FEATURE SELECTION FOR THE CLASSIFICATION OF LONGITUDINAL HUMAN AGEING DATA

Tossapol Pomsuwan

School of Computing

University of Kent

This dissertation is submitted for the degree of Master by Research

October 2017

Total word count: 35468

DECLARATION

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration, except where specifically indicated in the text. It has not been previously submitted, in part or whole, to any university of institution for any degree, diploma, or other qualification.

Signed:____

Date:____

Tossapol Pomsuwan

ABSTRACT

We address the feature selection task in the special context of longitudinal data – where variables are repeatedly measured across different time points. When analysing longitudinal data, a standard feature selection method would typically ignore the temporal nature of the features and treat each feature value at a given time point as a separate feature. That is, a standard algorithm would ignore the important difference between values of the same feature (measuring the same property of an instance) across different time points and values of fundamentally different features (measuring different properties of an instance) at the same time point.

This thesis presents two main contributions. The first one is the creation of the longitudinal datasets used in the experiments, including the construction of features capturing longitudinal information for predicting age-related diseases. The datasets were created from data in the English Longitudinal Study of Ageing (ELSA) database. The second contribution consists of proposing four new variants of the Correlation-based Feature Selection (CFS) method for selecting features to be used as input by a classification algorithm. These CFS variants take into account (in different ways) the temporal redundancy associated with variations in the value of a feature across different time points.

The results are summarised from two main perspectives. Firstly, in terms of predictive accuracy, one of the proposed CFS variants (called Exh-CFS-Gr – exhaustive search-based CFS per group of temporally redundant features) obtained a statistically significantly better predictive performance than the performance obtained by standard CFS and the baseline approach of no feature selection when using Naïve Bayes as the classification algorithm. However, there was no statistically significant difference between the predictive accuracies obtained by J48, a decision tree induction algorithm, for all different variants of CFS (including standard CFS). Secondly, regarding the feature subsets selected by different variants of CFS, the number of features selected by Exh-CFS-Gr was substantially greater than that of all other three CFS variants for all datasets. This helps explaining why this feature selection method obtained the best results in the experiments with Naïve Bayes; i.e., it seems that the other CFS variants selected relatively too few features for Naïve Bayes. Additionally, the features originally observed in the ELSA database were, in general, selected more often (by all variants of CFS) than the constructed features capturing longitudinal information.

ACKNOWLEDGEMENTS

I would like to show my appreciation to my supervisor Professor Alex Freitas for the constant support I received throughout my study. Because of his enthusiasm and patience, I was able to engage in my thesis with ease. He always provided me with great advice to improve my research performance and, especially, writing, which helped me to improve the skills necessary for my future PhD.

In addition, I wish to thank Dr Marek Grzes, the seminar organiser, for holding the weekly seminar series for the Computational Intelligence research group. The presentations from both external speakers and my colleagues were helpful and contributed to my thesis.

Furthermore, I would also like to thank all the lab members in the Computational Intelligence group for all the advice and encouragement. Not only did they welcome me with open arms, but they also gave me good suggestions for my thesis as well as their mental support. They made my experience as a Master by Research student less stressful than I thought it would be.

Besides, I must express my gratitude to my financial supporter, The Royal Thai Government, who was responsible for my tuition fee and living expenses throughout my study.

Lastly, I would like to thank my parents and friends who always supported, comforted and motivated me during all the ups and downs.

CONTENTS

LIST OF TABLES AND FIGURES

1 INTRODUCTION

According to a United Nation’s report on World Population Ageing (2015) (United Nations, Department of Economic and Social Affairs and Population Division 2015), the number of people worldwide aged 60 or over is estimated to reach 2.1 billion by the year 2050. The process of getting older eventually leads to a decline in the physical and mental health of people. Also, as people get older they become increasingly more likely to develop life-threatening age-related diseases.

In the last few years, developments in bioinformatics methods for data collection have increased the need for computational methods to organise and analyse very large and continuously growing amounts of biological data, such as human ageing data. The main goal of bioinformatics is to apply existing algorithms, or developing new ones, to discover and evaluate several relationships between biological entities (Baldi and Brunak 2001; Altman 2001). Previous studies report that machine learning, which consists of a collection of automatic and intelligent learning techniques, is able to evaluate biological data (Bhaskar, Hoyle and Singh 2006). The biology of ageing is one of the most interesting topics in our era, yet it is challenging in terms of understanding (Comfort 1964; Hofer and Sliwinski 2001; Lexell, Taylor and Sjöström 1988; Adam 2001). Whilst a substantial amount of data on ageing is available, discovering interesting knowledge from such data is not trivial, due to the complexity of the biological process of ageing.

In this thesis, our primary goal is not to analyse data directly about human longevity, i.e. we do not try to predict the longevity (lifespan) of individuals. Rather, our primary goal is essentially to analyse data about age-related diseases, i.e., we try to predict whether an individual will develop some age-related disease in the future, based on past biomedical data about that individual. In order to achieve this goal, we employed machine learning techniques to identify abnormal behaviours and predict life-threatening diseases in older adults, such as heart attack, stroke, high blood pressure, dementia, etc. These harmful diseases are also known as age-related diseases for which old age is one of the greatest risk factors.

Therefore, this research was constructed to study biomedical data about human ageing, where the data sets were derived from the English Longitudinal Study of Ageing (ELSA) (Marmot *et al.* 2016). The ELSA study is a longitudinal survey of ageing and quality of life among older people that explores the dynamic relationships between health and functioning, social networks and participation, and economic position as people plan for, move into and progress beyond retirement. In this thesis, however, we focus only on biomedical data, such as the results of blood tests and other data collected by nurses, and the relationship between that data and the age-related diseases of patients, as will be described in more detail later.

In machine learning, a classification algorithm aims to find a predictive relationship between features and the class variable. This is done by building a classification model from pre-classified instances. Afterwards, this model is used to predict the class label of previously unseen instances.

In classification datasets with a large number of features, feature selection methods are often applied in a data pre-processing step (Li *et al.* 2016; Liu 1998; Wang, Wang and Chang 2016) in order to remove irrelevant or redundant features. This can lead to higher predictive accuracy and reduce the training time of classification algorithms.

The vast majority of works on the classification task focus on analysing the standard type of classification data, where each variable is measured at a single time point, so that there is no explicit temporal structure in the data. However, many important data sources – particularly in the biomedical domain – contain longitudinal data, where the values of a variable are repeatedly measured across several time points (often called waves) (Ribeiro *et al.* 2017). For instance, many hospital databases contain records with blood test results measured for the same patient across many time points.

In this thesis, we address the feature selection task in the special context of longitudinal data. When analysing longitudinal data, a standard feature selection method would typically ignore the temporal nature of the features and treat each feature value at a given time point as a separate feature. That is, a standard algorithm would ignore the important difference between values of the same feature (measuring the same property of an instance) across different time points and values of fundamentally different features (measuring different properties of an instance) at the same time point.

This thesis presents two main contributions. The first one is the creation of the longitudinal datasets used in the experiments, including the construction of features capturing longitudinal information for predicting age-related diseases. The datasets were created from data in the English Longitudinal Study of Ageing (ELSA) database (Marmot *et al.* 2016). The created datasets contain two types of features, namely originally observed features (directly taken from the ELSA database) and the aforementioned constructed longitudinal features, where both feature types occur in three waves. Besides, each dataset contains a single class variable representing an age-related disease, so that multiple datasets were created for different diseases. In each dataset, the classification task consists of predicting whether or not an individual will develop a given age-related disease in a later wave of the longitudinal data in ELSA, based on values of biomedical features describing characteristics of the individual in previous waves.

The second contribution consists of proposing four new variants of the Correlation-based Feature Selection (CFS) method. CFS is used in a data pre-processing phase for selecting features to be used as input by a conventional classification algorithm. These CFS variants take into account (in different ways) the temporal redundancy associated with variations in the value of a given feature across different waves (time points). In essence, the four proposed CFS variants can be categorised into two types of modifications of the standard CFS method, namely two of the variants modify the standard CFS' search method; whilst the other two variants modify the standard CFS' evaluation function.

The remainder of the thesis is organised as follows. Chapter 2 presents the background for this research. Chapter 3 presents the main contributions, namely dataset creation (including the creation of longitudinal features capturing temporal information) and four new variants of CFS. Chapter 4 reports the computational results. Chapter 5 presents the conclusions and future research directions.

2 BACKGROUND

In this chapter, we review the background on feature selection and classification algorithms relevant to the thesis. This chapter is organised as follows. Section 2.1 introduces concepts and methods for the classification task, focusing on decision tree and Naïve Bayes algorithms, which are the types of classification algorithms used in the experiments reported in Chapter 4. Section 2.2 discusses feature selection methods, including general approaches and types of search methods. This section also describes in detail the Correlation-based Feature Selection method, since this thesis proposes variants of this method in Chapter 3. Section 2.3 explains the basic concepts of the longitudinal classification task. Section 2.4 presents related work on feature selection for longitudinal classification.

2.1 Concepts and Methods for the Classification Task

A data set is a collection of instances (records). Each instance consists of two parts, a set of predictive features and a class. This means that every instance belongs to a predefined class. A classification algorithm performs the process of building a classification model which allows us to predict the class of an instance, given the values of its predictive features. This model is built by using a training set, where the classes of the instances are known. After that, the model is used to predict the classes of instances in the test set, where the classes are unknown. To summarize, the classification process involves the induction of a classification model from the training set and its application to predict the class of instances in the test set (unseen during training).

2.1.1 Decision Tree Induction Algorithms

Decision tree induction algorithms typically build a decision tree in a top-down fashion (Quinlan 1993), by using a recursive learning process, as follows. Firstly, the algorithm considers all training instances, which are allocated to the root node of the decision tree. Then, a feature (*f*), which best separates the classes based on a given feature selection criterion, is selected to label the current (root) node. Next, the set of instances (*I*) in the current node is partitioned into *s* mutually exclusive subsets of instances (I_1, …, I_k) according to the values of the selected feature *f*; where *k*, the number of instance subsets, is determined based on the type of selected feature *f*, as follows. If feature *f* is nominal (categorical), *k* is typically the number of values taken by the feature, so that an instance subset is created for each of the feature values. If feature *f* is numerical (continuous), typically *k* is set to 2, so that the algorithm creates two instance subsets, one with the instances satisfying the condition $f \leq thr$ and the other with the instances satisfying the condition $f > thr$, where *thr* is a threshold automatically determined to maximise class separation among the two instance subsets.

