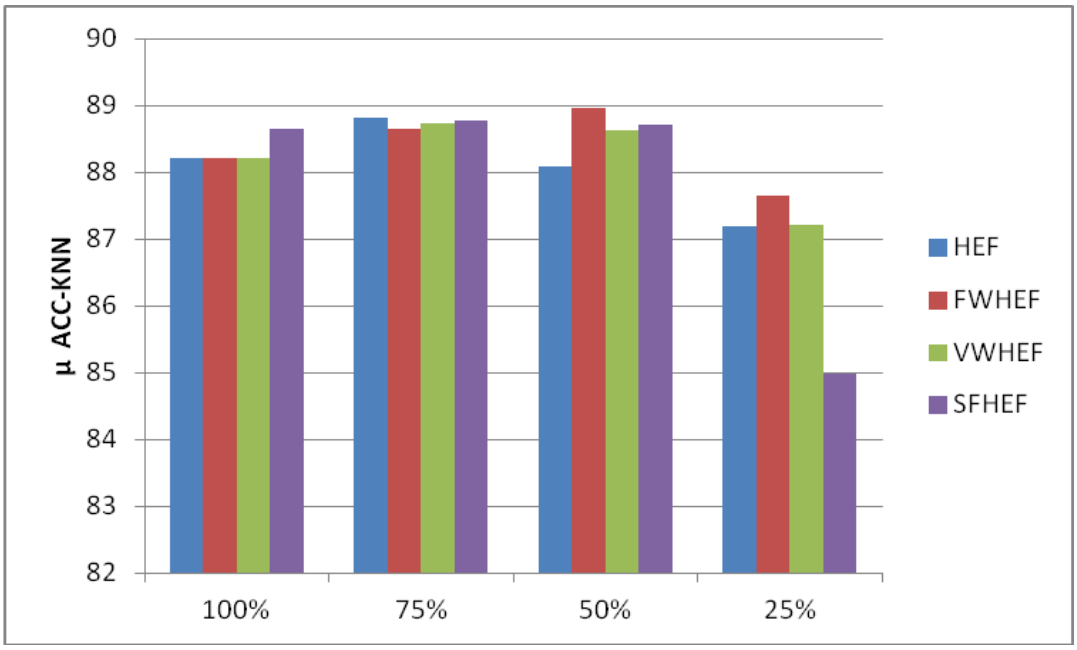


Figure 7.6: The average test accuracy of KNN using 10 datasets focusing on different methods

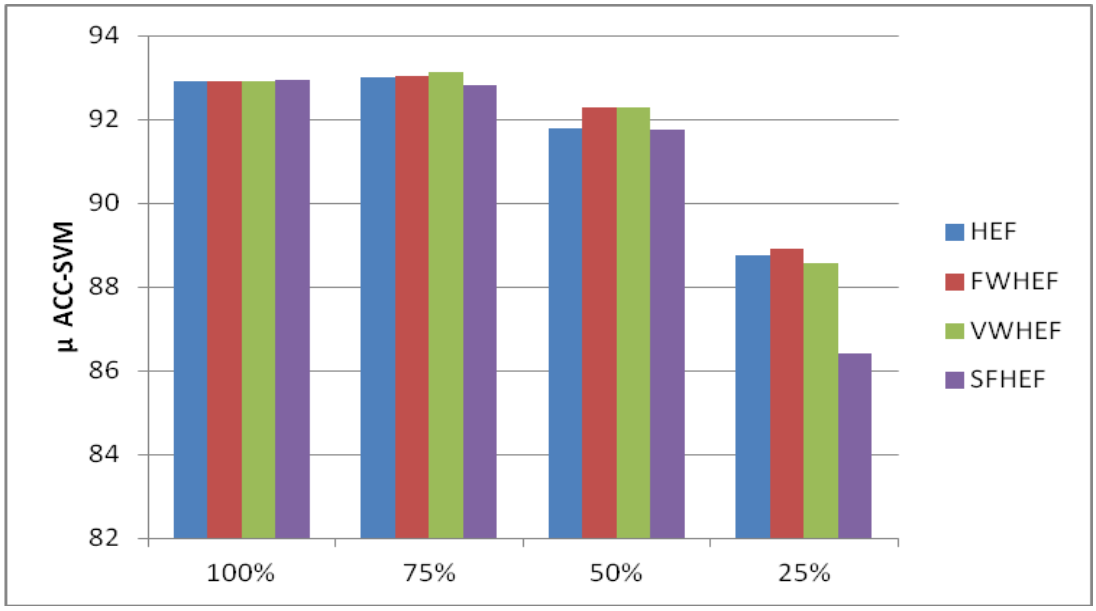


Figure 7.7: The average test accuracy of SVM using 10 datasets focusing on different methods

On the other hand, focusing on the ensemble approaches, SFHEF-75% had the highest accuracy by NB. In contrast, it was the lowest one when using only 25% of the selected features with all the classifiers. FWHEF-50% had the highest accuracy by KNN and VWHEF-75% had the highest accuracy by SVM. However, the ensemble approaches produced different accuracies when using different classifiers. So, no particular preferences were given to one over the others, which was proved statistically by the Nemenyi test. The difference between the four ensemble approaches with all the

classifiers was not significant, at p <0.05, except for SVM with 25% features selected, as we can see in Figure 7.8. We can identify two groups of ensemble approaches: the accuracy of SFHEF-25 is significantly worse than that of FWHEF-25, but we cannot tell which group VWHEF-25 and HEFb-25 belong to. The statistical statement would be that the experimental results are not sufficient to reach any conclusion regarding VWHEF-25 and HEFb-25 belonging to any groups.

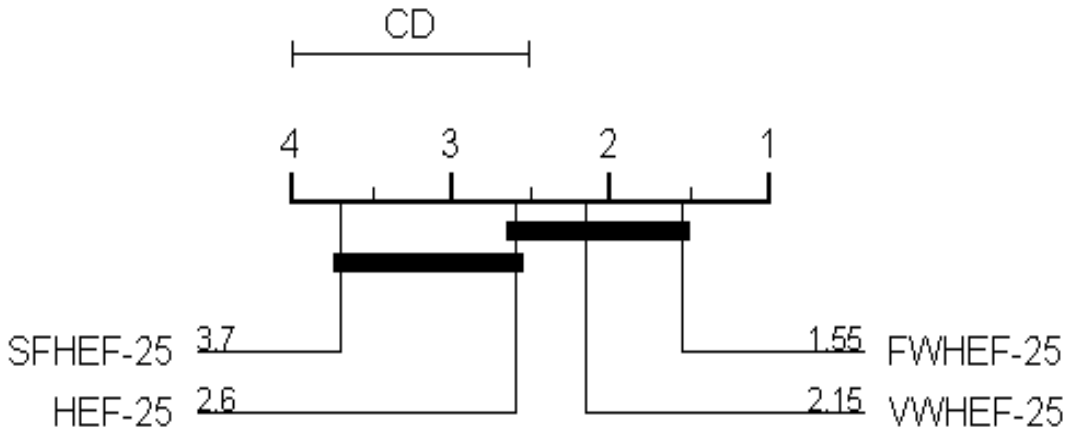


Figure 7.8: Comparison of all ensemble approaches against each other by SVM, using 25% of selected features with Nemenyi test

The next section analyses whether the proposed ensemble approaches were stable and to what extent they remained more stable than the simple HEFb.

7.5.2. Stability Evaluation

In practice, the high stability of feature selection is as equally important as high classification accuracy (Jurman et al., 2008). Numerous feature selection algorithms have been proposed; however, if we repeat the feature selection process by slightly changing the data, these algorithms do not inevitably identify the same candidate feature subsets (Yu et al., 2008). Therefore, many different subsets of features might be found from the method of the same feature selection or from different feature selection methods which can also achieve the same or similar predictive accuracy (Michiels et al., 2005). An unstable FS method is generally believed to have little value (Zhang et al., 2009). As a consequence, the confidence level in selecting optimal features would surely be reduced due to the instability of the feature selection results (Awada et al., 2012).

In this section, we discuss the stability of the three proposed ensemble approaches and compare them with the simple HEFb (in Chapter 6.4) without adding weight or using the training dataset.

Table 7.5: The stability measures of ATI with the features selected by four ensemble approaches over 10 runs of 10-fold cross-validation

ATI	HEFb	HEFb-75%	FWHEF	FWHEF-75%	VWHEF	VWHEF-75%	SFHEF	SFHEF-75%
Zoo	0.93	**0.97**	0.93	0.94	0.93	0.89	0.92	0.86
Dermatology	**0.94**	0.92	**0.94**	0.88	**0.94**	0.92	0.86	0.79
Promoters	0.71	0.78	0.71	0.75	0.71	0.78	0.74	**0.81**
Splice	0.80	0.88	0.80	0.76	0.80	**0.90**	0.86	0.88
M-feat-fact	**0.82**	0.78	**0.82**	0.72	**0.82**	0.78	0.70	0.64
Arrhythmia	**0.71**	0.67	**0.71**	0.68	**0.71**	0.68	0.53	0.48
Colon	**0.49**	**0.49**	**0.49**	0.47	**0.49**	**0.49**	0.46	0.48
SRBCT	**0.60**	0.576	**0.60**	0.57	**0.60**	0.57	0.47	0.44
Leukaemia	**0.47**	0.45	**0.47**	0.44	**0.47**	0.45	0.29	0.30
Ovarian	**0.55**	0.53	**0.55**	0.46	**0.55**	0.49	0.34	0.37
Average	0.702	**0.7046**	0.702	0.667	0.702	0.683	0.617	*0.605*

Table 7.6: The stability measures of CWrel with the features selected by four ensemble approaches over 10 runs of 10-fold cross-validation

CWrel	HEFb	HEFb-75%	FWHEF	FWHEF-75%	VWHEF	VWHEF-75%	SFHEF	SFHEF-75%
Zoo	0.94	**1.00**	0.94	0.94	0.94	0.88	0.90	0.84
Dermatology	0.85	0.91	0.85	0.84	0.85	**0.92**	0.83	0.79
Promoters	0.81	0.86	0.81	0.84	0.81	0.87	0.84	**0.89**
Splice	0.82	0.91	0.82	0.80	0.82	**0.93**	0.89	**0.93**
M-feat-fact	**0.84**	**0.84**	**0.84**	0.78	**0.84**	**0.84**	0.78	0.75
Arrhythmia	**0.81**	0.79	**0.81**	0.79	**0.81**	0.79	0.68	0.63
Colon	**0.65**	**0.65**	**0.65**	0.63	**0.65**	**0.65**	0.61	0.63
SRBCT	0.78	**0.79**	0.78	**0.79**	0.78	0.78	0.65	0.62
Leukaemia	**0.67**	0.61	**0.67**	0.59	**0.67**	0.61	0.44	0.46
Ovarian	**0.71**	0.69	**0.71**	0.63	**0.71**	0.66	0.49	0.53
Average	0.788	**0.805**	0.788	0.763	0.788	0.793	0.711	0.707

Table 7.5 shows how each ensemble approach with 75% of the selected features had a different stability within the same dataset; thus, it is apparent that some approaches were more stable than others when the samples had been changed. As we can see, HEFb-75% showed a higher average stability for all the datasets and VWHEF-75% was in second position, scoring 0.683. In contrast, SFHEF-75% was unstable in the face of changes in the samples, while HEFb, FWHEF and VWHEF scored in between because they have the same features with changes in the ranking order only.

The results in Table 7.6 show the details of the stability measures for CWrel. Similar patterns like those that appeared in Table 7.5 could again be observed. Again, HEFb-75% was irrefutably found to be more stable than other approaches, while HEFb, FWHEF and VWHEF produced values in the middle. The remaining tables show the stability evaluation with top 50% and top 25% of the selected features were represented in Appendix B.

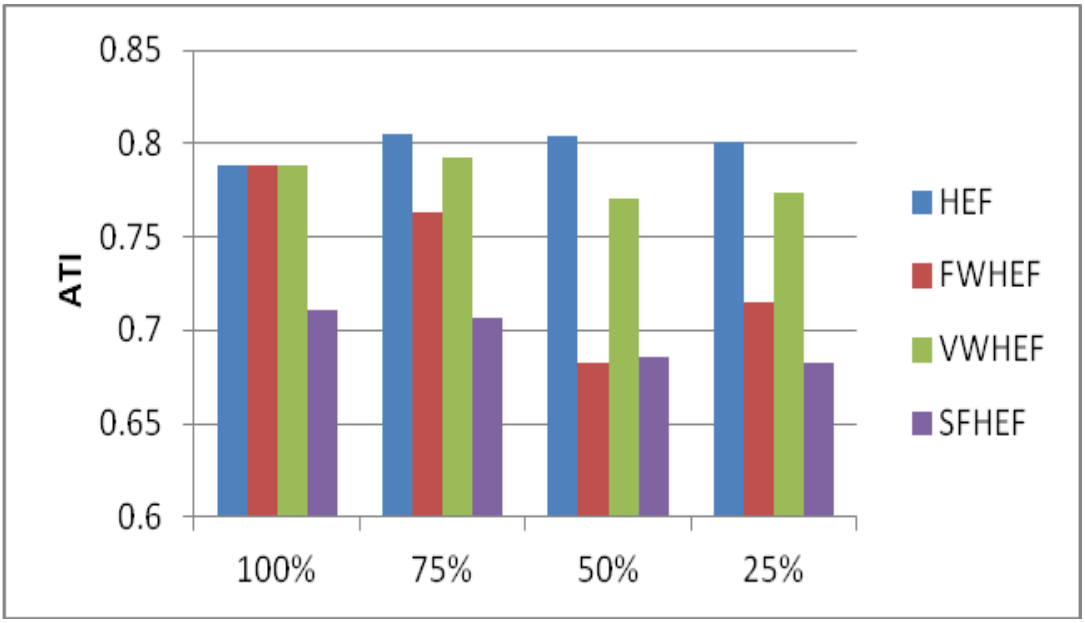


Figure 7.9: The average ATI using 10 datasets focusing on different methods

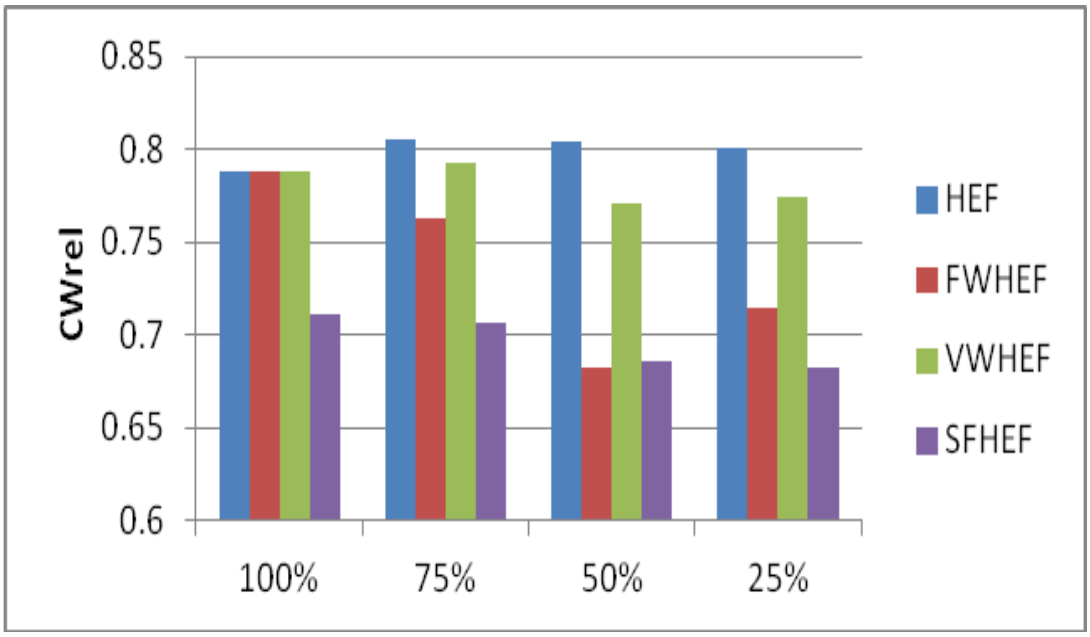


Figure 7.10: The average CWrel using 10 datasets focusing on different methods

Figures 7.9 and 7.10 show the average stability of ATI and CWrel respectively, using 10 datasets focusing on different methods. It is clearly seen that HEFb with all the selecting levels (100%, 75%, 50% and 25%) had the highest stability and outperformed the other proposed ensemble approaches. In contrast, SFHEF with all the cutting levels (100%, 75%, 50% and 25%) had the lowest stability.

The Nemenyi test showed that the accuracy of HEFb with selecting levels of 75% and 25% is significantly better than SFHEF with selecting levels of 75% and 25% (see Figure 7.11). We can identify two groups of ensemble approaches: the accuracy of HEFb with 75% and 25% is significantly better than that of SFHEF with 75% and 25%). We cannot tell which group VWHEF and FWHEF belong to. The results in CWrel showed similar patterns as those in Figure 7.11.

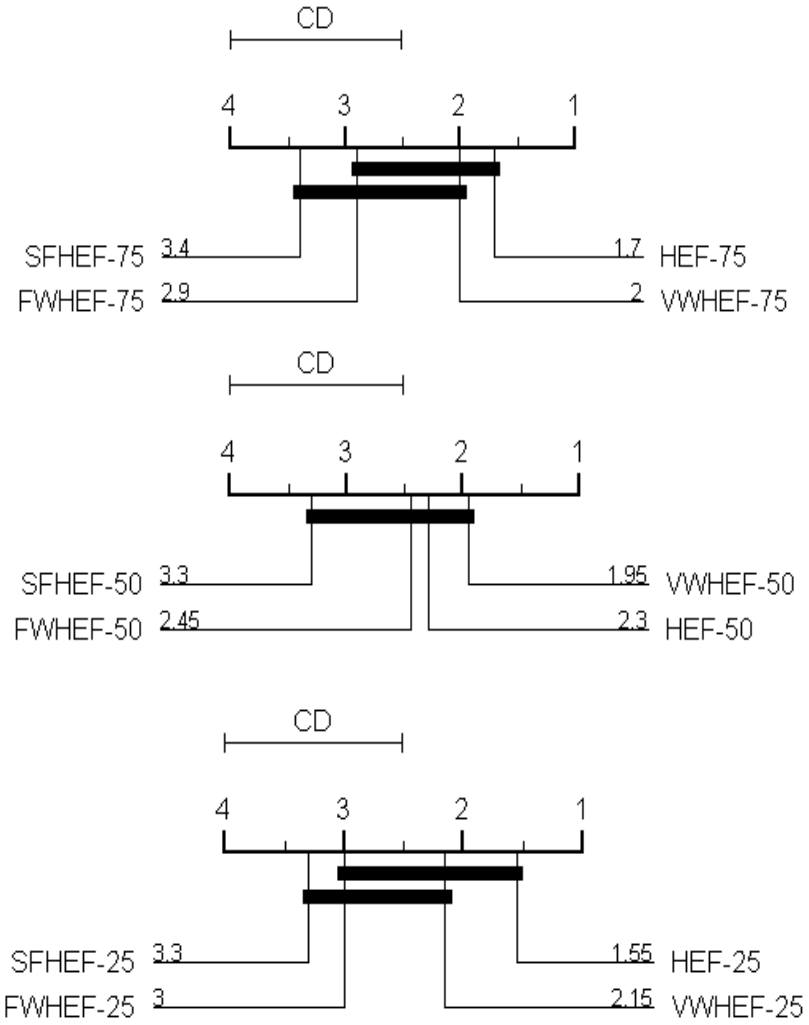


Figure 7.11: ATI comparison of all ensemble approaches against each other with Nemenyi test using 75%, 50% and 25% of selected features

In sum, we can conclude that the simple HEFb was more stable in dealing with changing samples than the other proposed ensemble approaches. In contrast, SFHEF was mostly unstable regarding changes in the samples, which proved that the HEFb method has a high level of stability, even if some of the members were relatively unstable.

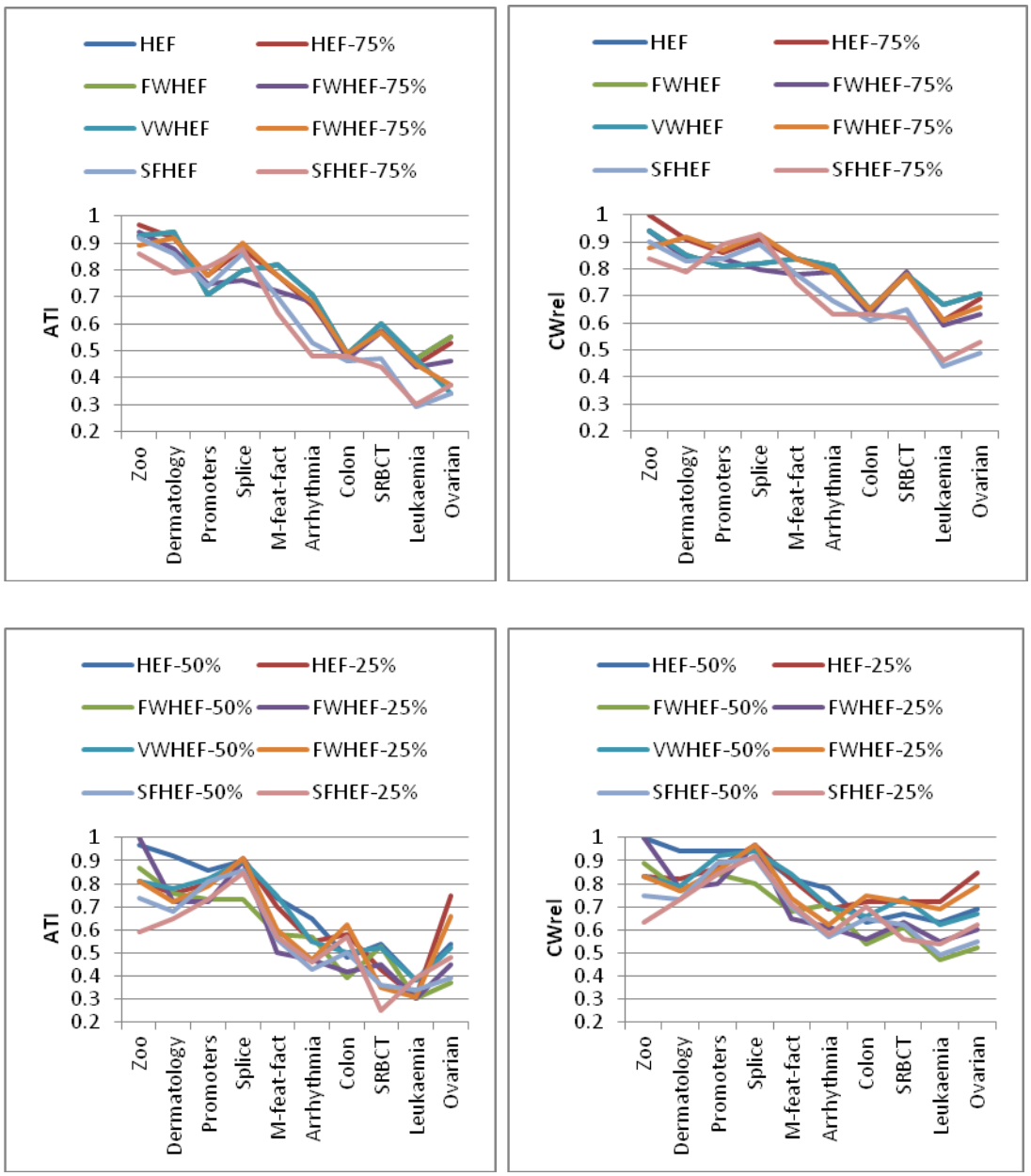


Figure 7.12: The mean stability measures of ATI and CWrel with the features selected by proposed ensemble approaches over 10 runs of 10-fold cross-validation

Furthermore, 7.12 showed that HEFb and other proposed ensemble approaches had been more stable in some datasets than in others, based on certain factors such as number of samples, number of features and number of class labels. It could be seen that ensemble approaches on microarray datasets were less stable than on other dataset types, as the number of features tended to be high and the number of samples very low.

7.5.3. Runtime Performance

In this section, we measured the execution time needed to run the ensemble approach and then to build the classifiers. It is important to compare the computational performance of each ensemble approach, since the model building phase using the validation set is computationally time-consuming, as used in VWHEF and SFHEF in order to determine the weight of each filter.

Tables 7.7 to 7.9 record the running time for each ensemble approach using NB, KNN and SVM, respectively. This test was repeated 10 times to give the average execution time required to run each ensemble approach and to build the classifier.

We can observe that HEFb and FWHEF are consistently faster than VWHEF and SFHEH. The time savings from HEFb and FWHEF become more obvious when the data dimensionality increases. In many cases the time saving are in degrees of magnitude. These results verify the superior computational efficiency of HEFb and FWHEF over VWHEF and SFHEF, since with HEFb and FWHEF there is no need to run the classifier using the validation dataset in order to determine the weight of each filter.

Table 7.7: Running time (seconds) for each ensemble approach with NB classifier on 10 real datasets

Runtime performance-NB	HEFb	FWHEF	VWHEF	SFHEF
Zoo	0.03	0.03	0.15	0.15
Dermatology	0.08	0.08	2.72	2.72
Promoters	0.03	0.03	0.23	0.23
Splice	0.26	0.26	56.38	62.06
M-feat-fact	1.4	1.4	115.32	113.07
Arrhythmia	0.45	0.45	9.74	9.78
Colon	9.02	9.02	14.34	14.37
SRBCT	12.45	12.45	23.54	23.78
Leukaemia	100.42	100.42	150.24	126.2
Ovarian	532.35	532.35	770.2	773.1
Average	**65.649**	**65.649**	114.286	112.546
Average without Ovarian	**13.79**	**13.79**	41.40667	39.15111

Table 7.8: Running time (seconds) for each ensemble approach with KNN classifier on 10 real datasets

Runtime performance KNN-	**HEFb**	**FWHEF**	**VWHEF**	**SFHEF**
Zoo	0.03	0.03	0.15	2.03
Dermatology	0.08	0.08	2.72	2.72
Promoters	0.03	0.03	0.23	0.39
Splice	2.78	2.78	65.44	61.6
M-feat-fact	2.13	2.13	116.15	120.74
Arrhythmia	0.44	0.44	9.65	16.01
Colon	9.23	9.23	14.28	14.71
SRBCT	13.17	13.17	23.52	24.85
Leukaemia	120.44	120.44	147.97	149.96
Ovarian	459.34	459.34	298.447	298.447
Average	**60.76**	**60.76**	67.85	69.14
Average without Ovarian	**16.48**	**16.48**	42.234	43.667

Table 7.9: Running time (seconds) for each ensemble approach with SVM classifier on 10 real datasets

Runtime performance-SVM	**HEFb**	**FWHEF**	**VWHEF**	**SFHEF**
Zoo	0.15	0.15	1.83	1.91
Dermatology	0.18	0.18	2.55	2.63
Promoters	0.04	0.04	0.38	0.39
Splice	5.32	5.32	215.87	241.49
M-feat-fact	1.76	1.76	134.39	121.04
Arrhythmia	0.93	0.93	16.36	16.2
Colon	9.74	9.74	14.72	14.61
SRBCT	12.98	12.98	24.65	24.29
Leukaemia	121.99	121.99	151.61	147.13
Ovarian	551.39	551.39	791.46	709.94
Average	**70.448**	**70.448**	135.382	127.963
Average without Ovarian	**17.01**	**17.01**	62.48	63.29

Figure 7.13 shows the average runtime performance of 9 real datasets with each ensemble approach using three classifiers. HEFb shows a significant reduction in computation time in comparison with VWHEF and SFHEF.

Figure 7.13: Average runtime performance of 9 real-world datasets (excluding Ovarian) using three classifiers

7.6. Discussion and Evaluation

In this chapter, we proposed a framework of a weighted heuristic ensemble of filters (WHEF), and examined the performance of three special cases. Our framework is mainly designed for an ensemble of filters and it is flexible as it can use (a) any type of filters as a member in the ensemble, (b) any aggregation methods, and (c) full or partial ranking of features from each filters. The three special cases considered are: fixed weight, variable weight and selective filters. The first case is FWHEF, which adds a fixed additional weight to SF and a fixed lesser weight to RF in order to allow SF to play more important roles in generating the consensus feature ranks. The second case is VWHEF, which adds a variable weight on some filters based on the classification accuracy. The third method is SFHEF, which selects the top two filters only, based on their accuracy, to aggregate their results based on selected features, disregarding the results of the three remaining filters. Then, we compared them with the simple HEFb, which aggregates the features using mean ranking order, without weighting the filter members.

The contributions of this chapter include: 1) employing the supervised learning approach for ensemble filters; 2) using a validation set by taking an average of 10 folds to identify which filters were better, in order to add more weight to them; 3) developing an algorithm to calculate the weight from a validation set based learning method; and 4) empirical verification of the effectiveness of the proposed approaches.

The experimental results showed that the simple HEFb at all selection levels performed with more stability and consumed less time for all the cases, while the accuracy was not significantly different to the three proposed ensembles, which mean HEFb more reliable than three proposed weighted ensembles.

Specifically,

(1) No single best approach for all the situations could be found, in term of accuracy. In other words, the accuracy performance of each approach varied from dataset to dataset and was also influenced by the type of classifiers chosen for the models. Thus, one approach might perform well in a given dataset for a particular classifier but would perform poorly when used on a different dataset or with a different type of classifier.

(2) Averaging over 10 datasets, SFHEF and SFHEF-75% showed the highest accuracy by NB and KNN and a little less by SVM. On the other hand, they showed the lowest value when using only 25% of the selected features. The remaining ensemble approaches showed different average accuracies by using different classifiers; no particular preferences should be given to one over the others, which was proved statistically by the Nemenyi test.

(3) HEFb showed the highest stability for ATI and CWrel. This result demonstrated that the simple ensemble HEFb that had been proposed by us (in Chapter 6) was more reliable and consistent than the three ensembles which were proposed later.

(4) Among the four categories of the feature selection, selecting 75% of the top ranked features was the best choice compared with other selection categories in terms of accuracy and stability.

7.7. Conclusion

The experimental results indicate that adding weight to filters in an HEFb has not achieved the expected improvement in accuracy, while it increases time and space complexity, and clearly decreases stability.

Our hypothesis that adding more weight to ‘good filters’ should lead to better results is not true based on our experimental results. This is because it is formulated purely based

on intuitive consideration of an ideal world. In practice, the assumption of 'good filters' does not always hold and is often untrue, because good filters that are found to be good on the training and/or validation datasets may not (and often are not) good on the testing dataset.

From the significance test, we can conclude that there is a significant difference between HEFb and the three proposed weighted ensemble methods in stability. However, there is not a significant difference between HEFb and three proposed weighted ensemble methods in accuracy. Therefore, the simple HEFb is better on balance.

Chapter 8

Evaluation and Discussion

8.1 Introduction

This chapter evaluates and discusses the work and the results of this thesis. In addition, it presents a comparison between the proposed HEFs and then selects the best one in terms of accuracy and stability.

In general, our core algorithm is a heuristic ensemble of filters (HEF), and each version of the HEF used in the previous chapters was encoded with a logical scheme, as summarised in Table 8.1 and Figure 8.1

Table 8.1: List of abbreviations for each version of HEF

Abbreviations	Represent
HEF	Heuristic Ensemble of Filters.
HHEF	Hybrid HEF, add a wrapper after HEF.
WHEF	Weighted HEF, add different weights to FS members in the HEF.
HEF-a/b/c	Three different schemes of mean rank aggregation with a partial list. If no letter is added after 'HEF', it indicates that counting the most frequently selected features was used as the aggregation function.
1. HEFa	Ranks the features based on frequency. If some features had equal frequencies, we ranked them by means of these features, and we made sure that each feature did not appear in the list; the position was equal to K+1, where *k* is the maximum number of features in the partial list.
2. HEFb	Ranks the features based on the mean, and we made sure that each feature did not appear in the list; the position was equal to K+1.
3. HEFc	Similar to previous methods (HEFb), but we divided the mean of each feature by the frequency of this feature.
HEF-R1	Heuristic ensemble of filters after removing any features selected by only one filter.
HEF-#%	Selects the top-ranking feature based on the percentage (75%, 50% or 25%).
HEF-4F	Uses four filters as FS members in the ensemble.
HEF-5F	Uses five filters as FS members in the ensemble.

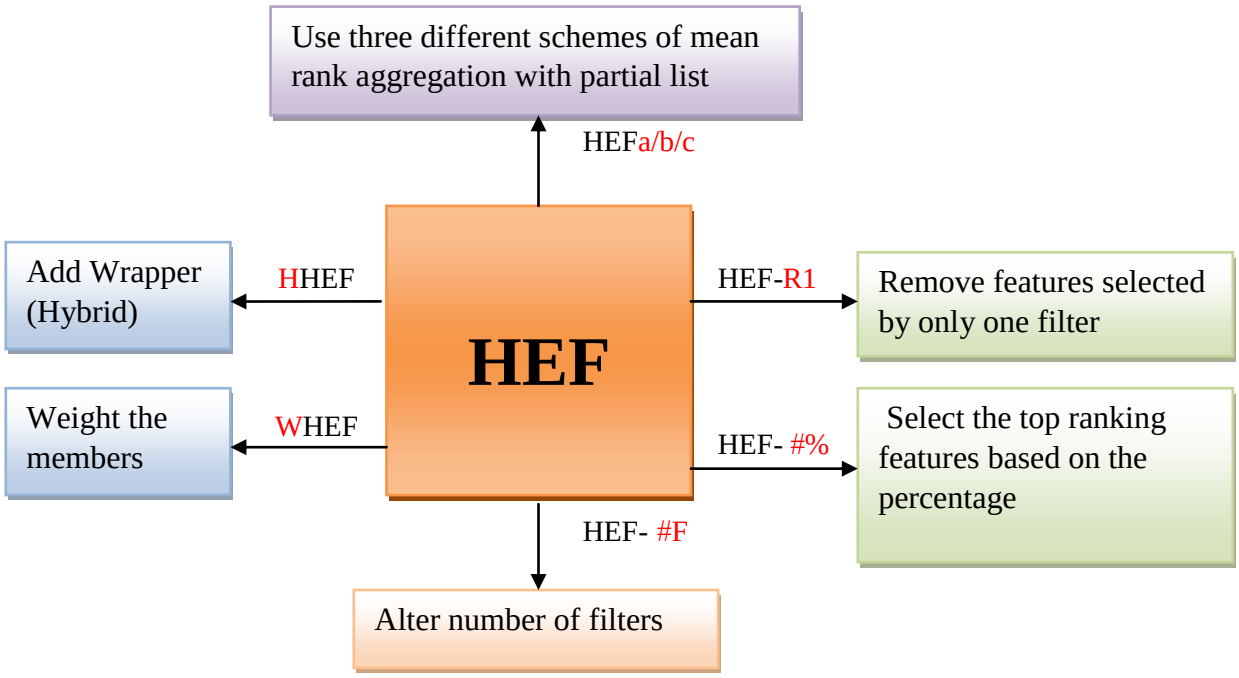


Figure 8.1: Naming strategy of each version of HEF

This chapter is organised as follows: in Section 8.2, we will give an overview of the research as a whole, then in Section 8.3, we will evaluate the HEF and the changes made to it in the previous chapters, with the aim of improving its accuracy and stability. In Section 8.4, we will evaluate the use of data in FS by applying a large number of datasets. In Section 8.5, we will discuss the aggregation methods we used in this thesis. In Section 8.6, we will evaluate how we can weight each member differently based on their accuracy. In Section 8.7, we compare the best HEF version with the results that were published by others in previous studies. Finally, in Section 8.7, we will summarise the research as a whole.