In any case, each instance subset (I_1,…, I_s) is allocated to a newly created child node, where the processes of feature selection and partitioning the current set of instances into subsets are recursively repeated. If all instances in a newly created child node (instance subset) belong to a single class, there is no need to keep partitioning the instances in that node, then the algorithm converts the current node into a leaf and stops the recursive process in this part of the decision tree. The current node can also be converted into a leaf for other reasons, even if it still contains instances of different classes – see the discussion on pre-pruning later in this section.

This process of attribute selection and instance set partitioning is recursively repeated for each non-leaf node of the tree, until all those nodes are converted into leaf nodes. The goal of this process is to allocate instances of different classes to different subsets of instances.

One of the well-known decision tree induction algorithms is the J48 algorithm available on WEKA data mining tool (Hall *et al.* 2009), which is an implementation of the C4.5 algorithm (Quinlan 1993). J48 was used as one of the classification algorithms in this research, as will be discussed later.

There are several feature selection criteria that can be used in order to select the most relevant feature to separate the classes at each node. One of the most commonly used criteria is the Information Gain (IG), which is used by several decision tree induction algorithms, such as Interactive Dichotomizer 3 (ID3) (Quinlan 1986). In information theory, Shannon's entropy function (also known as entropy) is a method to measure uncertainty in the outcome of an experiment. For example, consider a random variable X with v possible values. The Shannon entropy function, denoted *H(X),* is defined in equation (2.1):

$$H(X) = -\sum p_i \, log_2 \, p_i \tag{2.1}$$

where p_i is the probability that X takes its i-th value, i = 1,…,v. Let $H(X|Y = y_i)$ be the entropy of the variable X conditioned on the variable Y taking a certain value y_i. Then *H(X|Y)* is the weighted average of $H(X|Y = y_i)$ over all possible values y_i that Y may take, where the weights are the probabilities of the y_i values. This is the entropy of X given Y, defined in equation (2.2), also known as conditional entropy.

$$H(X|Y) = \sum_{i=1}^{v} p(Y = y_i)H(X|Y = y_i) \tag{2.2}$$

The IG criterion is based on the following concept: "Entropy represents the amount of uncertainty in the outcome of an experiment, so we want to minimize entropy when selecting a feature in a decision tree" (Liu 1998). The IG is defined in equation (2.3):

$$IG(I,f) = H(I) - H(I|f) \tag{2.3}$$

where *H(I)* denotes the amount of information contained in the set of instances *I* and *H(I|f)* is the reduced amount of information (reduced entropy) after using feature *f* to partition the set *I*. In addition, Symmetrical Uncertainty (SU) (Hall 2000) compensates for information gain's bias toward attributes with more values and normalises its value to the range [0; 1]. This is defined as shown in equation (2.4):

$$SU(X,Y) = 2.0 \times \left[\frac{I\mathrm{G}(X,Y)}{H(Y) + H(X)}\right] \tag{2.4}$$

However, many feature selection criteria can be used instead of the IG. The issue of which feature selection criterion is the best depends mainly on the data set being mined. For example, the drawback of using information gain is a bias favouring the choice of features with a lot of values. In an extreme case where a feature has a distinct nominal value for each instance, e.g., a patient's ID number, partitioning the set of instances according to the values of this feature would result in a "perfect" partition ($H(I|f_{ID}) = 0$). However, this is an extreme case of overfitting the decision tree to the training data, with no generalisation to new test set. Therefore, to reduce this bias of the information gain criterion, Information Gain Ratio (IGR), which is a modification of the information gain (Quinlan 1993), was invented. In terms of implementation, the IGR is used by C4.5, which is an improved version of ID3, and J48 algorithm (an alternative implementation of C4.5) available on WEKA data mining tool (Hall *et al.* 2009). The IGR is defined in equation (2.5):

$$IGR(I,f) = \frac{IG(I,f)}{PE(I,f)} \tag{2.5}$$

where *PE(I,f)* measures entropy produced partitioning the set of instances in the current tree node into *K* partitions, where *K* is the number of values of nominal feature *f*. Hence, IGR overcomes the drawback of IG criterion. Nevertheless, there is also a potential problem of using IGR since it may overcompensate. This means that IGR may choose a feature just because the term *PE(I,f)* is very low. A standard approach to fix this problem is to select the feature *f* with the highest value of *IG(I,f)* subject to the restriction that *f*'s *IG(I,f)* value must be equal to or greater than the average value of *IG(I,f)* for all features being considered.

Decision Tree Pruning is performed in order to remove irrelevant nodes constructed from the training set due to noise or outliers. As a result, the pruned trees are smaller and less complex to be interpreted. In general, there are two main tree pruning approaches, namely pre-pruning and post-pruning (Frank 2000). The former prunes the tree by halting its construction early. As an example of a pre-pruning criterion, the current node can be converted into a leaf node if the number of instances in the current node is smaller than a certain pre-defined threshold (since in this case there would be little statistical support for calculating the IG value of attributes at that node). On the other hand, post-pruning removes a sub-tree from a fully-grown tree. Taking Reduced Error Pruning (Elomaa and Kääriäinen 2001) as an example of post-pruning approaches, such method works by tentatively replacing subtree rooted at a given internal (non-leaf) node within the tree with a leaf, assigning all instances in that newly created leaf to the most frequent class among those instances. If the replacement of this subtree with a leaf does not increase the classification error of the tree, then the subtree is permanently removed, i.e. transformed into a leaf node. Doing so requires a validation set which can be obtained by holding out a part of the training set. After that, the classification error can, for instance, be derived according to the number of misclassified instances from the validation set. The process is repeated to iterate over all tree nodes until the pruning is no longer helpful.

There are a number of motivations for decision tree pruning. The first one is simplifying the model (the tree). The second one is reducing the risk of "overfitting", which occurs when the unpruned decision tree fits to details of the training set that do not generalise well to the test set. On the other hand, if pruning is too aggressive, this could lead to "underfitting" of the model to the data.

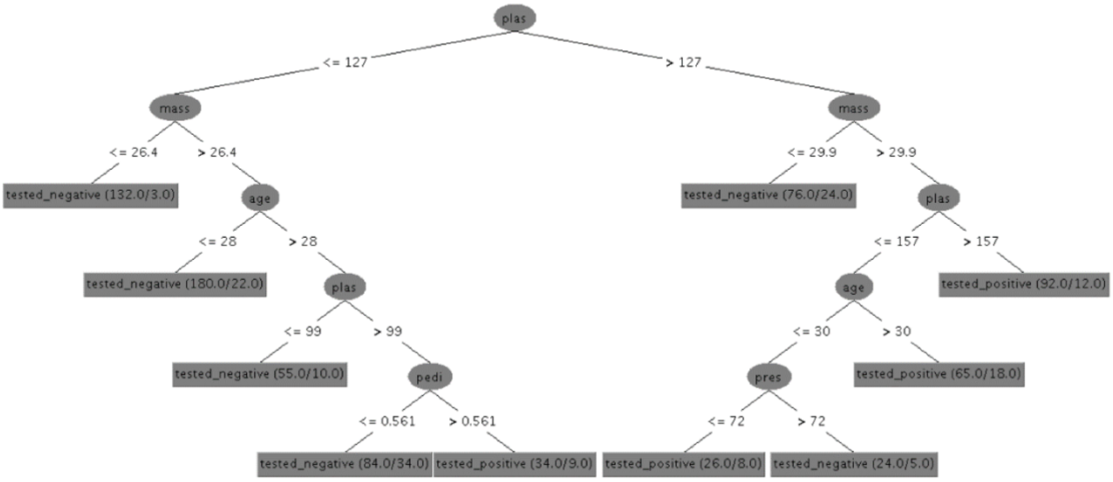
plas
<= 127
> 127
mass
<= 26.4
> 26.4
tested_negative (132.0/3.0)
age
<= 28
> 28
tested_negative (180.0/22.0)
plas
<= 99
> 99
tested_negative (55.0/10.0)
pedi
<= 0.561
> 0.561
tested_negative (84.0/34.0)
tested_positive (34.0/9.0)
mass
<= 29.9
> 29.9
tested_negative (76.0/24.0)
plas
<= 157
> 157
tested_positive (92.0/12.0)
age
<= 30
> 30
tested_positive (65.0/18.0)
pres
<= 72
> 72
tested_positive (26.0/8.0)
tested_negative (24.0/5.0)

Figure 2.1: an example of a decision tree

There are a number of advantages of decision tree induction algorithms. The main one is comprehensibility (Freitas 2013), since a decision tree is relatively easy to interpret (as long as the tree is not too large). In particular, a decision tree is represented in a graphical form (a diagram), which is a very user-friendly representation, as shown in Figure 2.1. The example decision tree in this figure was built, by running J48 algorithm in WEKA (Hall *et al.* 2009), from the Pima Indians Diabetes dataset (from the UCI dataset repository (Bache and Lichman 2013), where the class variable indicates whether or not the patient shows signs of diabetes and the features describe their general health information such as age, body mass index, blood pressure, etc.

Moreover, decision trees have a hierarchical structure that also facilitates their interpretation: in general, the closer a feature is to the root, the more relevant it is. In other words, the most relevant features for predicting an instance's class are automatically placed near to the root of the tree, which enables us to easily identify important features and helps in analysing the underlying predictive relationships between relevant features and the class (Freitas 2013). Taking Figure 2.1 as an example, this decision tree illustrates that the feature "plas" (Plasma glucose concentration) plays the most important role in the test of predicting whether or not a patient has diabetes. By contrast, "black box" classification algorithms such as Support Vector Machines (SVMs) are likely to achieve higher predictive accuracies, but they have the considerable drawback of producing non-interpretable classification models. The interpretability of the classification model is an important issue, especially when our aim is to discover new comprehensible knowledge (Cristianini and Shawe-Taylor 2000). Furthermore, a decision tree automatically reports feature interactions involving the features selected along each path from the root to a leaf.

Thirdly, decision tree algorithms can effectively cope with both numerical and nominal attributes, whilst some classification algorithms like instance-based learning (Aha, Kibler and Albert 1991) and Support Vector Machines (Cristianini and Shawe-Taylor 2000) do not cope so naturally with nominal attributes – e.g. such attributes would normally be converted into numerical attributes when using SVM, introducing an arbitrary numerical order among originally unordered nominal values.

Fourthly, decision tree algorithms' effectiveness in terms of predictive accuracy is in general acceptable, and especially high in some datasets where the values of the class variable are assigned to instances by analysing one-feature-at-a-time. For instance, in some credit datasets the class values assigned to the customers seem to be generated by a human analyst who manually chooses a sequence of relevant features for classifying the customer's credit risk, and this kind of sequential classification matches well the approach of identifying relevant features one-at-a-time when building a decision tree (Brazdil and Henery 1994). However, it should be noted that decision tree algorithms are not considered the state-of-the-art in terms of predictive accuracy in general, and more modern decision tree-based algorithms like random forests (Breiman 2001; Touw *et al.* 2013) tend to obtain higher predictive accuracies in general, using the power of an ensemble of decision trees to make more robust predictions. On the other hand, the fact that random forests use an ensemble of (with a large number of) decision trees makes the model much harder to interpret than a single decision tree.

Last but not least, the computational time spent on building a decision tree classifier is relatively fast, since decision-tree induction algorithms use the principle of "divide-and-conquer". In general, the time complexity of building a decision tree is $O(m \cdot n^2)$, where m is the number of instances in the training set and n is the number of predictor features. In addition, the time complexity of the classification of testing instances (unseen during training) is $O(t \cdot log\ s)$, where t is the number of instances in the testing set and s is the size of the decision tree (number of nodes, including leaves) (Su and Zhang 2006).

Despite these advantages, one major drawback of decision tree induction algorithms is the fragmentation problem (Freitas 2013; Rokach 2016). Specifically, the use of the divide-and-conquer principle means that at the deeper levels of a decision tree, the feature selection procedure uses fewer and fewer instances from the training set, so the feature selection process is less statistically reliable at deeper nodes of the tree. This usually leads to the generation of many locally important yet globally insignificant rules, which tends to decrease the predictive accuracy of the decision tree.

Another drawback of the large majority of decision tree algorithms is that they select features considering just one feature at a time, a relatively simple approach which does not cope well with strong interactions between attributes (e.g. when a good class separation requires a linear combination of numerical attributes, rather than a single attribute).

Therefore, the predictive accuracy of decision tree classification models is often inferior to other types of classification models such as neural networks and support vector machines (SVMs), as well as often inferior to random forests as mentioned earlier.

2.1.2 Naïve Bayes Algorithm

Naïve Bayes is a classification algorithm based on Bayes' theorem in the area of probabilities (Sulzmann, Fürnkranz and Hüllermeier 2007). Naïve Bayes models are simple and fast to build. The basic rationale for Naïve Bayes can be explained as follows. In order to classify an unforeseen instance, we look into a data set of instances whose classes are known. In the ideal case, suppose that there are a huge number of instances (records), so we have sufficient examples for each possible combination of values for all predictive features. Hence, a new instance can now be classified by choosing the most frequent class for the particular combination of feature values occurring in that instance. In practice, however, we are unlikely to have many instances for every possible combination of feature values. Therefore, Bayes' Theorem is employed to build Naïve Bayes classification models based on probabilities computed from the training set.

Specifically, the probability of an instance - or example – (*E_j*) having a class label (*C_j*) is computed from a training set as given in equation (2.6).

$$P(C_i|E_j) = \frac{P(E_j|C_i) \times P(C_i)}{P(E_j)} \tag{2.6}$$

If we look at equation (2.6) carefully, in order to classify a given instance E_j, the probability of the instance (i.e. the probability of the particular combination of feature values observed in the instance), denoted $P(E_j)$, is fixed for all classes, so we just have to choose the class C_i with maximum value of $P(\boldsymbol{E_j}|\boldsymbol{C_i}) * P(\boldsymbol{C_i})$. In addition, $P(C_i)$, the prior probability of class (C_i), is estimated as the relative frequency of C_i in the entire training set. The challenge is to estimate $P(E_j|C_i)$. The most common simplification is to make an assumption of independence among predictive features conditioned on the class. In other words, a Naïve Bayes classification model assumes that the presence of a particular feature is unrelated to the presence of any other feature, given the class. Hence, with this assumption, $P(E_j|C_i)$ can be estimated as shown in equation (2.7):

$$P(E_j|C_i) \ = \ P(f_{(1)}|C_i) \times P(f_{(2)}|C_i) \times \ldots \times P(f_{(m)}|C_i) \tag{2.7}$$

where m is the number of features and $f_{(1)}$, $f_{(2)}$, … , $f_{(m)}$ denote the values of the corresponding features in the instance j. To conclude, the instance Ej is assigned the class label C_i with a maximum value of $P(E_j|C_i) * P(C_i)$.

The most serious disadvantage of the Naïve Bayes algorithm is that it assumes each feature is independent from all other features given the class variable. This is usually an unrealistic assumption and it is often violated in real-world datasets. Another limitation is that the predictive accuracy of the models is particularly sensitive to redundant features. In particular, if two features are highly correlated (perhaps because they measure slightly different aspects of the same property of an instance), this high degree of feature correlation would be ignored by Naïve Bayes, which would count those features as providing two different pieces of evidence for the classification of an instance over-emphasizing the importance of those features.

Lastly, to use Naïve Bayes with continuous (real-valued) features, the probability density of the feature is sometimes approximated according to a given distribution; normally the Gaussian distribution is used (John and Langley 1995). However, using the same distribution for all numerical features, again, is unrealistic, so ideally we need to choose a distribution which best characterises each feature, but this is a complex task that is rarely done in practice. As a reasonable compromise, sometimes numerical features are discretized in a pre-processing step before running the Naïve Bayes algorithm (García *et al.* 2013; Liu *et al.* 2002); but in this case, there is a risk that the discretization method will produce discretized intervals that lose some important information about the data.

In spite of having the negative points mentioned above, the Naïve Bayes algorithm is a simple and powerful technique with several strengths. The first one is that it can handle missing values in a natural way by simply ignoring them. This is because features are used separately by the algorithm at both model construction and classification stages. Accordingly, if an instance has a missing value for a feature, it can be overlooked while constructing the model, and ignored when the probability is calculated for predicting the class label.

In addition, Naïve Bayes can perform well even though the size of training set is small. In particular, it only requires sufficient instances to estimate the probabilistic relationship of each feature in isolation with the class. Given that interactions between features are not taken into account in the classification model, training instances of these interactions are not needed for learning, thus, generally Naïve Bayes requires less instances for effective training than other algorithms, such as logistic regression (Xue and Titterington 2008). Hence, when the size of the training set is not large, the risk of overfitting is not large.

2.1.3 Measuring Predictive Accuracy

In general, there are two main evaluation criteria to evaluate classification models. By far the most used one (which is also used in this thesis) is predictive accuracy, which can be measured by different measures such as precision, recall and F-measure (as explained later in this section).

The second criterion, which is not evaluated in this research, is the comprehensibility (or interpretability) of classification models. Note that a model's comprehensibility is arguably fundamentally subjective, at least much more subjective than the conventional measures of predictive accuracy. In actual fact, some types of classification models do not provide much information about interpretability, such as SVM and Random Forest. In particular, the models built by these algorithms are normally used as "black box" models. On the other hand, decision trees and Naïve Bayes models (the types of models built in this research, as mentioned earlier) are easier to be interpreted, as discussed earlier. It should be noted that, although comprehensibility is a subjective concept, there are a lot of data mining (or machine learning) works that measure a model's comprehensibility (or more precisely a model's simplicity) in an objective way. For instance, the simplicity of a decision tree is often measured by its size, counting the number of parent and leaf nodes. In other words, the smaller the size, the simpler the model is. Such model simplicity measures have, however, the limitation of ignoring the meaning of the variables in the model and their interactions (Freitas 2013).

With regard to measuring predictive accuracy, if a classification algorithm learns from an entire data set, then attempting to maximise the predictive accuracy on the same data set is trivial, since the data has already been memorised – i.e. the algorithm knows the class for each instance in the dataset. Because of this, in order to measure the predictive accuracy, the set of instances in the full data set has to be partitioned into two parts, called the training set and the test set, where the former is used to build the classification model and the latter is used to measure the predictive accuracy. However, measuring predictive accuracy using a single training and test set partition is statistically unreliable due to the potential bias of selecting instances for the training and test tests. Therefore, the K-fold cross-validation technique is often employed to measure predictive accuracy in a way that uses multiple partitions of the data into training and test sets, whilst avoiding overlapping test sets (Hall *et al.* 2009). The K-fold cross-validation procedure starts by randomly splitting the whole data set into K subsets (or folds) of approximately equal size. Afterwards, the classification algorithm is run K times, each time with a different subset used as the test set and the other K – 1 subsets used as the training set. The reported measure of predictive accuracy is the mean of the accuracy over the K test sets. In general, K = 10, as used in this research, is the most popular form of cross-validation, also known as 10-fold cross validation (Kohavi 1995).

Although the K-fold cross-validation technique is a statistically robust approach to measure predictive accuracy, in real-world applications we usually need to report a single classification model for the user. However, none of the K models is considered to be the best model that can be built from the data, because each of those models uses only K –1 folds. In general, other things being equal, the more instances are used for learning the model, the better the predictive performance of the model. In order to build the best classification model to be reported to the user, the entire data set must be used in the learning process. Hence, in this thesis we use 10-fold cross-validation to estimate predictive accuracy and the full dataset, test data included, to build the final models (some of which are interpreted, as discussed in Chapter 4).

Table 2.1: The structure of a confusion matrix

		True Class	
		“+”	“-“
Predicted Class	“+”	TP	FP
	“-“	FN	TN

So far, we discussed the methodology used for estimating predictive accuracy, and now we turn to the discussion of actual measures of predictive accuracy. Consider a binary classification problem with two classes, denoted “positive” (“+”) and “negative” (“–”) classes. The predictive performance of a classification model can be summarized by a confusion matrix, whose structure is shown in Table 2.1. Each cell of this matrix show the number of test instances whose actual class is the class given in the corresponding column and whose predicted class is the class given in the corresponding row heading. The acronyms in the cells are defined as follows: TP = number of “true positive” instances, FP = number of “false positive” instances, FN =

number of “false negative” instances and TN = number of “true negative” instances. Hence, the cells in the main diagonal of the matrix (TP and TN) represent correct classifications, whereas the other two cells (FP and FN) represent different types of misclassifications.

The simplest measure, the classification accuracy, is defined as the number of correct classifications divided by the total number of classifications, i.e. (TP + TN) / (TP + FP + FN + TN). However, this measure is not suitable for evaluating predicting accuracy in datasets where the class distribution is very unbalanced. This is because it would be very easy to obtain a very high value of classification accuracy by always predicting the majority class. For instance, if the relative frequency of the majority class in the dataset is very high, like 95%, we could trivially obtain a (very high) classification accuracy of 95% by always predicting the majority class for all instances, regardless of the attribute values of that instance. However, that trivial classification model would be undoubtedly useless. Therefore, there are other measures that cope better with class imbalance, such as the precision, recall and F-measures (Japkowicz and Shah 2011), discussed next.