8.2 Overview of the Research as a Whole

First, our HEF was tested using 10 benchmark datasets and three types of classifiers – NB, KNN and SVM – to verify the consistency of the methods used to select the features. In the statistical tests performed in Chapters 4 and 5, we used Student's paired two-tailed t-test with a significance level of 0.05 in order to test the results of the classification of the models, which had been trained using the features selected by the current selector. This test was performed to determine whether these models were

significantly better or worse than the models trained with all the original features. We used this test because we needed to compare two results (paired). However, in Chapters 6 and 7, we used the non-parametric Friedman test with a significance level of 0.05 (Demšar, 2006). It ranked the algorithms for each dataset independently. The best performing algorithm was ranked 1, the second best was ranked 2 and so on. In the case of ties, average ranks were assigned. If the null hypothesis was rejected, the Nemenyi test was then conducted. We used this test because we needed to compare the accuracy of multiple algorithms applied to multiple testing datasets. The accuracy of two algorithms was considered significantly different if the corresponding average ranks differed by at least the critical difference.

In Chapter 4, we introduced the HEF algorithm, which is composed of two types of filters (SF and RF), and we counted the frequency of the selected features as its consensus function. We were motivated to design the HEF because an ensemble of FS had already been shown to be superior to a single FS in terms of reliability and in some cases in terms of accuracy, especially in difficult and challenging datasets. However, the review of previous research in the area of EFS found that the majority of these studies were predominantly limited to using one filter with instance level perturbation (e.g., boosting) or combining different numbers of RFs as the components of an ensemble, which produced a ranking of features. Moreover, previous studies did not use an ensemble of SF, nor did they combine SF and RF. The idea of combining SF and RF in HEF exploits the advantages of each, as will be discussed later. Moreover, additional work was done to determine the cut-off point required to produce a subset of selected features.

Initially, in Chapter 4, we used entire datasets in the experiments, a practice commonly used in FS, and then selected the features as inputs for the classifier (i.e., the ALL method). However, this convention has been questioned in recent years, primarily because it may cause over training. Therefore, in Chapter 5, we investigated the use of data in FS by comparing the ALL method with the PART method, which performs FS inside the cross-validation loop by executing the FS method on the training set before constructing the classifier in each iteration. We then concluded that the PART approach could prevent bias to some extent, although its superiority decreased as the sample sizes increased. Hence, we used the PART approach in the remaining chapters. Further discussion about this study will be presented in Section 8.4. After deciding which approach to use, we went back in Chapter 6 and focused on the main aim of this

research, which is to develop the HEF that can improve the reliability by measuring the stability in conjunction with improving the performance by measuring the classification accuracy. In Chapter 6, we attempted to improve the HEF through three procedures. Firstly, we extended the HEF by applying different wrappers after the results obtained by HEF, with the aim of reducing the number of features selected while preserving the same accuracy and stability. Secondly, we added more filters as components in the HEF. Thirdly, we changed the aggregation method from counting the frequency of each feature selected to mean rank aggregation, by sorting the selected features based on the means of their ranks in all the ranking filters. In addition, we discussed the partial rank and the ways to deal with this situation. On the other hand, in order to improve the performance of HEF, we investigated the ideas of weighting each filter member differently. Intuitively speaking, it is reasonable that the filters should be treated differently in accordance with their performance, as in reality there are some differences in the performance of filters. Thus, the use of different weights to calculate the total scores of the selected features may improve the performance. Therefore, in Chapter 7, we investigated ways to determine the appropriate weight for each filter in an ensemble, with the aim of further improving the HEF.

8.3 Heuristic Ensemble of Filters (HEF)

In Chapter 4, in our initial experiments, which use HEF with the ALL method, the experimental results showed that the HEF performed better overall in terms of consistency and accuracy than using a dataset without FS or an individual filter. Specifically, HEF-R1 performed the best for NB and KNN, whereas HEF performed the best when the SVM classifier was used, which demonstrates that the proposed ensemble was more accurate and consistent than the single filters. However, there is no single best approach for all situations. This was expected because the accuracy of the single filter varied from dataset to dataset, and was influenced by the type of model chosen as the classifier. Thus, one filter may perform well on a given dataset for a particular classifier but perform poorly when used in a different dataset or with a different type of classifier. Although the HEF has the all features selected from the four filters, it is still much less than the full feature set by up to 50 times for genetic datasets. When two types of filters were combined in the ensemble, we found that the accuracy of SF (FCBF and CSF) was frequently better and less frequently worse on average than the accuracy of the RF.

There are some key characteristics that make our HEF a good choice:

(1) We combined SF with RF in our ensemble algorithm to exploit the advantages of each, whereas the majority of the previous research on EFS focused on ranking filters only.

(2) We applied heuristic cut-off rule which used the highest number of features in the SF as a cut-off point for the top-ranking features of the remaining ranking filters, which accelerated the ensemble algorithm. Therefore, we did not need to select various feature numbers to test the accuracy of the rankers (as other researchers have done) or to use a wrapper to choose the appropriate number of features.

(3) We applied heuristic consensus rules to remove the selected features that had low frequency. Because the combination method used counts the most frequently selected features, it is therefore possible that a high number of features are selected by the ensemble filters.

(4) We designed the HEF to use any number of FS members and any type of aggregation method. In addition, we could use full or partial ranking of the features of each filter.

(5) We designed the HEF to accept different characteristics of datasets from different domains, which was not the case in most EFS studies. Any type of classification data could be used with HEF, such as binary, multivariate, nominal, numerical, a high number of samples and a high number of features. Because we chose filter members that could manipulate all these issues within a reasonable time.

However, this initial study requires further investigation, such as adding a wrapper after the HEF. In addition, further research needs to determine the types and number of filters that should be included in the proposed ensemble.

For the above reasons, in Chapter 6, we applied the wrapper after HEF (HHEF) to make the wrapper capable of focusing on the remaining relevant features; after that most of the irrelevant features were removed by HEF. The aim was to identify the most important features while preserving the same accuracy and stability. We chose three wrappers that were considered fast and were popular in the literature: greedy forward search, linear forward selection and re-ranking search. In the experiment, wrappers were selected to work incrementally at the feature level, such as greedy forward search, and

also to work incrementally at the block or set of features level, such as linear forward selection and re-ranking search. However, the results of applying the three types of wrappers after the HEF led to the selection of very few features. However, accuracy and stability were clearly reduced. The three types of HHEF had a lower average accuracy, especially in the microarray datasets, which indicates that although HHEF and HHEF-R1 might help to identify the most important features, they leave out some less important features, which leads to a decrease in the accuracy of the classification. The experimental results demonstrated that the HEF was more reliable, consistent and effective than HHEF because the features selected by the HEF achieved better accuracy and stability results. Furthermore, HHEF helped to reduce the number of selected features by as many as three times, especially in microarray datasets. Thus, it revealed the most important features but left out less important features, which led to sacrificing some overall accuracy and stability of the classification. Therefore, based on this result, we did not continue to work on HHEF but instead extended the investigation by adding more filters as members with the aim of further improving the HEF.

In the same chapter (Chapter 6) we discussed the types and number of filters that should be included in the proposed ensemble in order to improve the reliability of HEF's feature selection. In Chapter 4, we categorised these evaluation criteria into groups broadly based on the following studies (Saeys et al., 2007, Liu and Yu, 2005): distance, information, dependency and consistency. We then studied the popular filters in each category in order to choose the appropriate filters from each category. Then we chose two SFs (CFS and FCBF) and two RFs (ReliefF and GR). After that in Chapter 6, in order to further improve the ability of HEF to select more reliable and stable features, we categorised these evaluation criteria into groups broadly based on the following: distance, information, dependency, statistics and consistency (Fahad et al., 2014). Then we chose Chi-X^2 and added it to our ensemble because it is based on statistical measures that were not considered in the earlier experiments (Chapter 4). It should be noted that each filter algorithm in our HEF used a different criterion to evaluate the relevance of the candidate features in the datasets. When combined, many different aspects of the candidate features were assessed.

In terms of determining the number of member filters, we followed the guidelines given in (Wang et al., 2010b, Wang et al., 2012), in which the ensemble of a very few carefully selected filters is similar to or better than the ensembles of many filters.

Two key findings emerged regarding the number and type of filter member:

(1) The results showed that HEF-5F and HEF-R1-5F achieved the best accuracy result but never the worst result. Therefore, they improved in terms of classification accuracy. In addition, in terms of stability, HEF-5F and HEF-R1-5F showed greater increases in the stability than HEF-4F and HEF-R1-4F. Chi-x^2 showed a higher average stability for all datasets, which indicates that adding more stable members increases the stability of the ensemble.

(2) Among the filter members used in our heuristic ensemble of filters, RF (ReliefF, GR and Chi-X^2) were more stable than SF. In particular, Chi-X^2 showed higher average stability in all the datasets.

(3) Different numbers and types of ensemble members led to the selection of different features, which led to different levels of classification accuracy and stability.

In summary, the addition of the Chi-X^2 filter to HEF (HEF-5F) improved stability and slightly improved accuracy which led to increasing the reliability, whereas the addition of the wrapper after HEF (HHEF) reduced accuracy and stability because it left out some less important features, which led to decreasing the accuracy of the classification. Thus, we continued this research by using HEF with five filters (including Chi-X^2). We discarded the idea of adding a wrapper after HEF. In Chapter 6, the next stage of improving the HEF involved changing the aggregation method, which will be evaluated in Section 8.5.

8.4 Use of Data in FS

In Chapter 5, we determined appropriate approaches for using data in feature selection. It is important to investigate this issue, since it is a general and important issue in FS, and because no clear answer has been obtained by the existing studies, especially when filters are used. Consequently, we investigated this issue in Chapter 5 before conducting the remaining research.

In order to answer this question, we first described the characteristics of the PART method (which performs FS inside the cross-validation loop by executing the FS method on the training set before constructing the classifier in each iteration) and the ALL method (which used entire datasets and then used the selected features as an input

for the classifier) which were used in Chapter 4. Secondly, we generated 21 synthetic datasets with different numbers of features, samples and levels of noise. Thirdly, we used suitable similarity and stability measures to evaluate the stability of each method and to evaluate the ability of each method to identify relevant features, in addition to the traditional way of evaluating FS by using a classifier. Finally, we compared the results of the ALL method and PART method in 10 real-world benchmark datasets, 21 generated synthetic datasets and 4 synthetic benchmark datasets.

The experimental results of this investigation showed the following: when the dataset contained a large number of samples, there was no noticeable difference between these two approaches in terms of stability and accuracy. When the dataset was small, the stability of the ALL and PART methods was almost similar. However, there was a clear difference in terms of their accuracy; that is, the ALL approach achieved a higher accuracy than the PART approach, which indicates that the accuracy estimate was possibly overstated and that bias occurred. Therefore, the PART approach could prevent bias to some extent although its superiority decreased as the sample sizes increased. Hence, we used the PART approach in the remaining chapters of this research.

8.5 Aggregation Method

There is another issue that has been investigated in this research, which is aggregation. It is a key component in the feature selection ensemble at it combines the different outputs from different FS methods into a single result and thus directly influences the performance of an ensemble. Hence, a suitable aggregation method must be chosen.

In Chapter 4, we focused on ensemble feature selection techniques that work by aggregating the feature subsets provided by the different filters in a final consensus subset. Counting the most frequently selected features was used as the consensus function (or aggregation method). The most frequently selected features were placed at the top, and the least frequently selected features were placed at the bottom (outer ranking). However, because the probability of any two features having the same frequency is high, and to resolve the issue of frequency collision (and to take advantage of RF by knowing the most important features), we introduced a mean ordering strategy derived from RF. The score of each feature was determined by the average ranking score in all the ranking lists. The sorting was performed in increasing order (inner

ranking). One issue in integrating multiple scores is that different filtering algorithms often provide evaluation scores with different scales. In order to combine the evaluation results of multiple filters, the evaluation scores must be transformed into a common scale. Therefore, the softmax scaling (Yang et al., 2010) process was adopted to transform the feature evaluation results of each filtering algorithm into the range of [0-1].

However, aggregating the outputs by counting the most frequently selected features may produce a high number of selected features, including the low frequency levels selected by only two filters or even a single filter. In order to address this issue and to obtain further important features, a heuristic consensus rule was applied to produce the final output of the HEF. The first heuristic ensemble of filters, named HEF-R0, has all the features selected by all members, whereas HEF-R1 is the heuristic ensemble of filters after the removal of any features selected by only one filter. This experiment used HEF-R0, or simply HEF, and HEF-R1.

In Chapter 6, in attempting to improve HEF and to apply the idea of weighted ensemble filters described in Chapter 7, we changed the aggregation method from simply counting the frequency of each feature selected to mean rank aggregation by sorting the selected features based on the means of their ranks in all the ranking filters.

However, two issues had to be resolved before the aggregation method could be changed. In the first issue, SF produced subset features without ranking these features, which forced us to use the frequency and limit our options of using other rank aggregation methods. Thus, we converted the subset filters (FCBF and CFS) to ranked subset filters with suitable ranking evaluation criteria. In the second issue, each filter member produced subset features even for RF because we had selected top features based on the highest subset from SF. To solve this issue, we considered the partial rank and dealt with this situation by proposing and investigating three schemes of mean rank aggregation with a partial list.

After solving the two issues by ranking the SF and dealing with the partial list, we were able to use other techniques to aggregate the rank features. Therefore, we decided to use mean aggregation, which is the most commonly used rank list and aggregation technique. This choice was justified in Section 6.4.3.

Three different schemes of mean rank aggregation with a partial list were compared. The first scheme ranked the features based on frequency. If some features had equal

frequencies, we ranked them by means of these features, and we made sure that each feature did not appear in the list; the position was equal to K+1, where K is the maximum number of features in the partial list. We represented this scheme as HEFa-5F. The second scheme ranked the features based on the mean, and we made sure that each feature did not appear in the list; the position was equal to K+1. We represented this scheme as HEFb-5F. The third scheme ranked the features based on the mean, and we made sure that each feature did not appear in the list; the position was equal to K+1. Then we divided the mean of each feature by the frequency of this feature, and we represented this scheme as (HEFc-5F).

The results of the comparison of the three schemes of mean rank aggregation, which dealt with the top-K list, confirmed that ranking the feature based on the mean and making sure that each feature did not appear in the list with a position equal to K+1 (HEFb-5F) was the best scheme in terms of accuracy and stability, in most cases.

In summary, as described in Section 8.3 the HEF-5F was better than other HEF method in most cases, especially HEF-R1-5F. On the other hand, as described in Section 6.4, the HEFb-5F was better than other HEF method in most cases especially HEFb-75%-5F. Now we will compare HEFb-5F and HEF-5F. Both have the same number and type of filter members but different aggregation methods. Because the best result of HEFb-5F was achieved by HEFb-75%-5F and the best result of HEF-5F was obtained from HEF-R1-5F, we will limit the comparison to these two methods.

Table 8.2: Average test accuracy and stability over 10 real benchmark datasets with two different aggregation methods

HEF	NB	KNN	SVM	ATI	CWrel
HEFb-75%-5F	91.91	88.82	**92.998**	**0.7046**	**0.815**
HEF-R1-5F	**91.92**	**89.172**	92.35	0.66	0.767

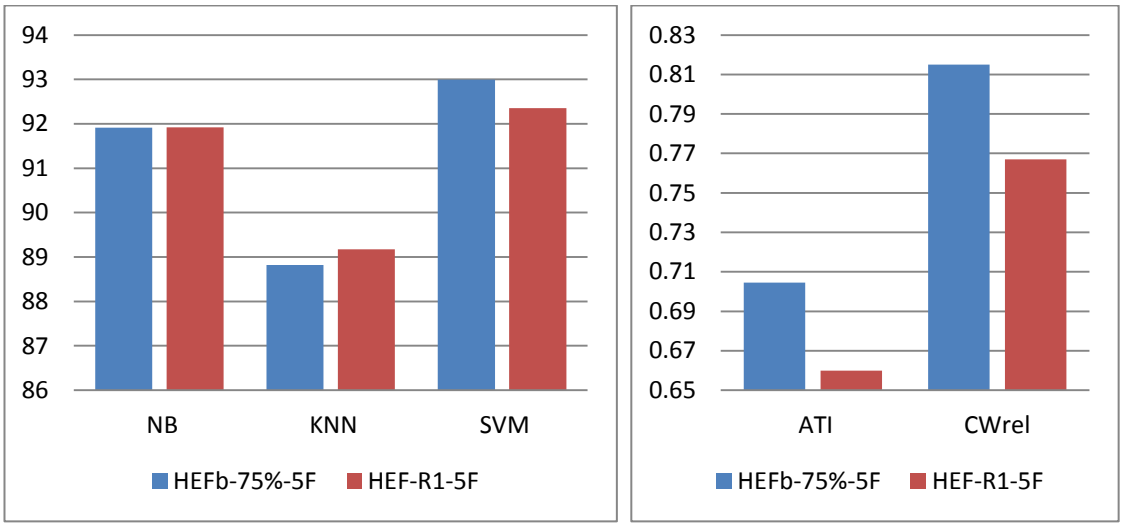


Figure 8. 2: a) Average test accuracy (b) Average stability over 10 real datasets with two different aggregation methods

Table 8.2 and Figure 8.2(a) show the test accuracy averages of the two aggregation methods according to three classifiers independent of the dataset. The highest accuracy was achieved by HEFb-75%-5F by SVM. HEF-R1-5F was slightly higher than HEFb-75%-5F by KNN, whereas HEFb-75%-5F and HEF-R1-5F had a similar average accuracy by NB.

Table 8.2 and Figure 8.2(b) show the average stability of two aggregation methods according to two similarity measures independent of the dataset. The highest stability was achieved by HEFb-75%-5F in both similarity measures.

In addition, we compared two ensemble methods with different aggregation methods according to the number of features selected.