Consider a certain class, say the positive class. The precision measure is the ratio of the number of instances which truly belong to the positive class and were classified by the algorithm in that class divided by the total number of instances classified by the algorithm in the positive class. That is, Precision = TP / (TP + FP). Recall is the proportion of positive-class instances that were correctly classified as positive. That is, Recall = TP / (TP + FN). Recall is also called the rate of true positives. The F‑measure is the harmonic mean of precision and recall, calculated by equation (2.8):

$$F\text{-}measure = 2 \times \frac{Precision \times Recall}{(Precision\ +\ Recall)} \tag{2.8}$$

The above definition of precision, recall and F-measure is specific to the positive class, but the same measures are also computed for the negative class (or any other class, in problems with more than two classes). The final precision, recall and

F-measure values computed for a classification model are then an average of the corresponding values over all classes.

Note that this final (averaged across classes) F‑measure is a suitable measure of predictive accuracy in classification problems with imbalance class distributions, since its maximization requires both classes to be predicted very well, in term of both precision and recall.

2.1.4 The Problem of Imbalanced Class Distribution

In data mining, when the number of positive class instances is far greater than that of negative class instances, it can be considered as a serious problem for a classification algorithm (Chawla, Japkowicz and Drive 2004). Such problem is practically common in various applications such as medical data, anomaly detection, etc. (Longadge, Dongre and Malik 2013). In general, classification algorithms are at their top-form when the numbers of instances of each class label are roughly equal (García, Sánchez and Mollineda 2012). However, when the number of instances of one class label far exceeds the other, a classification algorithm tends to be very focused on the majority class, since it can result in a higher predictive accuracy. Accordingly, a classification model which learns from such imbalance data is very likely to make many mistakes when classifying the minority (negative) class. In terms of data mining applications, this is not what we aim to achieve, since it is normally more interesting to learn and predict the minority class (Chawla 2009). For example, in survey data the number of patients suffering from a certain disease (positive class label) is relatively smaller than that of patients not having the disease (negative class label). Therefore, encouraging the algorithm to discover the factor(s) causing the disease is more interesting and in general more useful than to learn what healthy patients have in common.

Dealing with imbalanced class distribution involves techniques such as improving classification algorithms or balancing class labels in the training set in a data pre-processing phase, before providing the data as input to the algorithm. The later technique is usually preferred as it has wider application and, thus, is the general approach followed in this work. The former technique includes algorithmic ensemble techniques which modify existing classification algorithms to enrich them in a way appropriate for imbalanced class distributions. The main objective of an ensemble technique is to improve the performance of single classifiers (Rokach 2010; Polikar and Robi 2006). Particularly, this approach involves constructing several classification models from the original data and then aggregate their predictions.

Unlike algorithm-based approaches, sampling-based approaches (Guo *et al.* 2008) are one of the well-known solutions for coping with the class imbalance problem in a pre-processing step. The main process is to resample the dataset, so that the class distribution changes toward a balanced one.

There are two main sampling techniques. The first one is undersampling, which randomly removes some instances of the majority class, so this class has less effect on the classification algorithm. Furthermore, the classification algorithm’s run time and the memory space are improved by reducing the size of the training set. Nevertheless, doing so might discard informative instances of the majority class, leading to potential loss of useful information. As a consequence, some majority (positive) instances could be classified as negative class incorrectly. In other words, this issue is known as “underfitting” where the number of instances in the training set is insufficient for the algorithm to capture the relationship between features and the class.

The second resampling technique is oversampling, which duplicates some randomly sampled minority class instances. Although such technique throws away no information, this could lead to the “overfitting” problem due to a few instances being repeatedly copied (Liu *et al.* 2010). As a consequence, the classification algorithm captures the noise of the data instead of the underlying trend.

In addition, another resampling-based approach called Synthetic Minority Over-Sampling TEchnique (SMOTE) was introduced to address the problem with class imbalance (Chawla *et al.* 2002). Such technique is simple and similar to the oversampling technique, yet it is considered more effective because of the following reason. Instead of creating exact copies of minority class instances, SMOTE synthetically constructs new minority class instances via an algorithm specifically designed for this task. As mentioned above, oversampling leads to the overfitting issue caused by many repeated randomly sampled instances. However, if the newly added instances are generated by an algorithm rather than being exact copies of some original instances, then the problem of overfitting can be prevented. In more details, new synthetic instances are generated in the following way. First of all, for each minority class instance (m) an instance (n) from its k nearest neighbours (KNN) is picked at random. After that, the SMOTE algorithm constructs a new synthetic instance (o) whose features' values are calculated taking into account the features' values of m and n. Specifically, the value of each feature is calculated according to equation (2.9).

$$o.f_i = m.f_i + (m.f_i - n.f_i) \times rand(0,1) \tag{2.9}$$

Essentially, the value assigned to a feature is a coordinate of a randomly sampled point along the line segment between *m.f*$_i$ and *n.f*$_i$. It should be noted that if the feature *f* is categorical, the majority vote is used for the nominal value amongst the KNN. With the use of KNN, not only does SMOTE mitigate the problem of overfitting caused by random oversampling, but also it results in no loss of information. The main disadvantage of SMOTE is that it is time consuming, since the KNN needs to be computed when constructing new synthetic instances. In addition, there is a risk that some of the new constructed minority class instances contain noisy data that could be harmful to the classification process. This is because, since the newly constructed instances do not represent data observed in the past, it is possible that those instances actually belong to the majority class, rather than to the minority class as assumed when they were constructed.

2.2 Feature Selection

In the context of the classification task, as a pre-processing step (before applying a classification algorithm), feature selection is performed to select a subset of relevant features, out of all original features. In general, there are several motivations for using such procedure (Li *et al.* 2016). The main one is to remove irrelevant, noisy, or redundant features, which can actually reduce the predictive accuracy of the classification model (Liu 1998). Another motivation is that identifying the most relevant features is a form of discovered knowledge by itself. In addition, feature selection can improve the interpretability of the classification model due to the smaller number of features used to build the model. Finally, reducing the number of features can substantially speed up the execution of the classification algorithm, hopefully without sacrificing the predictive accuracy.

Most feature selection methods have two main components. Firstly, a search method decides how to generate new subsets of features (candidate solutions) to be evaluated. Secondly, an evaluation function assigns a numerical value of quality to each candidate feature subset. The next subsection discusses different types of feature selection approaches based on different types of evaluation function, whilst the following subsection discussed some search methods for feature selection.

2.2.1 Filter, Wrapper and Embedded Approaches

There are three types of feature selection approaches, depending on how feature subsets (candidate solutions) are evaluated. The first and most popular one is the filter approach (Wang, Wang and Chang 2016), which evaluates a feature subset without running the target classification algorithm (i.e. the algorithm that will use the selected features to build a classification model). Typically, filter approaches use statistical tests or related criteria as an evaluation function. An example of a commonly used evaluation function is the Information Gain (IG) or Information Gain Ratio (IGR), as defined in equations (2.3) and (2.5) respectively, in Section 2.1.1. These criteria essentially measure the amount of information about the class distribution that is gained when the value of a feature is known. Hence, a straightforward ranking-based filter method consists of computing one of these criteria for each feature and then select the k features with highest values of IG or IGR, where k is a user-defined parameter.

Alternatively, Pearson’s chi-squared (χ^2) can be used as an evaluation criterion to rank features in the same way. In particular, this criterion evaluates how likely it is that any observed difference between the expected feature value and the observed feature value in an instance arose by chance. Its value is defined as shown in equation (2.10):

$$\chi^2 = \sum_{i=1}^{v} \sum_{j=1}^{w} \frac{\left(N_{x_i y_j} - E_{x_i y_j}\right)^2}{E_{x_i y_j}} \tag{2.10}$$

where x_i and y_j are the i^{th} and j^{th} values of feature X, with v values, and class variable Y, with w values, respectively. Nx_iy_j and Ex_iy_j denote, respectively, the observed and expected frequency with which the values x_i and y_j occur together in the same instance. Note that the observed frequency is a count computed from the training set, and the expected frequency is a count calculated using probability theory, by assuming that the class variable Y and the feature X are independent.

In spite of the fact that such ranking-based filter methods are relatively fast, the disadvantages of such univariate methods are non-trivial. For example, they ignore feature interactions, since they only measure association between each feature and the class variable, not detecting redundancy (strong associations) between features. Furthermore, these univariate filter methods, in general, rank the features according to their evaluation function, but after the ranking they still need a parameter (k) specifying which number of top-positions in the rank will be selected. An example of a filter method that avoids these limitations is the Correlation-based Feature Selection method, which is the basis for this research, and will be described in detail later in this Section.

In contrast to the filter approach, the wrapper approach and the embedded approach require running the target classification algorithm as part of the feature selection process. These allow these approaches to select features that are tailored to the target classification algorithm; unlike the filter approach, which selects features based on their intrinsic predictive power regardless of the target classification algorithm.

In essence, the wrapper approach evaluates the quality of a candidate feature subset by measuring the predictive accuracy (on a subset of the training data) of the classification model built with that feature subset. This approach is in general very computationally expensive, especially when the dataset has a very large number of features, since it requires many runs of a classification algorithm.

The embedded approach involves building a classification model and carrying out feature selection at the same time, rather than performing feature selection in a pre-processing step. For example, once a decision tree has been built, the relevant features are automatically selected by the algorithm. This approach can also be computationally expensive, depending on the type of classification algorithm used.

Therefore, in this research we focus on the filter approach, which is more computationally efficient (faster) and more scalable to a large number of features.

2.2.2 Types of Search Methods

Search methods (also called search strategies) are one of the two main components of a feature selection method. They decide how to generate new subsets of features (candidate solutions) to be evaluated by an evaluation function. As mentioned in (Liu 1998), search methods can be categorised into three broad types. The first one is complete search, which guarantees an "optimal" solution according to a predefined evaluation function – which does not necessarily guarantee the optimal predictive accuracy on the test set, unseen during training. In general, exhaustive search is a good example here (Branch and Bound (Narendra and Fukunaga 1977)) is also considered as complete search that guarantees an optimal feature subset), since it fully explores the search-space, i.e., it evaluates all possible feature subsets and selects the best candidate feature subset. In terms of computational efficiency, for a given set of *m* input features, the time complexity of this method is O(2^m), so it is categorised as an exponential time algorithm. As a consequence, the exhaustive search method is computationally feasible only if the number of input features is relatively small.