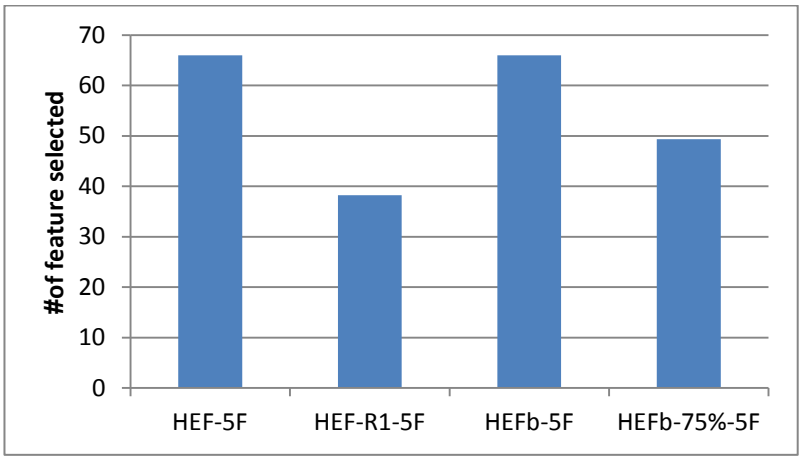


Figure 8.3: Average number of features selected using two different aggregation methods.

Figure 8.3 shows the average number of features selected using two aggregation methods independent of the dataset. HEF-R1-5F, which uses the heuristic consensus role in Section 4.2.3, selected fewer features on average than did the direct selection of the top 75% from the ranking feature in HEFb. These results show that the heuristic consensus role proposed in this thesis is better than the direct selection of the top 75% ranking features as a way to cut off the number of features in the ensemble algorithm. However, the accuracy results of both methods were not greatly affected as shown in Figure 8.2(a) while the stability was higher by HEFb-75%-5F.

Accordingly, we concluded that mean rank aggregation with a partial list (HEFb-75%-5F) improved the HEF in terms of stability and slightly improved the HEF in terms of accuracy. The results of HEFb-75%-5F were better than the results of HEF-R1-5F in most cases, particularly in terms of stability. Thus, in Chapter 7, we used HEFb to determine whether the WHEF further improved the HEFb-5F or not.

8.6 Weighed HEF

In Chapter 7, in order to improve the HEFb further, we assumed that the members in the HEFb should be weighted differently based on their accuracy. Thus, we investigated ways to determine the appropriate weight for each filter in the HEFb.

Three methods were proposed to investigate the impact of the weighted filters on the final ensemble results: the first one was the fixed weight HEF method (FWHEF), which assigns a fixed weight to the SF and less weight to the RF. The justification for this is

that many SF methods have been demonstrated to be more accurate in removing features that are both irrelevant and redundant than RF. The second method is called the variable weighted HEF method (VWHEF). It assigns variable weights to some filters, assuming that if a filter produces high accuracy, it can select more relevant and important features and vice versa, using the same classifier. Because VWHEF uses the classification accuracy values to compute the weights of each filter, a validation set is required. The third method is called selective filters HEF method (SFHEF). It assigns weights equal to one to some filters and assigns weights equal to zero to other filters. In other words, it selects some filters and discards others based on the validation set. The justification is that it ignores the features selected by the worst performing filters and focuses on the features selected by the best filters, aggregating them.

In order to demonstrate the capability of the proposed ensemble approaches in improving the results, we compared these three ensemble approaches with the simple HEFb using the same aggregation method.

The experimental results showed that the simple HEFb at all selection levels performed with a greater stability and consumed less time in all cases, whereas the accuracy of the three proposed ensembles did not significantly differ. Specifically, the results showed the following:

(1) No single best approach to all the situations could be found, in term of accuracy. In other words, the accuracy of each approach varied from dataset to dataset, and it was influenced by the type of classifier chosen for the model. Thus, one approach might perform well in a given dataset for a particular classifier but would perform poorly when used on a different dataset or with a different type of classifier.

(2) In averaging over 10 datasets, SFHEF and SFHEF-75% showed the highest accuracy with NB and KNN, and a slightly less accuracy with SVM. In contrast, they showed the lowest value when using only 25% of the selected features. The remaining ensemble approaches showed different average accuracies when using different classifiers. No preference should be given to one over the others, which was proved statistically by using the Nemenyi test, see Section 7.5.1.

(3) HEFb showed the highest stability for ATI and CWrel. This result further demonstrated that the simple ensemble HEF proposed by us was more reliable and consistent than the three weighted ensembles that were proposed later.

(4) Among the four categories of the feature selection, the selection of 75% of the top ranked features was the best choice in most cases compared with other selection categories in terms of accuracy and stability.

In summary, intuitively speaking, adding more weight to "good filters" should lead to better results but in reality it is very uncertain, simply because the assumption of 'good filters' does not always hold and is often untrue. This assumption was found to be correct for some examples in our experiment. However, for other situations, filters which had been assumed to perform well showed poor accuracy and hence led to even worse results. Overall, adding weight to filters might not achieve some much expected improvement in accuracy, but on the other hand it increases complexity and time consumption, and clearly decreases stability.

Consequently, HEFb-75% was identified (in Section 6.4) as being more reliable and consistent in most cases than HEF (Chapter 5), HEF+5F (in Section 6.3) and the three weighted ensembles that were proposed later (Chapter 7). Therefore, we consider HEFb-75% (Section 6.4) to be superior ensemble algorithm developed in this thesis. In the following section, we will compare HEFb-75% with the findings of other studies.

8.7 Comparison between HEF and Other Research

The comparison strategy of this research had two phases: Firstly, we compared the results obtained from our ensemble with the results obtained separately from each filter member. Secondly, we compared our ensemble results with other ensemble results, either our own previous ensemble studies or other ensemble studies in the literature. In Chapter 5, we compared the ensemble results in Chapter 5 with the ensemble results presented in Chapter 4. In addition, in Chapters 6 and 7, we compared the different versions within each chapter. Moreover, in this chapter, we compare our ensemble results with the findings of previous ensemble studies in the literature if they used the same datasets.

Table 8.3: Comparison of HEFb-75% with other EFS studies. Values given are average accuracy; parentheses show the number of features selected, and the last column presents the methods of other FS studies (FS + Classifier + Evaluation Scheme).

Data	**Our Results (HEF)**	**Some of the results report in the Literature**	
		Results	**Methods (FS + Classifier + Evaluation Scheme)**
Colon	NB 84.24 (35.25)	75.07(6)	MF-GE +NB +3 Fold CV (Yang et al., 2010)
	KNN 78.52(35.25)	70 (20) 79.2(40)	RF-Ensemble + 5NN+ 10 Fold CV (Saeys et al., 2008) En SVM-RFE + INN + 10 Fold CV (Han and Yu, 2012)
	SVM 86.83(35.25)	86.5(36) 74 (20) 82.5(40) 77.42(50)	MCF-RFE+SVM+632 Bootstrap (Yang and Mao, 2011) RF-Ensemble + SVM + 10 Fold CV (Saeys et al., 2008) En SVM-RFE + SVM + 10 Fold CV (Han and Yu, 2012) SVM-RFE + SVM + 10 Fold CV (Kalousis et al., 2007)
Leukaemia	NB 96.21(96.75)	95.27 (4.7)	MF-GE +NB +3 Fold CV (Yang et al., 2010)
	KNN 95.57 (96.75)	95.7 (50) 88 (71)	En SVM-RFE + INN + 10 Fold CV (Han and Yu, 2012) RF-Ensemble + 5NN + 10 Fold CV (Saeys et al., 2008)
	SVM 96.55(96.75)	96.5 (25) 96.8 (50) 91 (71)	MCF-RFE+SVM+632 Bootstrap (Yang and Mao, 2011) En SVM-RFE + SVM + 10 Fold CV (Han and Yu, 2012) RF-Ensemble + SVM + 10 Fold CV (Saeys et al., 2008)
Ovarian	KNN 99.48(60)	66 (151)	RF-Ensemble + 5NN + 10 Fold CV (Saeys et al., 2008)
	SVM 100(60)	82 (151) 99.60 (50)	RF-Ensemble + SVM + 10 Fold CV (Saeys et al., 2008) SVM-RFE + SVM + 10 Fold CV (Kalousis et al., 2007)

It is not always possible to make exact comparisons with the work of others because the differences in data pre-processing, accuracy evaluation schema and experimental design are not reported in enough detail to facilitate duplication. However, for comparison, we searched for FS studies that used the same datasets and classifiers as we used in our thesis. We then categorised the studies and presented them in two tables. Table 8.3 shows the comparison of our results with the most popular and latest EFS studies, which

used similar evaluation methods with more challenging datasets. The studies shown in this table will be evaluated and discussed in this section. Table C.1 provides a comparison of our results with the findings of several studies that used a single filter, wrapper or hybrid (which were not included as members in our ensemble) on the same benchmark dataset that we used. Table C.1 is shown in Appendix C because it contains a large number of different FS studies, which might require a long discussion in the main text. In addition, some of these studies are not recent, and some used different evaluation strategies.

Table 8.3 shows our results and those of other studies for the colon, leukaemia and ovarian datasets under comparable conditions. The results of HEFb-75% and other research studies will be compared according to average accuracy results and the number of features selected. We will start by comparing the HEFb-75% with the multi-filter enhanced genetic ensemble (MF-GE) proposed by Yang et al. (2010), which is similar to our ensemble algorithm, by applying multiple filtering.

The MF-GE algorithm is used to give scores for each candidate feature in the dataset. In addition, the softmax scaling process is used to compress the gene evaluation results of each filtering algorithm into the range of [0-1]. The algorithm used 3-fold CV instead of the 10-fold CV that we used.

We now compare MF-GE with HEFb-75% in the colon and leukaemia datasets using the NB classifier. In the colon dataset, an accuracy of 84.24% was achieved by selecting about 35 features, whereas MF-GE had an accuracy of 75.07% by selecting 6 features. Therefore, the results showed that HEFb-75% obtained higher accuracy than MF-GE by 9.17%, whereas the number of features in MF-GE was lower, which may be the reason for the reduced accuracy of MF-GE from which some relevant features were removed. A similar pattern was found in the leukaemia dataset, which has an accuracy of 96.21%, whereas the accuracy of MF-GE was 95.27%.

Saeys et al. (2008) used four FS algorithms (filter and embedded). An ensemble version was created by instance perturbation using bootstrap aggregation to generate 40 bags from the data. For each bag, a separate feature ranking was performed, and the ensemble was formed by aggregating the single rankings by using linear aggregation. The classification accuracy was assessed for accuracy by using a 10-fold cross-validation setting. For each fold, a feature selection was performed using only the

training part of the data, and a classifier was built using only the top 1% features returned by the feature selector.

The results of the colon dataset with KNN classifiers were as follows: the accuracy of HEFb-75% was 78.52% when about 35 features were selected. The accuracy of the RF-Ensemble (Random Forest) was 70% when 20 features were selected. Therefore, the results showed that HEFb-75% had higher accuracy than the RF-Ensemble by 8.52%, whereas the number of features in the RF-Ensemble was lower, which may the reason for the reduced accuracy; some relevant features were removed. Similar to the pattern with SVM classifiers, the accuracy was 86.83%, and the accuracy of the RF-Ensemble was 74%.

Similar results were found in the leukaemia dataset. KNN using HEFb-75% had 95.57% accuracy by selecting about 96 features, whereas the RF-Ensemble had a very low accuracy of 88% by selecting 71 features. Similar observations were made in the case of the SVM classifiers, which showed an accuracy of 96.55%, and the accuracy of the RF-Ensemble was 91%.

The ovarian dataset with KNN classifiers using HEFb-75% had an accuracy of 99.48% when 60 features were selected, whereas the RF-Ensemble showed an accuracy of 66% when about 151 features were selected. A high number of features is considered to produce very poor accuracy based on the strategy of using only the top 1% features returned by the feature selector. Similar observations were made in the case of SVM classifiers, where an accuracy of 100% was obtained by selecting 60 features. The RF-Ensemble obtained an accuracy of 82% by selecting about 151% features. In general, HEFb-75% showed better accuracy than the RF-Ensemble, although the RF-Ensemble had more stability than the single RF. According to (Saeys et al., 2008), ”Comparing the performance of the Random Forest ensemble feature selection version to the single version, it is clear that the substantial increase in robustness comes at a price, and results in lower accuracies for all datasets”.

Han and Yu (2012) applied SVM-RFE ensembles by using a bagging ensemble with 20 bootstrapped training sets to construct each ensemble. Then, to aggregate the different rankings into a final consensus ranking, the complete linear aggregation scheme summed the ranks of a feature based on all bootstrapped training sets. This study was similar to our study because it measured the average accuracy of 10 runs of 10-fold CV and used SVM and KNN as classifiers.

In the colon dataset, HEFb-75% showed an accuracy of 78.52% by selecting about 35 features, whereas EN-SVM-RFE showed an accuracy of 79% by selecting 40 features

with KNN. In addition, when the SVM classifier was used with the same data, the results showed that HEFb-75% had an accuracy of 86.83%, whereas the accuracy of EN-SVM-RFE was 82.5%. The results showed that HEFb-75% had a lower number of features in both classifiers, and its accuracy was higher than EN-SVM-RFE in SVM by 4.33%. In the leukaemia dataset, using KNN, HEFb-75% had an accuracy of 95.57% by selecting about 96 features, whereas EN-SVM-RFE had an accuracy of 95.7% by selecting 50 features. Moreover, in the same dataset, using SVM, HEHb-75% had an accuracy of 96.55%, whereas EN-SVM-RFE had an accuracy of 96.8%. The results showed that in the leukaemia dataset, the algorithms had similar accuracy, but EN-SVM-RFE had fewer features based on the cutting strategy, which started from the top 10 to 50 in increments of 10.

Yang and Mao (2011) proposed multi-criterion fusion-based recursive feature elimination (MCF-RFE), which integrated five different feature selection criteria, including Fisher's ratio, Relief, ADC (asymmetric dependency coefficient), AW-SVM (absolute weight of SVM) and SVM-RFE. Recursive feature elimination (RFE) is used as a search strategy to remove portions of the worst features. The accuracy was estimated using .632 bootstrap with 300 repeats.

The comparison between the HEFb-75% with MCF-RFE showed that the accuracy results were nearly similar in both datasets (colon and leukaemia). The accuracy of HEFb-75% in the colon dataset was 86.83% when about 35 features were selected. The accuracy of MCF-RFE was 86.5% when 36 features were selected. Moreover, in the leukaemia dataset, the accuracy of HEFb-75% was 96.55% when about 96 features were selected. The accuracy of MCF-RFE was 96.5% when 25 features were selected. The accuracy of MCF-RFE slightly increased to 97.8% when 100 features were selected.

Kalousis et al. (2007) studied SVM-RFE, which is based on repetitive applications of a linear support vector machine algorithm where the 10% lowest ranked features are eliminated at each iteration of the linear SVM. The ranks of the features are based on the absolute values of the coefficients assigned to them by the linear SVM. The results showed that HEFb-75% was more accurate than SVM-RFE in the colon and ovarian datasets. In the colon dataset, the accuracy of HEFb-75% was 86.83% when about 35 features were selected. The accuracy of SVM-RFE was 77.42 when 50 features were selected, which was lower than HEFb-75% by 9.41%. Similar observations were made

in the case of the ovarian dataset, where the accuracy of HEFb-75% was 100%, and the accuracy of SVM-RFE was 99.60%.

In fact, although previous research has discussed EFS, we cannot compare those findings with our results because they used different datasets and their software has not been made public. The majority of studies on EFS focused on binary datasets such as the colon, leukaemia and ovarian datasets. These datasets are easier than multi-class datasets to manipulate. Moreover, some members of the EFS studies cannot be used in multi-class datasets.

Based on these evaluations and comparisons, we conclude that our improved ensemble algorithm HEFb-75%-5F mostly performed better than the previously published methods in terms of classification accuracy and the number of selected features in the same datasets. Furthermore, in some cases, HEFb-75%-5F produced higher classification accuracy by using fewer features.

8.8 Summary

In this chapter, the work and the results of this thesis have been evaluated and discussed. It started by presenting an overview of the research as a whole, then evaluating our core algorithm (HEF) and its variations, then selecting the better one in most cases in terms of accuracy and stability. After that, it discussed the evaluation methods used to determined appropriate approaches for using data in feature selection. Also, it evaluated the aggregation methods used in this research. Then, it evaluated the idea of treating each filter differently, the methods that have been used to determine the appropriate weight, and the results of these three weighted ensemble algorithms. Finally, it presented a comparison between the proposed HEF and other ensemble studies in the literature.

Chapter 9

Conclusions

9.1 General Conclusions

In this thesis, we have developed an effective ensemble that can improve the accuracy and stability of feature selection. During the course of the research undertaken, we have achieved the following:

1. We developed a novel heuristic ensemble of filters (HEF) algorithm to improve the accuracy and stability of the selected features. Tested on the benchmark datasets, the proposed algorithm outperformed the other ensemble algorithms and individual filters, in most cases. The proposed HEF algorithm has the following characteristics: it can

- Handle binary and multi-class datasets.
- Apply any number and type of FS as members.
- Accelerate the ensemble algorithm by obtaining quick answers by appropriately cutting off the number of features in the ranker through running the subset filters.
- Use heuristic consensus rules to reduce the number of selected features in the filter ensemble and improve the accuracy of classification.
- Combine SF with RF to exploit the advantages of each, whereas the majority of the previous studies on feature selection ensembles focused only on ranking features.

2. The PART method is a more appropriate method for using data in feature selection than the ALL approaches. This work further extended previous works by comprehensively investigating the ALL and the PART methods on the filter method using four similarity and stability measures, in addition to the traditional way of evaluating FS, using 3 classifiers on 21 generated synthetic datasets, 10 real-world bench mark datasets and 4 synthetic benchmark datasets.

3. We proposed three novel schemes of mean rank aggregation with partial lists. The comparison results of these three novel schemes (HEFa, HEFb and HEFc) confirmed that ‘ranking the feature based on mean and making each feature not to appear in the list, with position equal to K+1 (HEF-b)’ is the ~~best~~ better scheme in terms of accuracy and stability, in most cases.

4. Adding wrappers after HEF (HHEF) did not contribute to improving the results, in this experiment. Although the results led to the selection of a very few features, accuracy and stability were reduced.

5. We proposed three novel schemes for a weighted heuristic ensemble of filters (WHEF). However, the experimental results demonstrated that adding weight to filters in a HEF does not achieve much of the expected improvement in accuracy, while it increases time and space complexity and clearly decreases stability. Therefore, the simple proposed ensemble algorithm (HEFb-75%-5F) was more reliable and consistent than the three weighted ensembles, which were proposed later.

9.2 Limitations

As in any research, this study has some limitations. The limitations of the research presented in this thesis are summarised as follows:

1. On the number of test datasets: this thesis developed an ensemble of feature selection that can improve the stability and accuracy of feature selection. The datasets we used have some general representations of real work problems, including different categories, numbers of features (ranging from 17 to 15,154), numbers of sample (ranging from 60 to 3,191) and different data shapes. Also, they include binary-class and multi-class classification problems; this should provide a basis for testing and should be well-suited to the feature selection methods under differing conditions. Hence, this research should be generally applicable to other problems. However, it will be better if further research is conducted using a greater number of datasets, to further validate the findings of this thesis.

2. On hybrid ensemble: this research adds three different wrappers (greedy forward search, linear forward selection and re-ranking search) after the HEF method to identify relevant feature subsets. Although there are only three, they are fairly representative wrappers. However, using other wrappers may improve the accuracy to some degree.

3- On the type of classifiers: this thesis used three classifiers (NB, KNN and SVM) to evaluate the HEF by measuring the salience of the selected features. These three classifiers have been chosen because they represent three quite different approaches in machine learning, and they do not contain any embedded feature selection mechanisms; also, they are commonly used in data mining practice. However, using additional classifiers such as linear classifier (LDA) to evaluate the HEF will enhance and validate the findings of this thesis.

4- In this research, we compare our ensemble results with the findings of previous ensemble studies in the literature if they used the same datasets and classifiers. However, the differences in data pre-processing make the comparisons with the work of others may not very precise.

9.3 Further Work

The research presented in this thesis can be extended for further research, some of which is summarised as follows:

1. This research proposed the HEF method, which combines multiple filters for ensemble feature selection. However, in this thesis, different types of filters were used with the HEF. It may be worthwhile to use a greater variety of filters as members such as MRMR (Peng et al., 2005) and INTERACT (Zhao and Liu, 2007). Using these additional members with HEF might enhance the understanding of the role of different members in the ensemble.

2. This thesis used different datasets, such as microarray datasets, that were relatively different in the number of features and the number of samples; they consisted of thousands of features. They also included binary-class and multi-class classifications. However, it might be suitable to extend productively to other datasets from different applications, such as text mining or image processing. The results of other datasets could be combined with the results presented in this thesis in order to further validate the findings of the thesis. Another potential extension would be to apply the HEF algorithms to an imbalanced dataset, which is not the focus of this research. It would be interesting to examine this area and to determine further results.

3. In order to determine approaches that are appropriate for using data in feature selection, we generated 21 synthetic datasets in an attempt to identify several problems, such as increasing the number of irrelevant features, decreasing the number of instances and varying the levels of noise in the response variable, all of which are factors that make the FS task difficult. However, all 21 datasets were linear problems. A further study could be performed to generate sophisticated synthetic datasets with non-linear problems to further investigate the ALL and PART methods.

4. In this research, we proposed three novel schemes to determine the weight of the members in HEF. A further study could be performed to investigate different ways to

determine the weight of filter members, especially since this area is new, which might offer considerable results.

5. This research uses three classifiers (NB, KNN and SVM) to evaluate the performance of HEF. Using additional classifiers such as linear classifiers (LDA) to evaluate the HEF will enhance and validate the findings of this thesis.

6- In this research, we compare our ensemble results with the findings of previous ensemble studies in the literature if they used the same datasets and classifiers. A further study could be performed to compare the HEF with others by running their algorithms and making sure that all other factors are similar to HEF. In addition to that, the feature selection competition should measure the significant difference between them.

References

ABEEL, T., HELLEPUTTE, T., VAN DE PEER, Y., DUPONT, P. & SAEYS, Y. 2010. Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. *Bioinformatics,* 26**,** 392-398.

AHA, D. W. & BANKERT, R. L. 1996. A comparative evaluation of sequential feature selection algorithms. *Learning from Data.* Springer.

AHA, D. W., KIBLER, D. & ALBERT, M. K. 1991. Instance-based learning algorithms. *Machine learning,* 6**,** 37-66.

AILON, N., CHARIKAR, M. & NEWMAN, A. 2008. Aggregating inconsistent information: ranking and clustering. *Journal of the ACM (JACM),* 55**,** 23.

ALMUALLIM, H. & DIETTERICH, T. G. 1991. Learning With Many Irrelevant Features. *In:* Proceedings of the Ninth National Conference on Artificial Intelligence 547-552

ALTIDOR, W., KHOSHGOFTAAR, T. M., VAN HULSE, J. & NAPOLITANO, A. 2011. Ensemble Feature Ranking Methods for Data Intensive Computing Applications. *Handbook of Data Intensive Computing.* Springer.

AMBROISE, C. 2002. Selection bias in gene extraction on the basis of microarray gene-expression data. *The National Academy of Sciences,* 99**,** 6562-6566.

AMBROISE, C. & MCLACHLAN, G. J. 2002. Selection bias in gene extraction on the basis of microarray gene-expression data. *Proceedings of the National Academy of Sciences,* 99**,** 6562-6566.

ASLAM, J. A. & MONTAGUE, M. 2001. Models for metasearch. *In:* Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 276-284.

AWADA, W., KHOSHGOFTAAR, T. M., DITTMAN, D., WALD, R. & NAPOLITANO, A. 2012. A review of the stability of feature selection techniques for bioinformatics data. *In:* 13th International Conference on Information Reuse and Integration IEEE, 356-363.

BELANCHE, L. & GONZÁLEZ, F. 2011. Review and evaluation of feature selection algorithms in synthetic problems. *arXiv:1101.2320*.

BENSCH, M., SCHRÖDER, M., BOGDAN, M. & ROSENSTIEL, W. 2005. Feature selection for high-dimensional industrial data. *In:* Proceedings of European Symposium on Artificial Neural Networks (ESANN).

BERMEJO, P., DE LA OSSA, L., GÁMEZ, J. A. & PUERTA, J. M. 2011. Fast wrapper feature subset selection in high-dimensional datasets by means of filter re-ranking. *Knowledge-Based Systems,* 25**,** 35-44.

BERMEJO, P., GÁMEZ, J. & PUERTA, J. 2008. On incremental wrapper-based attribute selection: experimental analysis of the relevance criteria. *Proceedings of IPMU'08***,** 638-645.

BERMEJO, P., GÁMEZ, J. A. & PUERTA, J. M. 2009. Incremental wrapper-based subset selection with replacement: An advantageous alternative to sequential forward selection. *In:* Symposium on Computational Intelligence and Data Mining. IEEE, 367-374.

BLUM, A. L. & LANGLEY, P. 1997. Selection of relevant features and examples in machine learning. *Artificial Intelligence,* 97**,** 245-271.

BOLÓN-CANEDO, V., SÁNCHEZ-MAROÑO, N., ALONSO-BETANZOS, A., BENÍTEZ, J. & HERRERA, F. 2014. A review of microarray datasets and applied feature selection methods. *Information Sciences,* 282**,** 111-135.

BOSER, B. E., GUYON, I. M. & VAPNIK, V. N. 1992. A training algorithm for optimal margin classifiers. *In:* Proceedings of the fifth annual workshop on Computational learning theory. ACM, 144-152.

BRASSARD, G. & BRATLEY, P. 1996. *Fundamentals of algorithmics*, Prentice Hall New York.

BREIMAN, L. 1996. Bagging predictors. *Machine learning,* 24**,** 123-140.

BURKOVSKI, A., LAUSSER, L., KRAUS, J. M. & KESTLER, H. A. 2014. Rank Aggregation for Candidate Gene Identification. *Data Analysis, Machine Learning and Knowledge Discovery.* Springer.

CANNAS, L. M., DESSÌ, N. & PES, B. 2013. Assessing similarity of feature selection techniques in high-dimensional domains. *Pattern Recognition Letters,* 34**,** 1446-1453.

CARUANA, R. & FREITAG, D. 1994. Greedy attribute selection. *In:* Proceedings of the eleventh international conference on machine learning. Citeseer, 28-36.

CHEN, Y., LI, Y., CHENG, X.-Q. & GUO, L. 2006. Survey and taxonomy of feature selection algorithms in intrusion detection system. *In:* Information Security and Cryptology. Springer, 153-167.

CHRYSOSTOMOU, K. A. 2008. The role of classifiers in feature selection: Number vs nature. *School of Information Systems, Computing and Mathematics*.

COPELAND, A. H. 1951. A reasonable social welfare function. *University of Michigan Seminar on Applications of Mathematics to the social sciences.*

CZEKAJ, T., WU, W. & WALCZAK, B. 2008. Classification of genomic data: Some aspects of feature selection. *Talanta,* 76**,** 564-574.

DASH, M. 1997. Feature Selection via Set Cover. *In:* Proceedings of the Knowledge and Data Engineering Exchange Workshop. IEEE Computer Society, 165.

DASH, M. & LIU, H. 1997. Feature selection for classification. *Intelligent data analysis,* 1**,** 131-156.

DASH, M. & LIU, H. 2003. Consistency-based search in feature selection. *Artificial Intelligence,* 151**,** 155-176.

DAVIES, D. L. & BOULDIN, D. W. 1979. A cluster separation measure. *Pattern Analysis and Machine Intelligence, IEEE Transactions on***,** 224-227.

DE, A., DIAZ, E. D. & RAGHAVAN, V. V. 2012. Weighted Fuzzy Aggregation for Metasearch: An Application of Choquet Integral. *Advances on Computational Intelligence.* Springer.

DE CASTRO, L. N. & VON ZUBEN, F. J. 2000. The clonal selection algorithm with engineering applications. *In:* Proceedings of GECCO. 36-39.

DECONDE, R. P., HAWLEY, S., FALCON, S., CLEGG, N., KNUDSEN, B. & ETZIONI, R. 2006. Combining results of microarray experiments: a rank aggregation approach. *Statistical Applications in Genetics and Molecular Biology,* 5**,** 1-25.

DEMŠAR, J. 2006. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research,* 7**,** 1-30.

DENG, K., HAN, S., LI, K. J. & LIU, J. S. 2014. Bayesian aggregation of order-based rank data. *Journal of the American Statistical Association,* 109**,** 1023-1039.

DEVIJVER, P. A. & KITTLER, J. 1982. *Pattern recognition: A statistical approach*, Prentice/Hall International Englewood Cliffs.

DIETTERICH, T. 2000. Ensemble methods in machine learning. *Multiple classifier systems***,** 1-15.

DIETTERICH, T. G. 1998. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural computation,* 10**,** 1895-1923.

DITTMAN, D. J., KHOSHGOFTAAR, T. M., WALD, R. & NAPOLITANO, A. 2013. Comparison of rank-based vs. score-based aggregation for ensemble gene selection. *In:* 14th International Conference on Information Reuse and Integration IEEE, 225-231.

DOAK, J. 1992. An evaluation of feature-selection methods and their application to computer security (Technical Report CSE-92-18). *Davis: University of California, Department of Computer Science*.

DWORK, C., KUMAR, R., NAOR, M. & SIVAKUMAR, D. 2001. Rank aggregation methods for the web. *In:* Proceedings of the 10th international conference on World Wide Web. ACM, 613-622.

FAGIN, R., KUMAR, R. & SIVAKUMAR, D. 2003. Comparing top k lists. *SIAM Journal on Discrete Mathematics,* 17**,** 134-160.

FAHAD, A., TARI, Z., KHALIL, I., ALMALAWI, A. & ZOMAYA, A. Y. 2014. An optimal and stable feature selection approach for traffic classification based on multi-criterion fusion. *Future Generation Computer Systems,* 36**,** 156-169.

FLEURET, F. 2004. Fast binary feature selection with conditional mutual information. *The Journal of Machine Learning Research,* 5**,** 1531-1555.

FREUND, Y. & SCHAPIRE, R. E. 1996. Experiments with a new boosting algorithm. *In:* In International Conference on Machine Learning. MORGAN KAUFMANN PUBLISHERS, INC., 148-156.

GAN, J. Q., HASAN, B. A. S. & TSUI, C. S. L. 2011. A hybrid approach to feature subset selection for brain-computer interface design. *Intelligent Data Engineering and Automated Learning-IDEAL* Springer.

GAN, J. Q., HASAN, B. A. S. & TSUI, C. S. L. 2014. A filter-dominating hybrid sequential forward floating search method for feature subset selection in high-dimensional space. *International Journal of Machine Learning and Cybernetics,* 5**,** 413-423.

GHEYAS, I. A. & SMITH, L. S. 2010. Feature subset selection in large dimensionality domains. *Pattern Recognition,* 43**,** 5-13.

GINSBERG, M. 1993. *Essentials of artificial intelligence*, Morgan Kaufmann.

GUTLEIN, M., FRANK, E., HALL, M. & KARWATH, A. 2009. Large-scale attribute selection using wrappers. *In:* Computational Intelligence and Data Mining. IEEE, 332-339.

GUYON, I. & ELISSEEFF, A. 2003. An introduction to variable and feature selection. *The Journal of Machine Learning Research,* 3**,** 1157-1182.

GUYON, I., WESTON, J., BARNHILL, S. & VAPNIK, V. 2002. Gene selection for cancer classification using support vector machines. *Machine learning,* 46**,** 389-422.

HALL, M. 2000. Correlation-based Feature Selection for Discrete and Numeric Class Machine Learning. *In:* Proceedings of 17th International Conference on Machine Learning. 359-366.

HALL, M. 2014. *Weka attribute selection with cross-valdation* [Online]. Available: http://weka.8497.n7.nabble.com/Weka-attribute-selection-with-cross-valdation-and-arff-output-td31254.html [Accessed].

HALL, M. A. 1999. *Correlation-based feature selection for machine learning.* The University of Waikato.

HALL, M. A. & HOLMES, G. 2003. Benchmarking attribute selection techniques for discrete class data mining. *Knowledge and Data Engineering, IEEE Transactions on,* 15**,** 1437-1447.

HALL, M. A. & SMITH, L. A. 1997. Feature subset selection: a correlation based filter approach. *In:* International Conference on Neural Information Processing and Intelligent Information Systems, Berlin. Springer, 855-858.

HAN, Y. & YU, L. 2012. A variance reduction framework for stable feature selection. *Statistical Analysis and Data Mining,* 5**,** 428-445.

HAURY, A.-C., GESTRAUD, P. & VERT, J.-P. 2011. The influence of feature selection methods on accuracy, stability and interpretability of molecular signatures. *PloS one,* 6.

HAYKIN, S. 1994. Neural networks: A comprehensive foundation: MacMillan College. *New York*.

HE, Z. & YU, W. 2010. Stable feature selection for biomarker discovery. *Computational Biology and Chemistry,* 34**,** 215–225.

HOETING, J. A., MADIGAN, D., RAFTERY, A. E. & VOLINSKY, C. T. 1999. Bayesian model averaging: a tutorial. *Statistical science***,** 382-401.

HSU, C. N., HUANG, H. J. & DIETRICH, S. 2002. The ANNIGMA-wrapper approach to fast feature selection for neural nets. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 32**,** 207-212.

HUANG, J., CAI, Y. & XU, X. 2007. A hybrid genetic algorithm for feature selection wrapper based on mutual information. *Pattern Recognition Letters,* 28**,** 1825-1844.

IENCO, D., PENSA, R. G. & MEO, R. 2009. Context-based distance learning for categorical data clustering. *Advances in Intelligent Data Analysis VIII.* Springer.

JAIN, A. & ZONGKER, D. 1997. Feature selection: Evaluation, application, and small sample performance. *Pattern Analysis and Machine Intelligence,* 19**,** 153-158.

JOHN, G. H., KOHAVI, R. & PFLEGER, K. 1994. Irrelevant features and the subset selection problem. *In:* Proceeding of the Eleventh International Machine Learning, San Francisco. Morgan Kaufmann, 121-129.

JOHN, G. H. & LANGLEY, P. 1995. Estimating continuous distributions in Bayesian classifiers. *In:* Proceedings of the eleventh conference on uncertainty in artificial intelligence, San Francisco, CA, USA. Morgan Kaufmann, 338-345.

JURMAN, G., MERLER, S., BARLA, A., PAOLI, S., GALEA, A. & FURLANELLO, C. 2008. Algebraic stability indicators for ranked lists in molecular profiling. *Bioinformatics,* 24**,** 258-264.

KABIR, M. & ISLAM, M. 2010. A new wrapper feature selection approach using neural network. *Neurocomputing,* 73**,** 3273-3283.

KALOUSIS, A., PRADOS, J. & HILARIO, M. 2007. Stability of feature selection algorithms: a study on high-dimensional spaces. *Knowledge and information systems,* 12**,** 95-116.

KAREGOWDA, A. G., MANJUNATH, A. & JAYARAM, M. 2010. Comparative study of attribute selection using gain ratio and correlation based feature selection. *International Journal of Information Technology and Knowledge Management,* 2**,** 271-277.

KHOSHGOFTAAR, T. M., WALD, R., DITTMAN, D. J. & NAPOLITANO, A. 2013. Feature list aggregation approaches for ensemble gene selection on patient response datasets. *In:* 14th International Conference on Information Reuse and Integration (IRI),. IEEE, 317-324.

KIRA, K. & RENDELL, L. A. 1992. A practical approach to feature selection. *In:* Proceedings of the ninth international workshop on Machine learning. Morgan Kaufmann Publishers Inc., 249-256.

KIRKPATRICK, S. & VECCHI, M. P. 1983. Optimization by simulated annealing. *Science,* 220**,** 671-680.

KITTLER, J. 1978. Feature set search algorithms. *Pattern recognition and signal processing***,** 41-60.

KOHAVI, R. & JOHN, G. H. 1997. Wrappers for feature subset selection. *Artificial Intelligence,* 97**,** 273-324.

KOHAVI, R. & SOMMERFIELD, D. 1995. Feature Subset Selection Using the Wrapper Method: Overfitting and Dynamic Search Space Topology. *In:* KDD. 192-197.

KOLDE, R., LAUR, S., ADLER, P. & VILO, J. 2012. Robust rank aggregation for gene list integration and meta-analysis. *Bioinformatics,* 28**,** 573-580.

KOLLER, D. & SAHAMI, M. 1996. Toward optimal feature selection.