The second type of search method is heuristic search, which exploits only promising parts of the search space (the space of all candidate feature subsets). The quality of a candidate feature subset, which is used to decide which parts of the search space are explored, is measured according to a given evaluation function (a heuristic function). In other words, by sacrificing completeness it increases computational efficiency, since typically only a relatively small part of the search space is explored by such methods. Hence, heuristic search methods are more practical when the size of data set is large. Although this type of search method has no guarantee of finding the optimal solution, it attempts to find a near optimal solution within an acceptable computational time. Nevertheless, the most serious potential problem of such methods is that they can get stuck in a local minimum in the search space. Popular heuristic searches methods for feature selection include greedy search (or hill-climbing), beam search and best-first search.

We first explain greedy search, since it is simple to understand and to implement. In general, there are two main types of greedy search methods, namely greedy forward search and greedy backward search. The former initialises the set of selected features with the empty set and then adds one feature at a time to the current set of selected features. The feature added at each iteration is the best one, according to a predefined evaluation function. Features are added as long as this improves the value of the evaluation function.

In contrast with this type of method, greedy backward search initializes the set of selected features with the full set of features. Next, it removes one feature at a time from the current set of selected features (again, based on an evaluation function), as long as the value of the evaluation function does not degrade. Note that greedy backward search tends to be much slower than greedy forward search, since the former has to evaluate much larger feature subsets in the early iterations. Hence, greedy forward search methods are used more often than backward search methods in practice.

With regard to best-first search, the set of selected features is initialised with the empty set, the same as of the initialisation of greedy forward search. Then the method iteratively selects the best current feature subset, among the subsets generated so far by the search, and generates all possible new feature subsets by expanding the selected feature subset, i.e., adding a single feature to the selected subset. The whole process is repeated until a stopping criterion is satisfied, e.g. when none of the newly generated feature subsets has an evaluation function value better than the value of the most recently selected feature subset. Note that best-first search has to keep all unexpanded feature subsets in its memory during the search, since the best feature subset has to be selected among all those unexpanded subsets. Although a good evaluation function will improve the search's efficiency, the worst-case time complexity is still O(m^d) where *d* is the maximum depth of the search.

Because of this drawback, beam search was introduced by simplifying the best-first search method to focus more on exploitation of the best candidate solutions found so far, at the expense of performing less exploration of the search space. Instead of keeping all unexpanded feature subsets generated so far in the search space, beam search trims the possible paths to the best *b* subsets, where *b* is a parameter called the beam width. As a result, the worst-case time complexity is reduced to O(b^d), where *b* is much smaller than *m*. To conclude, heuristic search methods can find a reasonably good solution for many problems efficiently. Hence, this type of search type is used in this work.

Lastly, nondeterministic search methods can be used to avoid the problem previously mentioned for heuristic search, i.e, the problem of getting stuck in a local optimum in the search space. An example is the use of genetic algorithms for feature selection (de la Iglesia 2013; Goldberg 1989; Yang and Honavar 1998). Note that search methods of this type in general return different feature subsets when they are run with different values of a randomly-generated seed used to initialize the candidate solutions). Hence, they usually need to be run many times with different random seeds, and have their results aggregated over those many runs. This increases the computational time taken when using such methods.

Finally, it should be noted that the above types of methods are not mutually exclusive. In particular, non-deterministic methods can also be classified as a particular case of heuristic methods, since, due to their non-determinism, they do not guarantee to obtain the optimal solution.

2.2.3 Correlation-based Feature Selection (CFS)

Correlation-based Feature Selection (CFS) is a filter method which evaluates candidate feature subsets that can have multiple features, not just individual features – as it is the case with many simpler filter methods (Hall 2000). Unlike univariate filter methods based on ranking, CFS does not need a parameter for the number of selected features, it automatically decides the number of features to be selected. Moreover, CFS has the advantage of evaluating a subset of features, considering feature interactions, i.e., measuring, in particular, the degree of redundancy among features.

In essence, CFS works based on the following principle: good feature subsets contain features highly correlated with the class variable, but uncorrelated with each other, i.e., with little or no redundancy among features. To implement this principle, the standard CFS method (Hall 2000) tries: (a) to maximize the average correlation between each feature in a candidate subset and the class variable; and (b) to minimize the average correlation between each pair of features in a candidate subset. These two criteria can be combined into a single evaluation function as defined in equation (2.11):

$$Merit_{S_k} = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}} \tag{2.11}$$

where $Merit_{S_k}$is the merit of a feature subset S consisting of k features, $\overline{r_{cf}}$ is the average value of all feature-class correlations (given by equation (2.12)) with r_{cf_i} denoting the degree of correlation between feature *i* and the class variable, and $\overline{r_{ff}}$ is the average value of all feature-feature correlations (given by equation (2.13)) with $r_{f_if_j}$ denoting the degree of correlation between features *i* and *j*. In particular, the former represents the degree of relevance while the latter represents the degree of redundancy among the features in S.

$$\overline{r_{cf}} = \frac{\sum_{i=1}^{k} r_{cf_i}}{k} \tag{2.12}$$

$$\overline{r_{ff}} = \frac{\sum_{i=1}^{k} \sum_{j=1}^{i-1} r_{f_if_j}}{fp} \tag{2.13}$$

In addition, *fp* is the number of feature pairs in the feature subset *S*, which is given by equation (2.14):

$$fp = \frac{k(k-1)}{2} \tag{2.14}$$

More broadly, the term correlation is not limited to Pearson's correlation; it can also be measured by other measures of association between variables, e.g. an information-theoretic measure. Each type of association measures has its own advantages and disadvantages depending on the type of a variable pair to measure.

For example, when both features are continuous (real valued), Pearson's correlation is normally used to calculate their correlation, as shown in equation (2.15):

$$r_{xy} = \frac{\sum xy}{n\sigma_x\sigma_y} \tag{2.15}$$

where *n* is the number of instances in the dataset, σ_x and σ_y the standard deviations of variables x and y. However, when one feature is continuous and the other is categorical (nominal), a weighted Pearson's correlation is calculated as shown in (2.16) (Hall 2000). Specifically, for a categorical feature x and a continuous feature *Y*; if x has *v* values, then *v* binary attributes are correlated with *Y*; and then each of the binary features, X_{bi} = *1, ..., v*, takes value 1 in an instance when the ith value of X occurs in that instance and 0 otherwise. Each of the calculated correlations is weighted by the probability that X takes the ith value in the entire training set, as shown in equation (2.16).

$$r_{xy} = \sum_{i=1}^{v} p(X = x_i) r_{X_{bi}Y} \tag{2.16}$$

Likewise, when both features involved are categorical, binary features are created for both, and all weighted correlations are calculated for all pairs of one binary value of x and one binary value of y, as defined in equation (2.17), where *v* and *w* are the number of values that can be taken by X and Y, respectively.

$$r_{XY} = \sum_{i=1}^{v}\sum_{j=1}^{w} p\left(X = x_i, Y = y_j\right) r_{X_{bi}Y_{bj}} \tag{2.17}$$

By considering these three equations, it can be seen that Pearson's correlation works naturally with a pair of continuous variables, and it measures their linear correlation coefficient. Nevertheless, the correlation coefficient is less naturally applicable to categorical variables, which have to be transformed to a set of binary variables for computing the weighted correlation coefficient. Another point to consider is that the correlation coefficient is a measure of linear correlation, so that it may be zero or close to zero when a pair of features is non-linearly dependent.

Finally, it should be noted that there are non-standard versions of CFS for multi-label classification (where an instance can be assigned multiple class labels at the same time), as described in (Jungjit *et al.* 2013; Jungjit and Freitas 2015), but multi-label classification is out of the scope of this thesis.

2.2.4 Minimum-Redundancy-Maximum-Relevance (mRMR) feature selection

The mRMR method for feature selection was originally proposed in (Peng et al. 2005). In this method, the relevance of a feature set S consisting of k features to the class c, denoted *Rel(S,c)* is defined as the average value of all mutual information (MI) values between each individual feature f_i in S and the class variable c, as given by equation (2.18):

$$Rel(S,c) = \frac{1}{k}\sum_{f \in S} MI(f,c) \tag{2.18}$$

The redundancy of all features in the set S is the average value of all mutual information values between features f_i and f_j for all such pairs of features in S when the order of features in the pair is irrelevant, as given by equation (2.19):

$$Red(S) = \frac{1}{fp}\sum_{1 \le i < j \le k} MI(f_i, f_j) \tag{2.19}$$

where *fp* is the number of feature pairs in S defined in equation (2.14). To obtain a value of $Merit_S$ that represents a compromise between relevance and redundancy, $Merit_S$ is computed by using either of the following two approaches, namely the Mutual Information Difference criterion (MID), defined in equation (2.20), or the Mutual Information Quotient criterion (MIQ), defined in equation (2.21).

$$Merit_S = MID_S = Rel(S,c) - Red(S) \tag{2.20}$$

$$Merit_S = MIQ_S = \frac{Rel(S,c)}{Red(S)} \tag{2.21}$$

Note that when computing MID, *Rel(S,c)* and *Red(S)* should be normalised by rescaling each term to have values between 0 and 1. This rescaling is achieved by dividing the computed value of *Rel(S,c)* and *Red(S)* by the maximum value of *MI(f,c)* and *MI(*f_i*,*f_j*)*, respectively. In addition, when *Red(S)* is 0, one should compute the Merit using MID instead of MIQ, since the latter is undefined in this case. In general, MIQ outperformed MID in the experiments reported in (Ding and Peng 2005) but those experiments involved only five different datasets, so the results have very limited generality.

2.3 Longitudinal Classification

The vast majority of works on the classification task, including works performing feature selection in a pre-processing step, focus on analysing the standard type of classification data, where each variable – both features and the class variable – is associated with a single time point, so that there is no explicit temporal structure in the data. However, many important sources of data – particularly in the biomedical domain – contain longitudinal data, where the values of a variable are repeatedly measured across different time points, often called waves (Ribeiro *et al.* 2017). For instance, many hospital databases contain records with the results of blood tests measured for the same patient across many time points.