KONONENKO, I. 1994. Estimating attributes: analysis and extensions of RELIEF. *Proceedings of European Conference on Machine Learning Catania.* Italy: Springer.

KŘÍŽEK, P., KITTLER, J. & HLAVÁČ, V. 2007. Improving stability of feature selection methods. *In:* Computer Analysis of Images and Patterns. Springer, 929-936.

KUNCHEVA, L. I. 2007. A stability index for feature selection. *In:* Proceedings of the 25th IASTED International Multi-Conference: artificial intelligence and applications. ACTA Press, 390-395.

LECOCKE, M. & HESS, K. 2006. An empirical study of univariate and genetic algorithm-based feature selection in binary classification with microarray data. *Cancer informatics,* 2**,** 313-327.

LEUNG, Y. & HUNG, Y. 2010. A multiple-filter-multiple-wrapper approach to gene selection and microarray data classification. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB),* 7**,** 108-117.

LI, T., ZHANG, C. & OGIHARA, M. 2004. A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression. *Bioinformatics,* 20**,** 2429-2437.

LI, Y. & GUO, L. 2008. TCM-KNN scheme for network anomaly detection using feature-based optimizations. *In:* Proceedings of the ACM symposium on Applied computing, New York. ACM, 2103-2109.

LIANG, J., YANG, S. & WANG, Y. 2009. An optimal feature subset selection method based on distance discriminant and distribution overlapping. *International Journal of Pattern Recognition and Artificial Intelligence,* 23**,** 1577-1597.

LILLIS, D., TOOLAN, F., COLLIER, R. & DUNNION, J. 2006. Probfuse: a probabilistic approach to data fusion. *In:* Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 139-146.

LIN, S. 2010. Space oriented rank-based data integration. *Statistical Applications in Genetics and Molecular Biology,* 9.

LIN, S. & DING, J. 2009. Integration of ranked lists via Cross Entropy Monte Carlo with applications to mRNA and microRNA studies. *Biometrics,* 65**,** 9-18.

LIU, H., LI, J. & WONG, L. 2002a. A comparative study on feature selection and classification methods using gene expression profiles and proteomic patterns. *Genome Informatics Series***,** 51-60.

LIU, H., LIU, L. & ZHANG, H. 2008. Feature selection using mutual information: An experimental study. *PRICAI 2008: Trends in Artificial Intelligence.* Springer.

LIU, H., MOTODA, H. & YU, L. 2002b. Feature Selection with Selective Sampling. *Proceedings of the Nineteenth International Conference on Machine Learning.* Morgan Kaufmann Publishers Inc.

LIU, H. & SETIONO, R. 1995. Chi2: Feature selection and discretization of numeric attributes. *In:* Seventh International Conference on Tools with Artificial Intelligence. IEEE, 388-391.

LIU, H. & SETIONO, R. 1996a. Feature selection and classification-a probabilistic wrapper approach. *In:* the 9th International Conference on Industrial and Engineering Applications of AI and ES. 419-424.

LIU, H. & SETIONO, R. 1996b. A probabilistic approach to feature selection-a filter solution. *In:* International Conference of Machine Learning. Citeseer, 319-327.

LIU, H. & SETIONO, R. 1997. Feature selection via discretization. *IEEE Transactions on knowledge and Data Engineering,* 9**,** 642-645.

LIU, H. & YU, L. 2005. Toward integrating feature selection algorithms for classification and clustering. *Knowledge and Data Engineering,* 17**,** 491-502.

LIU, Y.-T., LIU, T.-Y., QIN, T., MA, Z.-M. & LI, H. 2007. Supervised rank aggregation. *In:* Proceedings of the 16th international conference on World Wide Web. ACM, 481-490.

LOSCALZO, S., YU, L. & DING, C. 2009. Consensus group stable feature selection. *In:* the 15th international conference on Knowledge discovery and data mining. ACM, 567-576.

LOUGHREY, J. & CUNNINGHAM, P. 2005a. Overfitting in wrapper-based feature subset selection: The harder you try the worse it gets. *Research and Development in Intelligent Systems XXI***,** 33-43.

LOUGHREY, J. & CUNNINGHAM, P. 2005b. Using early-stopping to avoid overfitting in wrapper-based feature selection employing stochastic search. Trinity College Dublin, Department of Computer Science.

MALDONADO, S. & WEBER, R. 2009. A wrapper method for feature selection using Support Vector Machines. *Information Sciences: an International Journal,* 179**,** 2208-2217.

MARTÍN-SMITH, P., ORTEGA, J., ASENSIO-CUBERO, J., GAN, J. Q. & ORTIZ, A. 2015. A Label-Aided Filter Method for Multi-objective Feature Selection in EEG Classification for BCI. *Advances in Computational Intelligence.* Springer.

MICHIELS, S., KOSCIELNY, S. & HILL, C. 2005. Prediction of cancer outcome with microarrays: a multiple random validation strategy. *The Lancet,* 365**,** 488-492.

MIN, H. & FANGFANG, W. 2010. Filter-wrapper hybrid method on feature selection. *In:* The Second WRI Global Congress on Intelligent Systems (GCIS). IEEE, 98-101.

MITCHELL, T. M. 1997. Machine learning. 1997. *Burr Ridge, IL: McGraw Hill,* 45.

MOORE, J. & WHITE, B. 2007. Tuning ReliefF for genome-wide genetic analysis. *Evolutionary computation, machine learning and data mining in bioinformatics***,** 166-175.

NEUMAYER, R., MAYER, R. & NØRVÅG, K. 2011. Combination of feature selection methods for text categorisation. *Advances in Information Retrieval.* Springer.

NG, H. T., GOH, W. B. & LOW, K. L. 1997. Feature selection, perceptron learning, and a usability case study for text categorization. *In:* ACM SIGIR Forum. ACM, 67-73.

OLSSON, J. & OARD, D. W. 2006. Combining feature selectors for text classification. *In:* Proceedings of the 15th ACM international conference on Information and knowledge management. ACM, 798-799.

OOI, C. H., CHETTY, M. & TENG, S. W. 2006. Differential prioritization between relevance and redundancy in correlation-based feature selection techniques for multiclass gene expression data. *BMC Bioinformatics,* 7**,** 320.

OZA, N. C. 2000. Ensemble Data Mining Methods. *NASA Ame Research Center*.

PENG, H., LONG, F. & DING, C. 2005. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* 27**,** 1226-1238.

PLATT, J. C. 1999. Fast Training of Support Vector Machines using Sequential Minimal Optimization. *In:* BERNHARD SCHÖLKOPF, AND, C. J. C. B. & SMOLA, A. J. (eds.) *Advances in Kernel Methods* Cambridge, MA, USA: MIT Press.

PRATI, R. C. 2012. Combining feature ranking algorithms through rank aggregation. *In:* The 2012 International Joint Conference on Neural Networks (IJCNN), . IEEE, 1-8.

PUDIL, P., NOVOVIČOVÁ, J. & KITTLER, J. 1994. Floating search methods in feature selection. *Pattern Recognition Letters,* 15**,** 1119-1125.

QUINLAN, J. R. 1986. Induction of decision trees. *Machine learning,* 1**,** 81-106.

QUINLAN, J. R. 1993. *C4. 5: programs for machine learning*, Morgan kaufmann.

RAMAN, B. & IOERGER, T. R. 2002. Instance based filter for feature selection. *Journal of Machine Learning Research,* 1**,** 1-23.

REFAEILZADEH, P., TANG, L. & LIU, H. 2007. On comparison of feature selection algorithms. *In:* Proceedings of AAAI Workshop on Evaluation Methods for Machine Learning II. 34-39.

REUNANEN, J. 2003. Overfitting in making comparisons between variable selection methods. *The Journal of Machine Learning Research,* 3**,** 1371-1382.

RICHELDI, M. & LANZI, P. L. 1996. ADHOC: A tool for performing effective feature selection. *In:* Proceedings Eighth IEEE International Conference on Tools with Artificial Intelligence. IEEE, 102-105.

ROBNIK-ŠIKONJA, M. & KONONENKO, I. 2003. Theoretical and empirical analysis of ReliefF and RReliefF. *Machine learning,* 53**,** 23-69.

RUIZ, R., AGUILAR, J. & RIQUELME, J. 2008. Best agglomerative ranked subset for feature selection. *In:* Workshop and Conference Proceedings of JMLR. 148-162.

RUIZ, R., RIQUELME, J. C. & AGUILAR-RUIZ, J. S. 2006. Incremental wrapper-based gene selection from microarray data for cancer classification. *Pattern Recognition,* 39**,** 2383-2392.

SAEYS, Y., ABEEL, T. & VAN DE PEER, Y. 2008. Robust feature selection using ensemble feature selection techniques. *Machine Learning and Knowledge Discovery in Databases***,** 313-325.

SAEYS, Y., INZA, I. & LARRANAGA, P. 2007. A review of feature selection techniques in bioinformatics. *Bioinformatics,* 23**,** 2507-2517.

SARKAR, C., COOLEY, S. & SRIVASTAVA, J. 2012. Improved feature selection for hematopoietic cell transplantation outcome prediction using rank aggregation. *In:* Federated Conference on Computer Science and Information Systems (FedCSIS),. IEEE, 221-226.

SARKAR, C., COOLEY, S. & SRIVASTAVA, J. 2014. Robust feature selection technique using rank aggregation. *Applied Artificial Intelligence,* 28**,** 243-257.

SARKAR, C., DESIKAN, P. & SRIVASTAVA, J. 2013. Correlation based Feature Selection using Rank aggregation for an Improved Prediction of Potentially Preventable Events.

SCHLIMMER, J. C. 1993. Efficiently inducing determinations: A complete and systematic search algorithm that uses optimal pruning. *In:* In Proceedings of the Tenth International Conference on Machine Learning. Citeseer, 284-290.

SCHOWE, B. 2011. Feature Selection for high-dimensional data with RapidMiner. *In:* Proceedings of the 2nd RapidMiner Community Meeting And Conference

SETIONO, R. & LIU, H. 1997. Neural-network feature selector. *Neural Networks, IEEE Transactions on,* 8**,** 654-662.

SINGH, S. & SILAKARI, S. 2009. An ensemble approach for feature selection of Cyber Attack Dataset. *Arxiv preprint arXiv:0912.1014*.

SINGHI, S. K. & LIU, H. 2006. Feature subset selection bias for classification learning. *Proceedings of the 23rd international conference on Machine learning.* ACM.

SOMOL, P. & NOVOVICOVA, J. 2010. Evaluating Stability and Comparing Output of Feature Selectors that Optimize Feature Subset Cardinality. *Pattern Analysis and Machine Intelligence,* 32**,** 1921-1939.

SOMOL, P. & NOVOVIČOVÁ, J. 2008. Evaluating the stability of feature selectors that optimize feature subset cardinality. *Structural, Syntactic, and Statistical Pattern Recognition.* Springer.

SOMOL, P., NOVOVIČOVÁ, J. & PUDIL, P. 2006. Flexible-hybrid sequential floating search in statistical feature selection. *Structural, Syntactic, and Statistical Pattern Recognition.* Springer.

SOMOL, P., PUDIL, P. & KITTLER, J. 2004. Fast branch & bound algorithms for optimal feature selection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* 26**,** 900-912.

TAN, P. N., STEINBACH, M. & KUMAR, V. 2006. *Introduction to data mining*, Pearson Addison Wesley Boston.

THRUN, S. B., BALA, J. W., BLOEDORN, E., BRATKO, I., CESTNIK, B., CHENG, J., DE JONG, K. A., DZEROSKI, S., FISHER, D. H. & FAHLMAN, S. E. 1991. The monk's problems: A performance comparison of different learning algorithms.

TONGCHIM, S., SORNLERTLAMVANICH, V. & ISAHARA, H. 2007. Examining the feasibility of metasearch based on results of human judgements on thai queries. *In:* 21st International Conference on Advanced Information Networking and Applications Workshops. IEEE, 283-288.

TUV, E., BORISOV, A., RUNGER, G. & TORKKOLA, K. 2009. Feature selection with ensembles, artificial variables, and redundancy elimination. *The Journal of Machine Learning Research,* 10**,** 1341-1366.

VEGE, S. H. 2012. Ensemble of Feature Selection Techniques for High Dimensional Data.

WALD, R., KHOSHGOFTAAR, T. M., DITTMAN, D., AWADA, W. & NAPOLITANO, A. 2012. An extensive comparison of feature ranking aggregation techniques in bioinformatics. *In:* The 13th International Conference on Information Reuse and Integration (IRI). IEEE, 377-384.

WANG, D. & LI, T. 2012. Weighted consensus multi-document summarization. *Information Processing & Management,* 48**,** 513-523.

WANG, H., KHOSHGOFTAAR, T. & GAO, K. 2010a. Ensemble feature selection technique for software quality classification. *In:* Proceedings of the 22nd International Conference on Software Engineering and Knowledge Engineering. 215-220.

WANG, H., KHOSHGOFTAAR, T. M. & NAPOLITANO, A. 2010b. A comparative study of ensemble feature selection techniques for software defect prediction. *In:* Ninth International Conference on Machine Learning and Applications (ICMLA). IEEE, 135-140.

WANG, H., KHOSHGOFTAAR, T. M. & NAPOLITANO, A. 2012. Software measurement data reduction using ensemble techniques. *Neurocomputing*.

WANG, H., KHOSHGOFTAAR, T. M., VAN HULSE, J. & GAO, K. 2011. Metric selection for software defect prediction. *International Journal of Software Engineering and Knowledge Engineering,* 21**,** 237-257.

WANG, W. 2008. Some fundamental issues in ensemble methods. *In:* International Joint Conference on Neural Networks. IEEE, 2243-2250.

WITTEN, I. H. & FRANK, E. 2000. WEKA (Waikato Environment for Knowledge Analysis). *Internet: http://www. cs. waikato. ac. nz/ml/weka,[Mar. 2, 2008]*.

WITTEN, I. H. & FRANK, E. 2005. Data Mining: Practical Machine Learing Tools and and Techniques.

XU, J., SUN, L., GAO, Y. & XU, T. 2013. An ensemble feature selection technique for cancer recognition. *Bio-medical materials and engineering,* 24**,** 1001-1008.

YANG, F. & MAO, K. 2011. Robust feature selection for microarray data based on multicriterion fusion. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB),* 8**,** 1080-1092.

YANG, P., HO, J. W. K., YANG, Y. & ZHOU, B. B. 2011. Gene-gene interaction filtering with ensemble of filters. *BMC Bioinformatics,* 12**,** S10.

YANG, P., ZHOU, B. B., ZHANG, Z. & ZOMAYA, A. Y. 2010. A multi-filter enhanced genetic ensemble system for gene selection and sample classification of microarray data. *BMC Bioinformatics,* 11**,** S5.

YU, L., DING, C. & LOSCALZO, S. 2008. Stable feature selection via dense feature groups. *In:* Proceedings of the 14th international conference on Knowledge discovery and data mining. ACM, 803-811.

YU, L. & LIU, H. 2003. Feature selection for high-dimensional data: A fast correlation-based filter solution. *In:* Proceedings of the Twentieth International Conference on Machine Learning. 856.

YU, L. & LIU, H. 2004. Efficient Feature Selection via Analysis of Relevance and Redundancy. *The Journal of Machine Learning Research,* 5**,** 1205-1224.

YUN, C. & YANG, J. 2007. Experimental comparison of feature subset selection methods. *In:* Seventh IEEE International Conference on Data Mining Workshops. IEEE, 367-372.

ZHANG, L.-X., WANG, J.-X., ZHAO, Y.-N. & YANG, Z.-H. 2003. A novel hybrid feature selection algorithm: using ReliefF estimation for GA-Wrapper search. *In:* International Conference on Machine Learning and Cybernetics. IEEE, 380-384.

ZHANG, M., ZHANG, L., ZOU, J., YAO, C., XIAO, H., LIU, Q., WANG, J., WANG, D., WANG, C. & GUO, Z. 2009. Evaluating reproducibility of differential expression discoveries in microarray studies by considering correlated molecular changes. *Bioinformatics,* 25**,** 1662-1668.

ZHANG, Y. & ZHANG, Z. 2011. Feature subset selection with cumulate conditional mutual information minimization. *Expert Systems with Applications*.

ZHAO, Z. & LIU, H. 2007. Searching for interacting features. *In:* Proceedings of the 20th international joint conference on Artifical intelligence. Morgan Kaufmann Publishers Inc., 1156-1161.

ZHU, Z., ONG, Y.-S. & ZURADA, J. M. 2010. Identification of full and partial class relevant genes. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on,* 7**,** 263-277.

ZHU, Z., ONG, Y. S. & DASH, M. 2007. Wrapper–filter feature selection algorithm using a memetic framework. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on,* 37**,** 70-76.

Appendices

Appendix A: Further results from Chapter 5

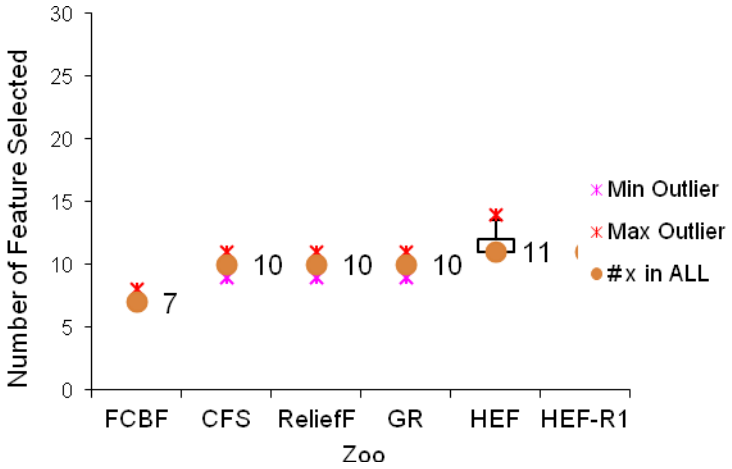


Figure A.1: Number of selected features by the PART method on the Zoo dataset.

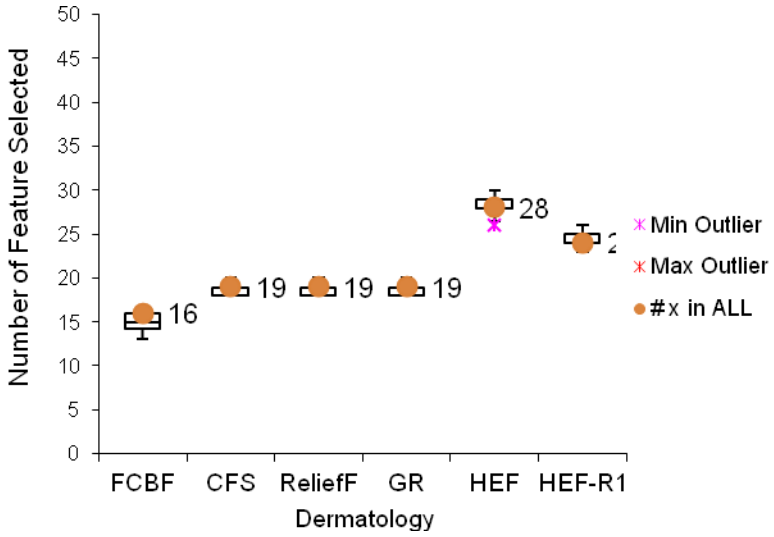


Figure A.2: Number of selected features by the PART method on the Dermatology dataset.

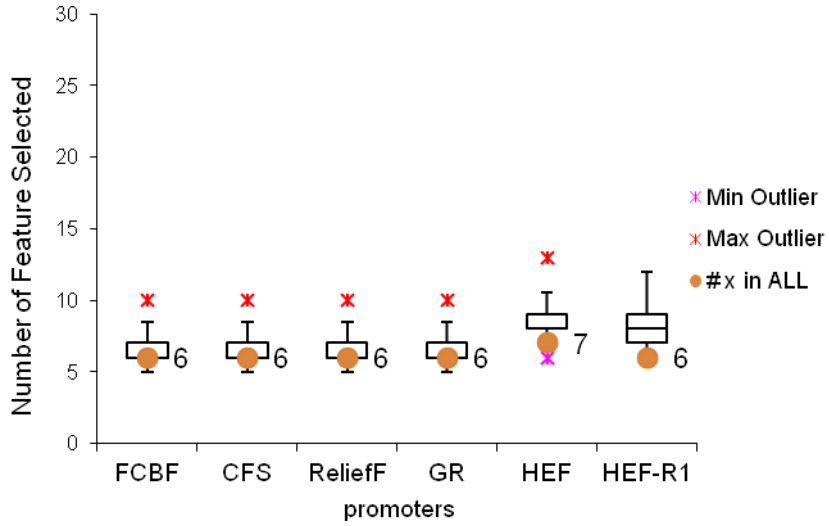


Figure A.3: Number of selected features by the PART method on the Promoters dataset.

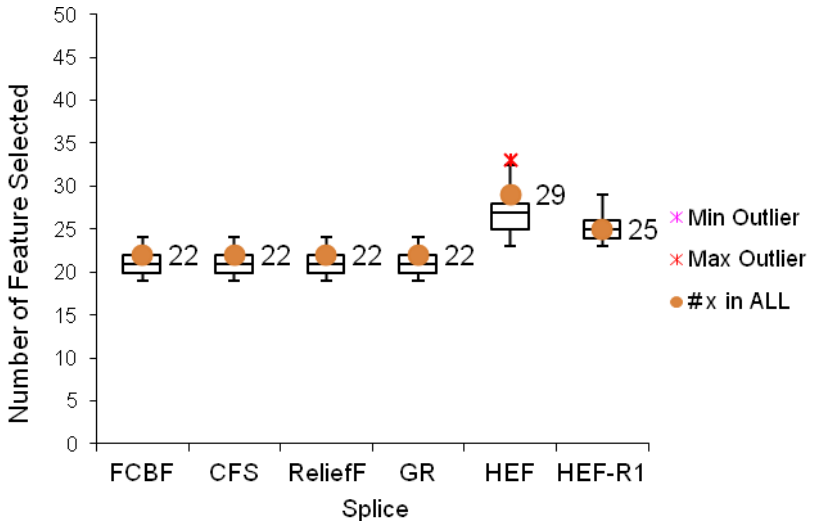


Figure A.4: Number of selected features by the PART method on the Splice dataset.

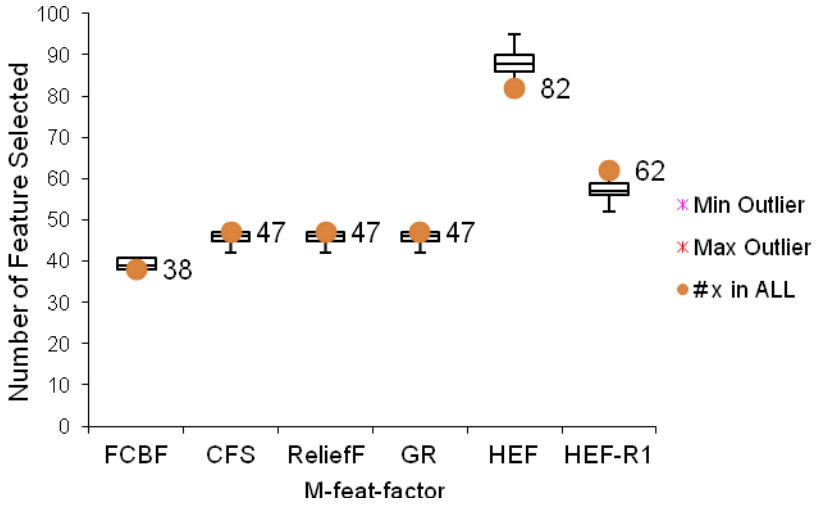


Figure A.5: Number of selected features by the PART method on the M-feat-factor dataset.

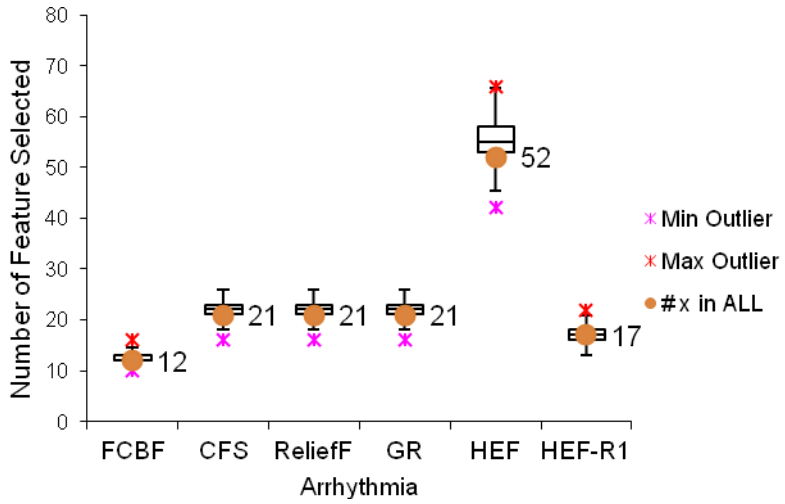


Figure A.6: Number of selected features by the PART method on the Arrhythmia dataset.

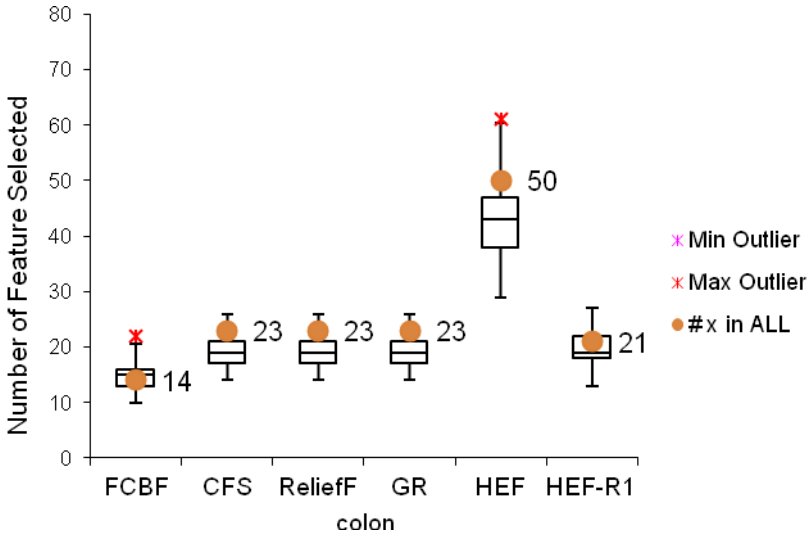


Figure A.7: Number of selected features by the PART method on the colon dataset.

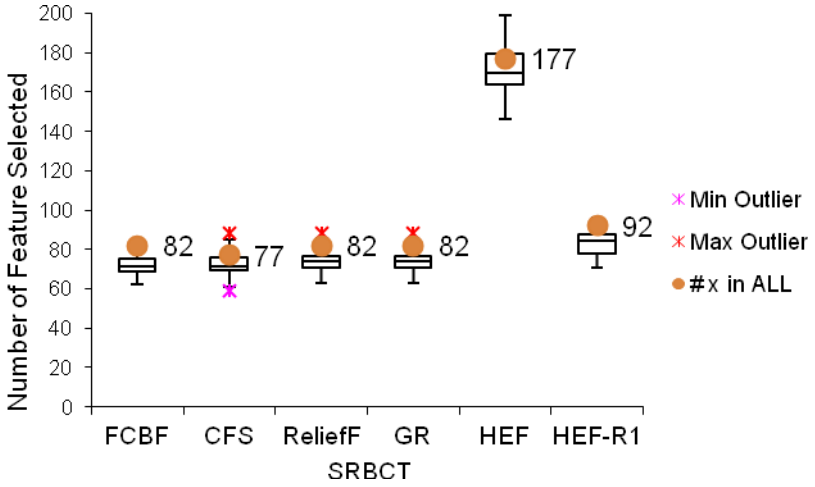


Figure A.8: Number of selected features by the PART method on the SRBCT dataset.

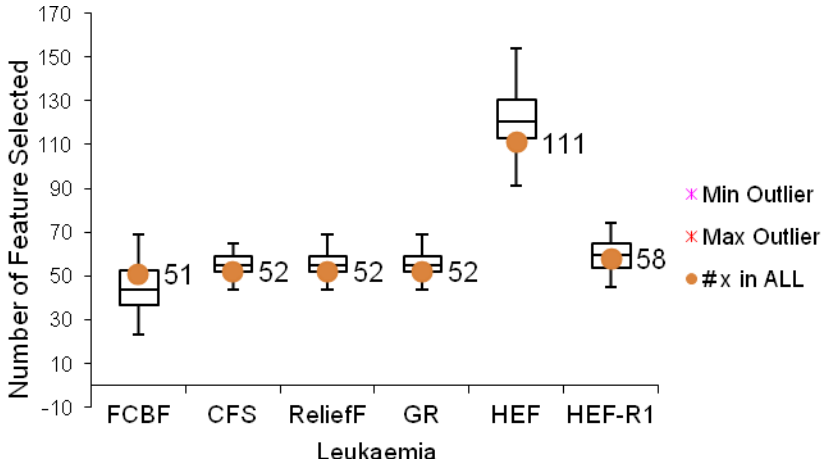


Figure A.9: Number of selected features by the PART method on the Leukaemia dataset.

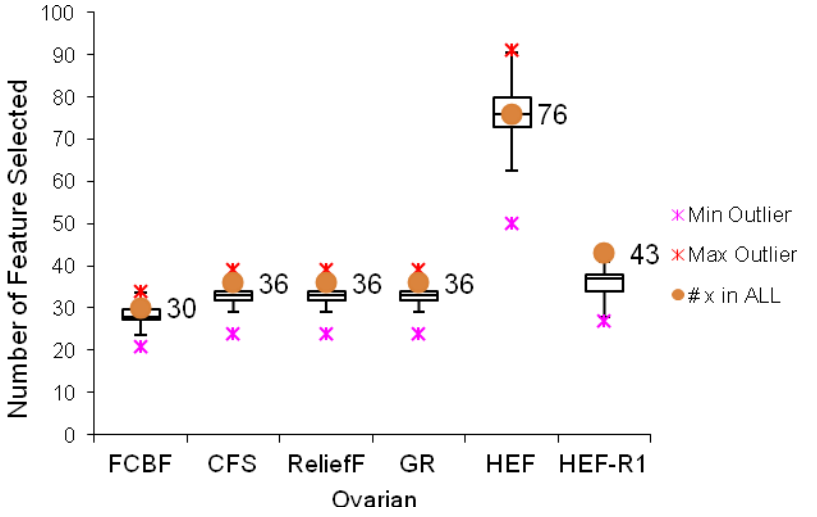


Figure A.10: Number of selected features by the PART method on the Ovarian dataset.

Appendix B: Further results from Chapter 7

Table B.1: The average test accuracy of NB classifiers trained with the features selected by HEF, FWHEF, VWHEF and SFHEF using 50% and 25% of these features being selected

Dataset	HEF-50%	HEF-25%	FWHEF-50%	FWHEF-25%	VWHEF-50%	VWHEF-25%	SFHEF-50%	SFHEF-25%
Zoo	91.58	88.41	91.3	85.35	91.01	85.26	91.11	76.83
Dermatology	90.6	83.66	97.32	86.14	97.05	84.99	94.58	79.93
Promoters	94.56	83.32	93.15	85.84	94.38	84.01	93.39	81.32
Splice	95.43	93.74	94.9	93.74	95.47	93.74	94.54	91.81
M-feat-fact	92.77	91.58	92.89	91.37	92.66	90.51	92.99	90.53
Arrhythmia	66.11	63.23	68.7	68.14	67.7	67.15	67.81	66.53
Colon	85.4	85.33	83.69	82.83	85.07	86.1	85.81	85.02
SRBCT	98.45	96.74	97.95	96.4	98.93	97.02	98.58	98.56
Leukaemia	96.09	95.52	96.23	95.8	96.23	95.66	95.8	96.07
Ovarian	98.53	98.13	99.48	99.13	98.57	98.61	99.53	99.4
Average	90.952	87.966	91.561	88.474	**91.707**	88.305	91.414	86.6

Table B.2 The average test accuracy of KNN classifiers trained with the features selected by HEF, FWHEF, VWHEF and SFHEF using 50% and 25% of these features being selected

Dataset	HEF-50%	HEF-25%	FWHEF-50%	FWHEF-25%	VWHEF-50%	VWHEF-25%	SFHEF-50%	SFHEF-25%
Zoo	93.45	90.39	93.06	87.12	92.86	88.62	92.76	81.39
Dermatology	90.26	83.41	95.32	85.28	95.13	84.08	93.14	79.08
Promoters	88.43	84.09	87.56	85.74	87.59	84.16	88.88	80.59
Splice	84.39	89.61	84.13	89.6	84.33	89.61	86.03	90.01
M-feat-fact	95.94	95.23	96.5	95.92	95.43	94.36	95.96	94.42
Arrhythmia	56.0	57.86	59.9	58.57	58.2	58.6	55.78	54.17
Colon	77.5	80.26	79.1	81.17	77.48	79.21	80.12	77.67
SRBCT	99.64	99.67	99.15	99.29	99.88	99.31	99.54	98.68
Leukaemia	95.7	92.0	95.28	94.0	95.71	94.45	95.0	93.91
Ovarian	99.56	99.37	99.65	99.84	99.68	99.65	99.96	99.84
Average	88.087	87.189	**88.965**	87.653	88.629	87.205	88.717	84.976

Table B.3: The average test accuracy of SVM classifiers trained with the features selected by HEF, FWHEF, VWHEF and SFHEF using 50% and 25% of these features being selected

Dataset	HEF-50%	HEF-25%	FWHEF-50%	FWHEF-25%	VWHEF-50%	VWHEF-25%	SFHEF-50%	SFHEF-25%
Zoo	93.45	91.09	92.87	86.2	93.07	87.04	92.28	81.0
Dermatology	90.35	84.07	96.83	85.47	96.2	84.27	94.13	78.91
Promoters	94.25	81.13	92.5	82.85	93.6	82.02	92.55	78.3
Splice	95.69	94.43	95.34	94.43	95.73	94.43	95.46	91.91
M-feat-fact	97.16	96.07	97.42	96.54	**97.14**	**96.09**	96.71	94.43
Arrhythmia	64.94	61.84	66.93	62.95	66.18	62.95	65.07	61.62
Colon	85.83	84.1	85.02	85.24	85.07	83.95	86.05	84.72
SRBCT	99.78	99.53	99.29	99.29	99.53	98.95	99.2	98.54
Leukaemia	96.5	95.5	96.52	96.23	96.37	96.23	96.1	95.0
Ovarian	100.0	99.68	100	99.88	**100.0**	**99.88**	99.96	99.84
Average	91.795	88.744	92.272	88.908	**92.289**	88.581	91.751	86.427

Table B.4: The stability measures of ATI with the features selected by four ensembles approaches over 10 runs of 10-fold cross-validation.