2.3.1 Basic Concepts of Longitudinal Data

Unlike standard (non-longitudinal) datasets, longitudinal datasets consist of features whose values are assigned at multiple time points, for each instance in a dataset. For example, a health-survey dataset, where instances represent patients, could contain features representing the results of different blood sample tests across several successive years. From a machine learning perspective, this type of datasets has temporal information about the features: how each feature's values change across time. In general, conventional classification algorithms do not explicitly exploit this temporal information, since they treat all occurrences of a feature in the same way, regardless of how recent the feature values are.

In addition, the different values of a feature across time can exhibit some temporal redundancy, in the sense that the value of a feature at a given time point may be correlated with values of the same feature in other time points (particularly closer time points). This is generally known as autocorrelation in the area of time series. Again, this kind of temporal redundancy is not explicitly detected by non-longitudinal classification or feature selection algorithms, which would not distinguish between measuring the correlation between two values of the same feature in two different time points (temporal redundancy) and measuring correlation between the values of two very different features in the same time point (non-temporal redundancy). By identifying these two types of redundancy, one can develop a feature selection algorithm that exploits the difference between them in order to try to improve the effectiveness of the feature selection procedure, as will be seen later.

As mentioned earlier, CFS can eliminate redundant and irrelevant features, but standard CFS ignores the temporal relation among the features, so that it does not explicitly address the above mentioned temporal redundancy as a specific issue in longitudinal datasets. In the next Section, we briefly review related work on longitudinal feature selection methods, which were explicitly designed for longitudinal classification data. First, however, the next two subsections briefly discuss basic approaches for longitudinal classification and different types of longitudinal classification algorithms.

2.3.2 Two Basic Approaches to Cope with Longitudinal Classification Data

In general, there are two approaches for longitudinal classification. The first one is the problem transformation approach, which transforms a longitudinal dataset into a non-longitudinal dataset, before applying a conventional classification algorithm. The second approach is the algorithm-adaptation approach, which adapts a non-longitudinal classification algorithm for longitudinal datasets. In this paper, we focus on the problem transformation approach, which is more generic (algorithm-independent), so that we can apply different classification algorithms and analyse different types of classification models.

2.3.3 Types of Longitudinal Classification Algorithms

As mentioned in (Jie *et al.* 2017), classification models built by longitudinal classification algorithms can be categorised into four types. The first type is Single-time-point Input Single-time-point Output (SISO). Most of the standard classification algorithms are in this category (when such algorithms are applied to longitudinal data). In other words, temporal information is not taken into account at all when predicting the class variable.

The other three types of algorithms are considered to be true longitudinal classification algorithms. The second type is Multiple-time-points Input Single-time-point Output (MISO). Examples of algorithms in this category can be found in (Minhas *et al.* 2015; Chen et al. 2015). The third type is Single-time-point Input Multiple-time-points Output (SIMO) and the last and most complex type is Multiple-time-points Input Multiple-time-points Output (MIMO). Examples of algorithms in MIMO can be found in (Adhikari *et al.* 2015; Cheung *et al.* 2015).

In this work, the proposed longitudinal feature selection algorithms – described in detail in Sections 3.2, 3.3 and 3.4 – cope with features (input) occurring in multiple time points and a class variable (output) occurring in a single time point. Hence, the algorithms proposed in this work belong to the MISO category.

2.4 Related Work on Feature Selection for Longitudinal Classification

Although there is a huge literature on conventional (non-longitudinal) feature selection (Li *et al.* 2016; Liu 1998; Wang, Wang and Chang 2016), there are relatively few published studies on longitudinal feature selection for classification tasks (Radovic *et al.* 2017; Lou and Obradovic 2012). In this section, we discuss the advantages and disadvantages of the longitudinal feature selection methods most related to this work.

In (Radovic *et al.* 2017), a longitudinal feature selection method was proposed for temporal gene expression data. They used the Minimum Redundancy Maximum relevance (mRMR) method, whose evaluation function is conceptually similar to the CFS method's one, being based on maximising the candidate features' relevance with respect to the class variable and minimising redundancy among the candidate features. The degree of relevance is represented by the arithmetic mean of the F-statistic (Dytham 2011; Tabachnick and Fidell 2013) for that feature over all the different time points. One drawback of this method is that the degree of relevance is computed separately for each time point, ignoring important differences between a feature's values at different time points – e.g. ignoring that feature values at recent time points are intuitively more relevant for class prediction than older feature values. Furthermore, using the F-statistic makes the strong assumption that the data are normally distributed. In addition, in this feature selection method the degree of redundancy among candidate features is measured by a distance derived from Dynamic Time Warping (DTW). DTW is also employed in other feature selection works, such as (Furlanello, Merler and Jurman 2006).

Another related work is (Lou and Obradovic 2012) which proposed a margin-based feature selection method which transforms a feature space into a weighted feature space. A temporal margin is defined based on a measure of distance between two time points, and then it selects the features with large weights that maximise each temporal margin. Although this method makes no assumption about the data distribution, it only considers a feature's relevance with respect to the class. In other words, the redundancy among features is ignored.

3 DATASET CREATION AND PROPOSED VARIANTS OF CORRELATION-BASED FEATURE SELECTION

In this chapter, we introduce the two types of contributions of this thesis. The first one is the creation of the longitudinal datasets used in the experiments, including the creation of features capturing longitudinal information for predicting age-related diseases. The second one is presented in three sections, which describe four proposed variants of the Correlation-based Feature Selection (CFS) method.

This chapter is organised as follows. Section 3.1 gives the details of how longitudinal datasets were constructed for predicting age-related diseases. Section 3.2 presents a variant of the Correlation-based Feature Selection (CFS) method based on exhaustive search applied with each small group of features. Section 3.3 presents another CFS variant that combines the previous variant with the application of the standard CFS method. Section 3.4 presents two other CFS variants that are based on assigning different weights to different types of redundancy between features.

The contents of Sections 3.1 and 3.2 have been partly included in a peer-reviewed paper that is currently in press (Pomsuwan and Freitas 2017).

3.1 Creation of Longitudinal Datasets for Predicting Age-Related Diseases

The classification datasets created in this work were derived from the English Longitudinal Study of Ageing (ELSA) (Marmot *et al.* 2016) – https://www.elsa-project.ac.uk/. The ELSA study is a longitudinal survey of ageing and quality of life among older people that explores the dynamic relationships between health and functioning, social networks and participation, and economic position as people plan for, move into and progress beyond retirement. In this work, however, we focus only on the biomedical data in ELSA, such as the results of blood tests and other data collected by nurses, and the relationship between that data and the health status of patients, as will be described in more detail later.

The ELSA subjects were recruited from a representative sample of the English population, who live in private households, aged 50 and over (Marmot *et al.* 2016). In addition, the data has been collected every two years, where each data collection period is known as a 'wave', so that we can observe the variation of each feature's values for each individual across those waves. In total, seven waves of data were collected and have well-documented data.

It should be noted that the data in the ELSA database was not collected specifically for machine learning purposes. Hence, we had to spend a large amount of time with data preparation for the classification task. The first step was to define the instances (objects to be classified), the classes and the predictive features used for classification. In essence, the instances represent individuals in the ELSA database, the class variables represent age-related diseases and the features represent biomedical information collected by nurses or other relevant characteristics of an individual (age and gender). The next three subsections describe data preparation in detail.

3.1.1 Creating class variables representing age-related diseases

We aim at building classification models which help us understand what health factors play an important role in predicting whether or not a patient will have a certain age-related disease in the future. Therefore, we looked into the ELSA core data, and then identified 10 age-related diseases, each used as a class variable in this work. These diseases are: angina, arthritis, cataract, dementia, diabetes, high blood pressure, heart attack, osteoporosis, Parkinson's and stroke. Hence, we created 10 datasets, each one with a different disease as the class variable to be predicted. More precisely, in each dataset, the binary class variable indicates the presence or absence of the corresponding disease in wave 7 (the most recent wave in ELSA).

Note, however, that for each disease, there is no variable in the ELSA database that directly indicates whether or not an individual has that disease in a given wave. This kind of information is rather represented indirectly, by several related variables whose values depend on both whether or not the individual (patient) had the target disease in the past and whether or not the patient still has the disease or whether the disease was first diagnosed in the current wave. Therefore, we needed to create a well-defined class variable for each disease separately, combining information from the several related variables associated with that disease. In order to create such class variables, in general the following rule was used for each disease, combining information about that disease's variables in wave 7:

```
IF ("whether confirms the disease diagnosis" = "yes")
OR ("whether still has the disease" = "yes")
OR ("the disease diagnosis newly reported" = "yes")
THEN Disease = "yes"
OTHERWISE Disease = "no".
```

In this rule, the terms between double quotes just before each "=" sign in the "IF" part of the rule refer to original variables in ELSA's wave 7 core data. For example, if the binary variable indicating whether or not 'a heart attack diagnosis has been confirmed' is set to yes (the first condition in the IF statement), then the class variable Heart-Attack is consistently assigned the value "yes". Note that, although each dataset has a different class variable, all datasets contain instances representing the same individuals and the same set of predictive features (as described next).

3.1.2 Creating predictive features based mainly on the Nurse data

In the created datasets, most features were created from raw variables available in the Nurse Visit data, part of the previously discussed ELSA database (Marmot *et al.* 2016). Those raw variables represent several types of biomedical information collected by a nurse, including for instance many types of blood sample tests. In addition, the nurse took several physical performance measurements that involved asking a patient to move his/her body in different ways. If a particular movement could not be done by the participant or he/she felt that it was unsafe to try to do it, the attempt was marked as ‘Not attempted’ or ‘Test not completed’. The Nurse variables are only available at ELSA waves 2, 4, and 6, so our created datasets contain only features for these waves. These features are then used to predict age-related diseases (classes) at the later wave 7, whose data were collected about two years later than the data in wave 6.

As mentioned earlier, the raw biomedical variables collected by the nurses were not collected specifically for machine learning, and they contain a large amount of obviously redundant or irrelevant information. Hence, we have created features for classification by extracting and combining information from the raw variables in the Nurse data files, as follows. First of all, we kept potentially predictive variables from the Nurse data, whilst many other variables which seem intuitively useless for predicting age-related diseases were removed because such variables were collected mainly to record problems in data collection for other variables. For example, several variables capturing information such as the reasons why taking a blood sample test was refused by a patient, and information about several types of problems in some physical performance measurements, were discarded.

In addition, many variables in the Nurse data represented clearly redundant information, in cases where the same variable (e.g. the result of a blood test) was measured in three different times in the same wave, in order to represent the variability in test results. This resulted in duplication of variables representing the same biomedical property in each wave, and none of those three measures can be considered ‘better’ than the other two. Hence, instead of using any of the three underlying variables, we created a feature defined as the mean value over those three measures, for each individual (instance), for each wave.

Another point to consider is the occurrence of different types of missing values in many raw variables in the Nurse data, which were originally labelled as different negative values, as follows (using as example a blood test result variable):

- -1 = Not applicable
- -6 = Period between collection and receipt in the lab > 5 days
- -8 = Don't know
- -9 = Refusal
- -11 = Blood sample not taken

Considering all these types of missing values separately would considerably complicate the task of the classification algorithms. Hence, to simplify, all these different negative values were assumed to have the same meaning of "missing value", so that we treated them in the same way by replacing all of them with the special missing value symbol "?" (the symbol used in the data mining tool WEKA (Hall *et al.* 2009)).

In addition to features created from the raw variables in the Nurse data files, we also included in our datasets two features directly extracted from the Core files in ELSA which intuitively represent potentially very relevant information for predicting age-related diseases, namely the features "w7indager" (age) and "indsex" (gender).

Finally, an important point is that, when creating the instances used in our datasets, only data from "core" members were used, so the ELSA records of their partners were ignored. The ELSA variable *"idauniq"*, which is a unique id for each individual, was added to our datasets to match up data about the same core member in different dataset files (across different waves). This variable was not used for classification purposes, of course, since it has no predictive power. Note that an instance was created for an individual only if that individual participated in wave 7, so the class variable values are available for all individuals (instances) in all created datasets. However, some individuals in our datasets may not have participated in all waves used to create features (i.e., waves 2, 4 and 6). If an individual did not participate in a given wave, the corresponding features in that wave will have a missing value for that individual, and the feature selection and classification algorithms cope with those missing values in their own ways.

3.1.3 Constructing Longitudinal Features

Recall that the features created from variables in the Nurse data (the vast majority of features in the created datasets) are measured across three different time points (waves), namely waves 2, 4 and 6 of the ELSA database. We use the term "conceptual feature" to refer to the abstract concept of such a feature regardless of its observed value in any given wave. E.g., "*chol*" (Blood total cholesterol level) is a conceptual (abstract) feature which is associated with three actual features, *w2chol*, *w4chol* and *w6chol*, which represent the observed value of that variable in waves 2, 4, and 6. For each such a conceptual feature, we created new features trying to capture temporal trends in the variation of that feature's values across the three waves, as follows.

First of all, we create *m* groups of temporally related features, one group for each of the *m* conceptual features, each group including all temporal variations of a conceptual feature across waves 2, 4 and 6, which are the waves before the wave with the class to be predicted (wave 7). This is, each group contains observed features that are variations of a conceptual feature across different waves. In the next step, these observed features are used to create six different types of Constructed Longitudinal Features (CLFs) – explained below. It should be noted that these kinds of constructed features only work for continuous (real-valued) observed features. Note also that CLFs are created for each group of temporally related features. Hence, each such group contains two types of features: the observed features for a given conceptual feature and CLFs constructed from those observed features. Next, we describe the several types of CLF.

The first type of CLF is *mono_w246*, indicating whether the value of a given feature monotonically increases or decreases across waves 2, 4 and 6. To illustrate this point, let $f_{(2)}$, $f_{(4)}$ and $f_{(6)}$ be the numeric values of feature f in waves 2, 4 and 6. Then, *f_mono_w246* (*mono_w246* for feature *f*) has the value 1 (monotonic increase) if $f_{(2)} < f_{(4)} < f_{(6)}$, the value -1 (monotonic decrease) if $f_{(2)} > f_{(4)} > f_{(6)}$, or value 0 (no monotonic property) otherwise.

However, a few features have their values observed in only two waves. This means that a *mono_w246* variable for such features cannot be created using the rule mentioned above. Accordingly, for such features we create instead another type of CLF, called *up_w*t_1t_2, which indicates whether the values of feature *f* in the two time-indices (wave numbers) t_1 and t_2 go up or not. For instance, *f_up_w24 has the value* 1 if $f_{(2)} < f_{(4)}$, or value 0 otherwise. Note that if the value of feature *f* is missing in any of the waves, either of these CLFs has a missing value (denoted by "?"). We create two types of *up_w*t_1t_2 features, namely *f_up_w24* and *f_up_w46* – there was no need for a CLF of the type *f_up_w26*, since no feature in our dataset had values only in waves 2 and 6.

Furthermore, each of the other three proposed types of CLFs represents the difference between the values of a pair of features referring to the same conceptual feature in two different waves. More precisely, let *f_diff_wij* denote the difference between the values of feature *f* in the two time-indices (wave numbers) *i* and *j*, for each of the three pairs of waves where $j > i$. Then, these CLFs are defined as follows:

- *f_diff_w24* = $f_{(4)} - f_{(2)}$
- *f_diff_w46* = $f_{(6)} - f_{(4)}$
- *f_diff_w26* = $f_{(6)} - f_{(2)}$

Hence, positive (negative) values of these constructed features denote an increase (decrease) in the value of feature *f* with time, between the two corresponding waves.

Table 3.1: All conceptual features used in the created data sets

Feature (Variable)	Description in the ELSA database, or definition	Available in			Numeric
		wave 2	wave 4	wave 6	
indsex	Sex - Priority: DiSex, DhSex	Not Applicable			
w6indager	Definitive age variable collapsed at 90+ to avoid disclosure	Not Applicable			✓
clotb	Blood sample: Whether has clotting disorder	✓	✓	✓	
fit	Blood sample: Whether ever had a fit	✓	✓	✓	
apoe	Blood APOE level (mmol/l)	✓			✓
hasurg	Whether respondent had abdominal or chest surgery in last 3 weeks	✓	✓	✓	
eyesurg	Whether respondent has had eye surgery in the last 4 weeks	✓	✓	✓	
hastro	Whether admitted to hospital for heart complaint in last 6 weeks	✓	✓	✓	
chestin	Whether respondent had any respiratory infection in last 3 weeks	✓	✓	✓	
inhaler	Whether used an inhaler/puffer in last 24 hours	✓	✓	✓	
mmssre	Side-by-side stand: Outcome	✓	✓	✓	
mmstre	Semi-tandem stand: Outcome	✓	✓	✓	
mmftre2	(D) Outcome of full tandem stand according to age	✓	✓	✓	
mmlore	Leg raise (eyes open): Outcome	✓	✓	✓	
mmlsre	Leg raise (eyes shut): Outcome	✓	✓	✓	
mmcrre	Chair rise: Single chair rise outcome	✓	✓	✓	
mmrroc	(D) Chair rise: Outcome of multiple chair rises, split by age	✓	✓	✓	
hipval	(D) Valid Mean Hip (cm)	✓	✓		✓
whval	(D) Valid Mean Waist/Hip ratio	✓	✓		✓
htpf	Highest technically satisfactory PF reading (litres per minute)	✓	✓		✓
wbc	White blood cell count (x 10^9 cells/litre)		✓	✓	✓
mch	Blood mean corpuscular haemoglobin level (pg/cell)		✓	✓	✓
sysval	(D) Valid Mean Systolic BP	✓	✓	✓	✓
diaval	(D) Valid Mean Diastolic BP	✓	✓	✓	✓
pulval	(D) Valid Pulse Pressure	✓	✓	✓	✓
mapval	(D) Valid Mean Arterial Pressure	✓	✓	✓	✓
cfib	Blood fibrinogen level (g/l)	✓	✓	✓	✓
chol	Blood total cholesterol level (mmol/l)	✓	✓	✓	✓
hdl	Blood HDL level (mmol/l)	✓	✓	✓	✓
trig	Blood triglyceride level (mmol/l)	✓	✓	✓	✓
ldl	Blood LDL level (mmol/l)	✓	✓	✓	✓
fglu	Blood glucose level (mmol/L) - fasting samples only	✓	✓	✓	✓
rtin	Blood ferritin level (ng/ml)	✓	✓	✓	✓
hscrp	Blood CRP level (mg/l)	✓	✓	✓	✓
hgb	Blood haemoglobin level (g/dl)	✓	✓	✓	✓
hba1c	Blood glycated haemoglobin level (%)	✓	✓	✓	✓
htval	(D) Valid height (cm)	✓	✓	✓	✓
wtval	(D) Valid weight (Kg) inc. estimated>130kg	✓	✓	✓	✓
bmival	(D) Valid BMI - inc estimated>130kg	✓	✓	✓	✓
wstval	(D) Valid Mean Waist (cm)	✓	✓	✓	✓
htfvc	Highest technically satisfactory FVC reading (litres)	✓	✓	✓	✓
htfev	Highest technically satisfactory FEV reading (litres)	✓	✓	✓	✓
mmgsd_me	Created variable: grip strength: dominant hand (Kg), mean of 3 measures (mmgsd1, mmgsd2, mmgsd3)	✓	✓	✓	✓
mmgsn_me	Created variable: grip strength: non-dominant hand (Kg), mean of 3 measures (mmgsn1, mmgsn2, mmgsn3)	✓	✓	✓	✓

Table 3.2 : Six types of Constructed Longitudinal Features (CLFs)

Feature (Variable)	Description in the ELSA database, or definition	Numeric
f_mono_w246	CLF: whether the value of VAR monotonically increases (1), decrease (-1), or otherwise (0)	
f_up_w24	CLF: whether the value of f increases (1), or not (0), from the wave 2 to wave 4	
f_up_w46	CLF: whether the value of f increases (1), or not (0), from the wave 4 to wave 6	
f_diff_w24	CLF: f value in wave 4 - f value in wave 2	✓
f_diff_w46	CLF: f value in wave 6 - f value in wave 4	✓
f_diff_w26	CLF: f value in wave 6 - f value in wave 2	✓

Table 3.1 shows the full set of 44 conceptual features used in all the datasets created in this work. This table shows, for each conceptual feature, its name and its description or definition in the ELSA database (Marmot *et al.* 2016), the data source used to create the features.

Note that the first two features, namely gender (indsex) and age (w6indager), have just one value for each individual. The value of gender is obviously independent of the wave numbers, whilst the age value is from wave 6. Although age values are also available in waves 2 and 4, such values are not used since they are obviously redundant, given the age value at wave 6.

The other 42 rows in Table 3.1 represent features from the Nurse data in ELSA, which in general are longitudinal features, having different values across waves (time points) for each individual. 36 of these 42 longitudinal features have values in 3 waves, whereas the other 6 are only available in some waves – more precisely: one feature (apoe) occurs only in wave 2, three features (hipval, whval, htpf) occur only in waves 2 and 4, and two features (wbc, mch) occur only in waves 4 and 6.