ATI	HEF-50%	HEF-25%	FWHEF-50%	FWHEF-25%	VWHEF-50%	VWHEF-25%	SFHEF-50%	SFHEF-25%
Zoo	0.97	0.81	0.87	1.0	0.81	0.81	0.74	0.59
Dermatology	0.92	0.76	0.76	0.72	0.78	0.72	0.68	0.65
Promoters	0.86	0.8	0.73	0.72	0.82	0.79	0.81	0.73
Splice	0.9	0.91	0.73	0.9	0.9	0.91	0.86	0.85
M-feat-fact	0.74	0.7	0.58	0.5	0.75	0.6	0.56	0.57
Arrhythmia	0.65	0.55	0.57	0.47	0.55	0.47	0.43	0.46
Colon	0.48	0.58	0.39	0.42	0.5	0.62	0.5	0.57
SRBCT	0.54	0.43	0.53	0.45	0.52	0.35	0.36	0.25
Leukaemia	0.38	0.31	0.3	0.3	0.38	0.31	0.34	0.39
Ovarian	0.54	0.75	0.37	0.45	0.52	0.66	0.39	0.48
Average	**0.698**	0.66	0.583	0.593	0.653	0.624	0.567	0.554

Table B.5: The stability measures of CWrel with the features selected by four ensembles approaches over 10 runs of 10-fold cross-validation

CWrel	HEF-50%	HEF-25%	FWHEF-50%	FWHEF-25%	VWHEF-50%	VWHEF-25%	SFHEF-50%	SFHEF-25%
Zoo	1.0	0.83	0.89	1.0	0.83	0.83	0.75	0.63
Dermatology	0.94	0.82	0.77	0.78	0.79	0.77	0.73	0.73
Promoters	0.94	0.87	0.84	0.8	0.92	0.86	0.89	0.84
Splice	0.94	0.97	0.8	0.97	0.94	0.97	0.91	0.92
M-feat-fact	0.82	0.82	0.68	0.65	0.84	0.74	0.7	0.71
Arrhythmia	0.78	0.69	0.71	0.61	0.7	0.62	0.57	0.58
Colon	0.63	0.72	0.54	0.56	0.66	0.75	0.65	0.7
SRBCT	0.67	0.72	0.61	0.63	0.74	0.72	0.62	0.56
Leukaemia	0.63	0.72	0.47	0.55	0.62	0.69	0.49	0.54
Ovarian	0.69	0.85	0.52	0.6	0.67	0.79	0.55	0.62
Average	0.804	0.801	0.683	0.715	0.771	0.774	0.686	0.683

Table B.6: The average test accuracy of NB classifiers trained with the features selected by FWHEF with different $\boldsymbol{\beta}$ *and* $\boldsymbol{\lambda}$ using75%, 50% and 25% of these features being selected

NB ACC	-75% $\boldsymbol{\beta}$=0.275	-50% $\boldsymbol{\beta}$=0.275	-25% $\boldsymbol{\beta}$=0.275	-75% $\boldsymbol{\beta}$=0.35	-50% $\boldsymbol{\beta}$=0.35	-25% $\boldsymbol{\beta}$=0.35	-75% $\boldsymbol{\beta}$=0.425	-50% $\boldsymbol{\beta}$=0.425	-25% $\boldsymbol{\beta}$=0.425
Zoo	94.04	91.19	85.45	93.15	91.3	85.35	93.45	91.96	85.65
Dermatology	98.41	95.33	84.69	98.22	97.32	86.14	97.84	97.84	86.3
Promoters	92.88	93.82	84.74	92.79	93.15	85.84	92.69	92.33	85.75
Splice	95.85	95.39	93.74	95.72	94.9	93.74	95.74	94.62	93.73
M-feat-fact	92.53	91.31	90.23	93.04	92.89	91.37	93.19	93.84	92.98
Arrhythmia	67.3	67.94	66.73	67.30	68.7	68.14	67.3	68.63	68.45
Colon	84.69	84.33	85.5	84.69	83.69	82.83	84.69	83.05	82.36
SRBCT	99.03	98.58	96.5	99.03	97.95	96.4	99.03	97.35	95.9
Leukaemia	96.35	96.23	95.66	96.35	96.23	95.8	96.35	96.23	96.07
Ovarian	98.61	99.29	98.85	98.61	99.48	99.13	**98.61**	**99.57**	**99.17**
Average	91.969	91.341	88.209	91.89	91.561	88.474	91.889	91.542	88.636

Table B.7: The stability measures of ATI with the features selected by FWHEF approaches with different $\boldsymbol{\beta}$ *and* $\boldsymbol{\lambda}$ using75%, 50% and 25% of these features being selected.

ATI	-75% $\boldsymbol{\beta}$=0.275	-50% $\boldsymbol{\beta}$=0.275	-25% $\boldsymbol{\beta}$=0.275	-75% $\boldsymbol{\beta}$=0.35	-50% $\boldsymbol{\beta}$=0.35	-25% $\boldsymbol{\beta}$=0.35	-75% $\boldsymbol{\beta}$=0.425	-50% $\boldsymbol{\beta}$=0.425	-25% $\boldsymbol{\beta}$=0.425
Zoo	0.88	0.95	0.99	0.94	0.87	1.0	0.93	0.91	0.76
Dermatology	0.87	0.8	0.78	0.88	0.76	0.72	0.94	0.88	0.75
Promoters	0.84	0.88	0.83	0.75	0.73	0.72	0.71	0.75	0.69
Splice	0.78	0.9	0.97	0.76	0.73	0.9	0.8	0.76	0.7
M-feat-fact	0.8	0.77	0.74	0.72	0.58	0.5	0.82	0.73	0.63
Arrhythmia	0.79	0.71	0.66	0.68	0.57	0.47	0.71	0.68	0.57
Colon	0.63	0.58	0.66	0.47	0.39	0.42	0.49	0.47	0.37
SRBCT	0.79	0.7	0.65	0.57	0.53	0.45	0.79	0.59	0.59
Leukaemia	0.59	0.51	0.61	0.44	0.3	0.3	0.59	0.45	0.47
Ovarian	0.63	0.58	0.7	0.46	0.37	0.45	0.63	0.5	0.57
Average	0.76	0.738	0.759	0.667	0.583	0.593	0.741	0.672	0.61

Table B.8: The stability measures of CWrel with the features selected by FWHEF approaches with different $\boldsymbol{\beta}$ *and* $\boldsymbol{\lambda}$ using75%, 50% and 25% of these features being selected.

CWrel	-75% $\boldsymbol{\beta}$=0.275	-50% $\boldsymbol{\beta}$=0.275	-25% $\boldsymbol{\beta}$=0.275	-75% $\boldsymbol{\beta}$=0.35	-50% $\boldsymbol{\beta}$=0.35	-25% $\boldsymbol{\beta}$=0.35	-75% $\boldsymbol{\beta}$=0.425	-50% $\boldsymbol{\beta}$=0.425	-25% $\boldsymbol{\beta}$=0.425
Zoo	0.89	0.92	0.99	0.94	0.89	1.0	0.91	0.77	0.98
Dermatology	0.9	0.78	0.73	0.84	0.77	0.78	0.85	0.76	0.73
Promoters	0.75	0.78	0.76	0.84	0.84	0.8	0.84	0.81	0.79
Splice	0.74	0.85	0.9	0.80	0.8	0.97	0.8	0.77	0.97
M-feat-fact	0.74	0.67	0.6	0.78	0.68	0.65	0.79	0.73	0.65
Arrhythmia	0.68	0.56	0.51	0.79	0.71	0.61	0.79	0.71	0.6
Colon	0.47	0.43	0.51	0.63	0.54	0.56	0.63	0.52	0.52
SRBCT	0.57	0.54	0.4	0.79	0.61	0.63	0.57	0.5	0.44
Leukaemia	0.44	0.33	0.32	0.59	0.47	0.55	0.44	0.3	0.28
Ovarian	0.46	0.42	0.55	0.63	0.52	0.6	0.46	0.34	0.42
Average	0.664	0.628	0.627	0.763	0.683	0.715	0.708	0.621	0.638

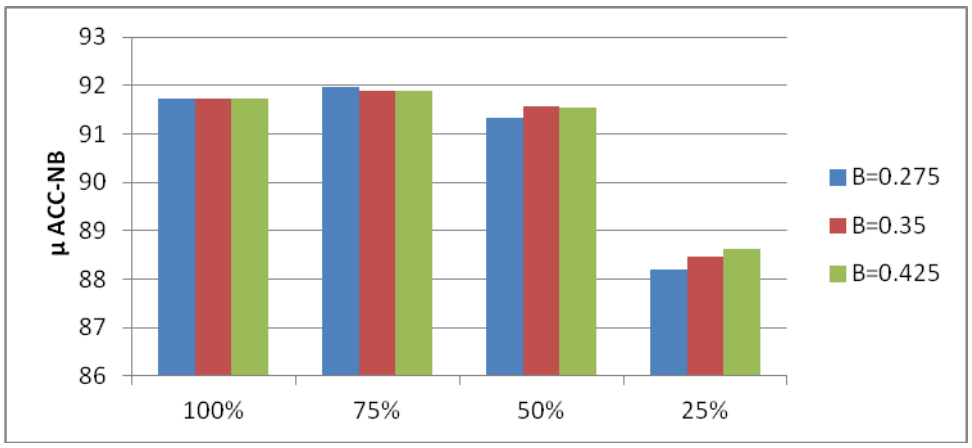


Figure B.1: The average test accuracy of NB by using FWHEF approach focusing on different value of β and λ

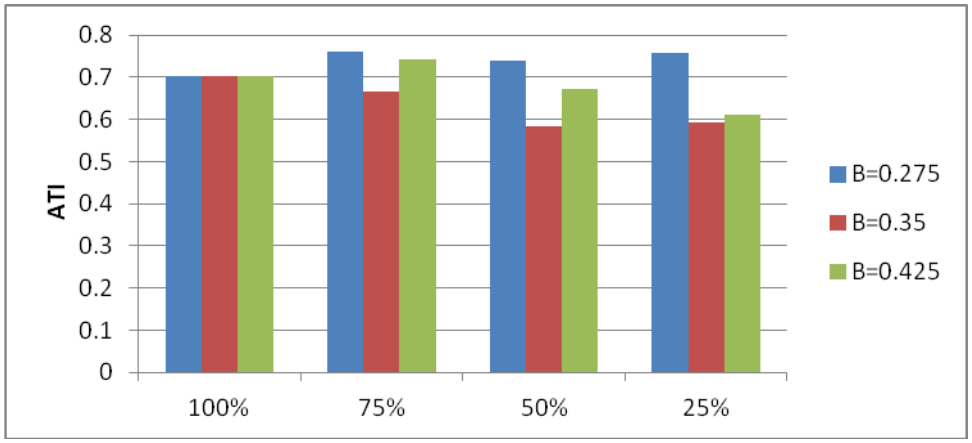


Figure B.2: The average ATI by using FWHEF approach focusing on different value of β and λ.

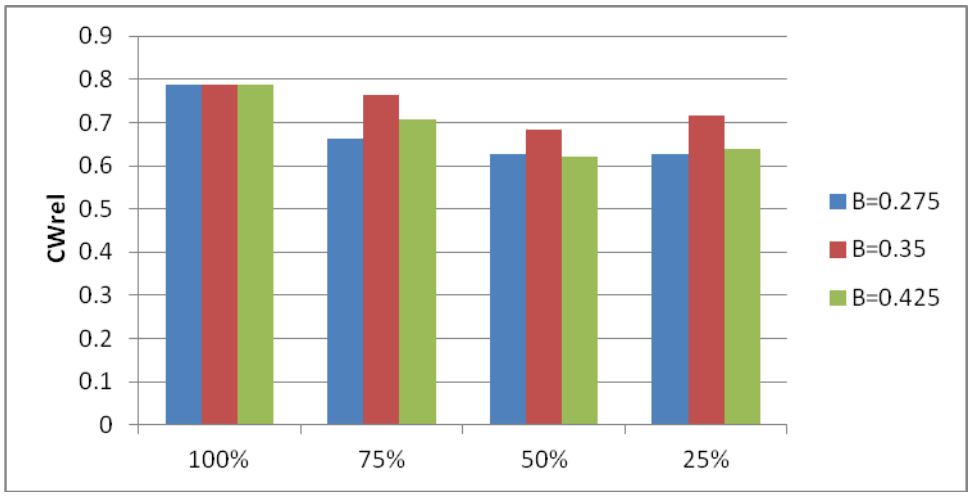


Figure B.3: The average CWrel by using FWHEF approach focusing on different value of β and λ.

Appendix C: Further results from Chapter 8

Table C.1: Comparison of HEFb-75% with other EFS studies. Values given are average accuracy; parentheses show the number of features selected, and the last column presents the methods of other FS studies (FS + Classifier + Evaluation Scheme).

Data	Our Results (HEF)	Best results report in the Literature	
		Results	**Methods (FS + Classifier + Evaluation Scheme)**
Zoo	NB 94.93(8.25)	93(6) 93(6)	DFL+NB+LOOCV, (Zhang and Zhang, 2011) CSE+NB+LOOCV+, (Zhang and Zhang, 2011)
	KNN 95.93(8.25)	90(NA) 94(NA) 94(NA) 90.5(NA)	Hill Climbing+KNN+10 Fold CV, (Loughrey and Cunningham, 2005a) Forward selection+KNN+10 Fold CV, (Loughrey and Cunningham, 2005a) Backward elimination +KNN+10 Fold CV, (Loughrey and Cunningham, 2005a) Genetic Algorithm +KNN+10 Fold CV, (Loughrey and Cunningham, 2005a)
	SVM 96.83(8.25)	94(6) 94(6)	DFL+SVM+LOOCV, (Zhang and Zhang, 2011) CSE+SVM+LOOCV, (Zhang and Zhang, 2011)
Dermatology	NB 98.14(21) KNN 96.14 (21) SVM 97.57 (21)	79.25(18) 93.74(10)	MIFS+ Average (NB+KNN+C4.5)+10 Fold CV , (Liu et al., 2008) CIMI+ Average (NB+KNN+C4.5)+10 Fold CV , (Liu et al., 2008)
Promoters	NB 92.64(8) KNN 88.44 (8) SVM 91.54 (8)	87.89(5) 87.89(5)	MIFS+ Average (NB+KNN+C4.5)+10 Fold CV , (Liu et al., 2008) CIMI+ Average (NB+KNN+C4.5)+10 Fold CV , (Liu et al., 2008)
Splice	NB 96.18 (20) KNN 81.82 (20) SVM 95.79 (20)	91.72(9) 91.72(9)	MIFS+ Average (NB+KNN+C4.5)+10 Fold CV , (Liu et al., 2008) CIMI+ Average (NB+KNN+C4.5)+10 Fold CV , (Liu et al., 2008) (ALL)
M-feat-factors	NB 92.7(72)	91.5(50) 88.4(50) 81.34 (6) 78.78(6)	HFSDD+NB+10 Fold CV, (Liang et al., 2009) mrmrMID +NB+10 Fold CV, (Liang et al., 2009) MIFS+ Average (NB+KNN+C4.5)+10 Fold CV , (Liu et al., 2008) CIMI+ Average (NB+KNN+C4.5)+10 Fold CV , (Liu et al., 2008)
	KNN 96.08(72)	94.6(50)	mrmrMID +KNN+10 Fold CV, (Liang et al., 2009)
	SVM 97.5(72)	96.4(50) 92.5(50)	HFSDD+SVM+10 Fold CV, (Liang et al., 2009) mrmrMID +SVM+10 Fold CV, (Liang et al., 2009)

Arrhythmia	NB 66.66(42) KNN 56.82(42) SVM 67.48(42)	68.84(274) 59.95(110)	Filter,DDC (decision dependent correlation) + Average (NB+KNN+C4.5)+10 Fold CV , (Liu et al., 2008) Filter,CR (conditional variable relevance)+ Average (NB+KNN+C4.5)+10 Fold CV , (Liu et al., 2008)
Colon	NB 84.24(35.25)	79.3 (100) 84.3 (91) 85.5 (46) 77(14) 80.65 (3.8) 83.87 (2.8) 77.14(4.6)	BAHSIC+NB+10 Fold CV, (Schowe, 2011) DRAGS+NB+10 Fold CV, (Schowe, 2011) CGS+NB+10 Fold CV, (Schowe, 2011) INTERACT+NB+5 Fold CV , (Bolón-Canedo et al., 2014) IWSS +NB+10 Fold CV, (Bermejo et al., 2009) IWSSr +NB+10 Fold CV, (Bermejo et al., 2009) FOCUS +NB+10 Fold CV, (Ruiz et al., 2006)
	KNN 78.52(35.25)	75.81(4.6) 77.42(4.9)	IWSS +KNN+10 Fold CV, (Bermejo et al., 2009) IWSSr +KNN+10 Fold CV, (Bermejo et al., 2009)
	SVM 86.83(35.25)	81(14) 80.65(30)	INTERACT+SVM+5 Fold CV , (Bolón-Canedo et al., 2014) SVM-RFE + SVM + 10 Fold CV,(Kalousis et al., 2007)
SRBCT	NB 99.04 (132)	67(150) 96.7(100) 87.4 (250) 90.2(210)	Information gain +NB +4 Fold CV , (Li et al., 2004) BAHSIC+NB+10 Fold CV, (Schowe, 2011) DRAGS+NB+10 Fold CV, (Schowe, 2011) CGS+NB+10 Fold CV, (Schowe, 2011)
	KNN 99.4 (132)	91(150)	Information gain +KNN +4 Fold CV , (Li et al., 2004)
	SVM 99.89(132)	98.53(90.5) 98.9(80) 78.3 (14) 95 (150) 99.43(110)	WFFSA+ SVM+ 10 Fold CV, (Zhu et al., 2007) Degree of differential prioritization (DDP)+DAGSVM+F-splits, (Ooi et al., 2006) NMICFS-PSO +SVM+ LOOCV, (Xu et al., 2013) Information gain +SVM +4 Fold CV , (Li et al., 2004) MBE-MOMA+SVM+632 bootstrap, (Zhu et al., 2010)
Leukaemia	NB 96.21(96.75)	87.5(2.5) 87.5(2) 93.04(2.5) 84.82(2.4)	IWSS +NB+10 Fold CV, (Bermejo et al., 2009) IWSSr +NB+10 Fold CV, (Bermejo et al., 2009) BIRS +NB+10 Fold CV, (Ruiz et al., 2006) FOCUS +NB+10 Fold CV, (Ruiz et al., 2006)
	KNN 95.57 (96.75)	88.89(2.8) 87.5(2.2) 94.2 (40)	IWSS +KNN+10 Fold CV, (Bermejo et al., 2009) IWSSr +KNN+10 Fold CV, (Bermejo et al., 2009) OR+KNN+10 Fold CV, (Cannas et al., 2013)
	SVM 96.55(96.75)		
Ovarian	NB 98.34(60)		
	KNN 99.48(60)		
	SVM 100(60)	99.84 (**2**) 99.84 (4) 100(32)	DFL+SVM+ Training & test set, (Zhang and Zhang, 2011) CSE+SVM+ Training & test set, (Zhang and Zhang, 2011) INTERACT+SVM+5 Fold CV , (Bolón-Canedo et al., 2014)