Since 5 conceptual features have values in only two waves, each of those 5 conceptual features generates four features in our datasets, i.e., one feature for each of the two waves plus two CLFs (one $up\_wt_1t_2$ feature and one $diff\_wt_1t_2$ feature, as defined earlier). Furthermore, out of the 36 conceptual features having values in 3 waves, there are 22 conceptual features whose values are continuous (real-valued).

Therefore, each of those 22 conceptual features generates 7 features in our datasets, i.e., one feature for each of the three waves plus 4 CLFs (namely *mono_w246*, *diff_w24, diff_w46* and diff_w26 CLFs, as defined earlier).

In addition, Table 3.2 shows the six types of CLFs, as explained earlier in this section. To sum up, the total number of features is 219.

Regarding missing values, a common approach to cope with this problem in standard (non-longitudinal) classification is to replace a missing value by a default value, typically the mean of the known values of the feature across the dataset, in the case of numerical features; or the mode (most frequent value), in the case of nominal features. This is a computationally efficient (fast) but naïve approach, which may introduce noise in the data. However, in our context of the constructed temporal difference features for longitudinal classification, we can exploit additional temporal information about feature values when calculating the value that will replace the missing value (instead of using a pre-defined default value), as follows.

$$f\_diff\_wij_x = \frac{f\_diff\_wkj_x \times mean\_f\_diff\_wij}{mean\_f\_diff\_wkj} \quad (3.1)$$

Let *i* and *j* be the indices of two waves associated with a temporal difference feature based on a given feature *f*, denoted by (*f_diff_wij*). If the value of the base feature *f* is missing for a given individual (instance) *x* in one of those two waves (say wave *i*), and the value of *f* is known in the other two waves (*j* and *k*), then the missing value of the constructed *f_diff_wij* feature for *x* will be replaced by a value calculated by equation (3.1), where wave index *k* denotes the "third" wave (i.e. nor wave *i* nor wave *j*) available in the dataset, so that data from all three waves are used to estimate the missing value. In addition, it should be noted that such method only copes with the missing values for the constructed features, i.e., it does not attempt to fill in the missing values for the base feature. This latter possibility is left for future research.

In equation (3.1), *mean_f_diff_wij* and *mean_f_diff_wkj* are the mean values of all known values of the constructed *f_diff features* for the corresponding waves. For example, if the value of *f* is missing in wave 4 for a given individual *x*, the value of the constructed feature *f_diff_w24* for *x* is computed as:

$f\_diff\_w26_x \times (mean\_f\_diff\_w24_x / mean\_f\_diff\_w26_x)$.

The motivation for this approach is that it considers not only the known values of *f* for other individuals in wave *i*, but also the known values of *f* for both the same individual and other individuals in waves *j* and *k*. In other words, the ratio *mean_f_diff_wij* to *mean_f_diff_wkj* acts as a normalization factor, correcting for the different scales of *f_diff* values in different time periods.

3.1.4 Imbalanced Class Distribution in the Created Datasets

The pie charts shown in Figure 3.1 provide information about how many patients had a certain age-related disease, where each chart represents one disease (a class variable) in wave 7, in the created datasets. In this figure, the numbers beside each pie represent the number and percentage of instances in each class, for each disease. It can be seen that the class distribution is imbalanced for every class variable; and in several cases extremely imbalanced. As a consequence, the predictive accuracy of classification models learning from imbalanced training sets is normally biased in favour of the majority classes (He and Garcia 2009).

Hence, a class balancing approach was necessarily applied to the datasets before running the feature selection and classification algorithms. Specifically, the instances of the majority class were reduced by using the undersampling technique (Batista, Prati and Monard 2004) in such a way that the class distributions dropped to the ratios of 4:1, 2:1 and 1:1. The undersampling approach was performed by randomly deleting an instance belonging to the majority class from the current training set until the desired ratio was reached. Note that undersampling was applied only to the training set (not to the test set), to encourage the construction of models predicting the minority class a reasonable number of times, instead of models predicting (almost) always the majority class. If undersampling was applied to the test set too, the classification problem would be transformed into a much easier one, which would be a very different problem from the original real-world one – an undesirable situation for our goal of analysing real-world human ageing data.

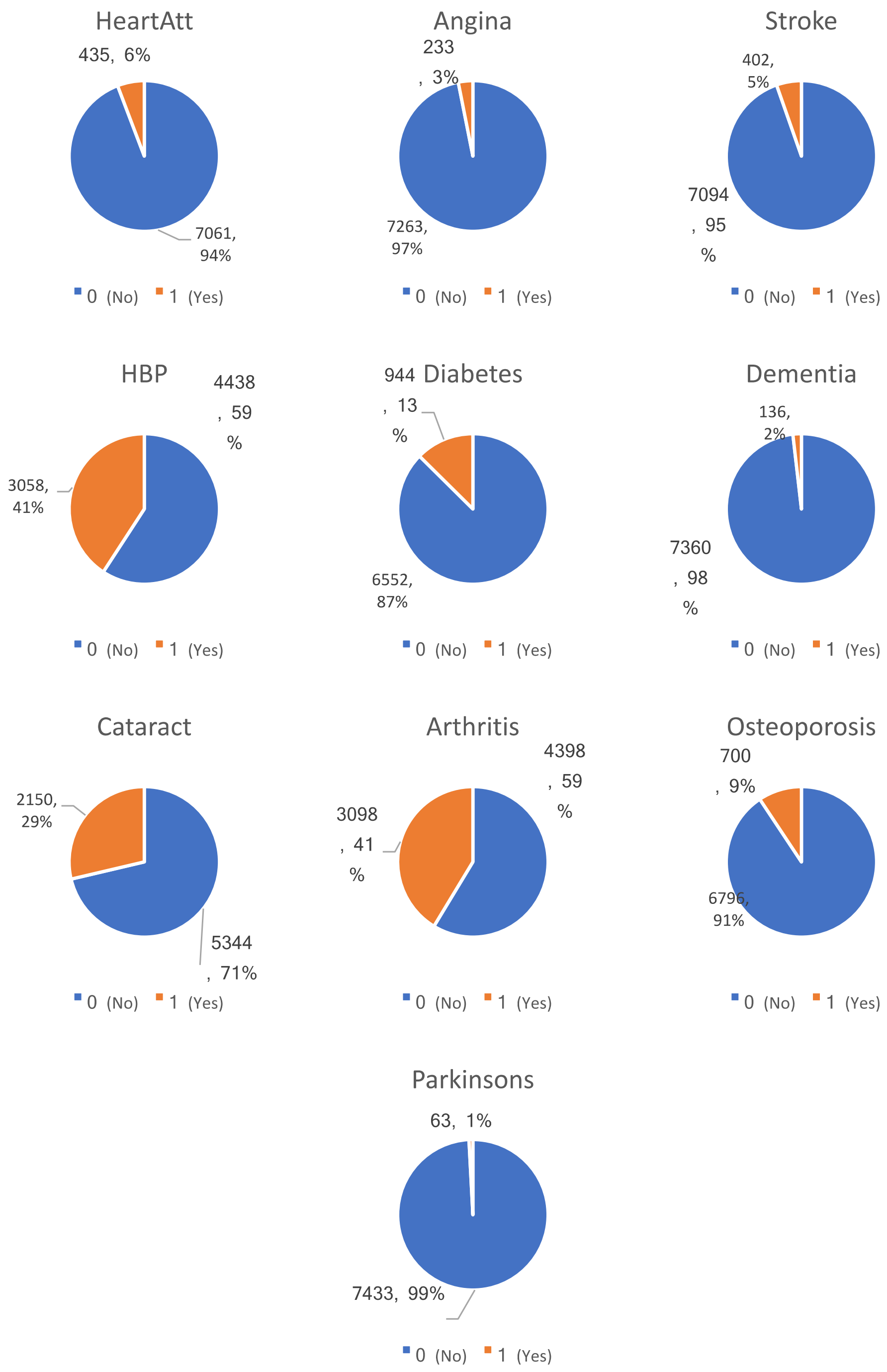

Figure 3.1: the class distribution of each class variable in wave 7

3.2 Exhaustive Search Correlation-based Feature Selection applied within each Conceptual Feature Group (Exhaustive CFS per Group: Exh-CFS-Gr)

The Exh-CFS-Gr method is based on the idea of first dividing the set of features into groups of temporally related features, with one group for each conceptual feature (see Section 3.1.3). Each group contains two types of features: (a) all features representing different values of a conceptual feature across the different waves (time points); (b) Constructed Longitudinal Features (CLFs) for the corresponding conceptual feature. For instance, the group of features for the conceptual feature "chol" (cholesterol level) contains 7 features: w2chol, w4chol, w6chol, chol_mono_w246, chol_diff_w24, chol_diff_w46 and chol_diff_w26; where the first 3 features are the chol values at waves 2, 4 and 6, and the last 4 features are CLFs.

Recall that the CFS method (discussed in Section 2.2.3) consists of a search method and an evaluation (Merit) function. Here we propose a variation of CFS that involves the search method only, whilst using the same Merit function as CFS.

The basic idea of the proposed variant of CFS is to use exhaustive search to select features separately from each group of temporally related features, rather than using a heuristic search method applied to the full set of input features as in the original CFS. That is, exhaustive search evaluates all possible feature subsets for each group of temporally related features, and selects the best candidate feature subset within each group based on the CFS Merit function.

Note that the exhaustive search method is computationally feasible only if the number of candidate features is relatively small. This is because, for a given set of m candidate features, the number of candidate feature subsets evaluated by this method is $2^m - 1$, where the 1 being subtracted refers to the empty feature set, which is considered an invalid candidate solution for the feature problem. When using the original CFS method, in general m is the number of features in the dataset, which is typically too large to allow the use of exhaustive search in real-world applications.

However, in the proposed CFS variant, the division of the features into groups, as explained in the previous Subsection, effectively creates groups that are small enough to allow the use of exhaustive search. More precisely, given the previously defined groups of temporally related features, the number of features in each group is at most 7 (3 observed features and 4 CLFs), as discussed in detail in the previous Subsection. Therefore, in order to address the temporal redundancy problem, exhaustive search is applied to each feature group separately.

Afterwards, the algorithm simply merges all sets of selected features across all groups – i.e., it applied the set union operator to all the selected feature subsets. Hence, a single feature subset is obtained and output as the result from the feature selection process. Note that this merging process ignores the redundancy between features in different groups; which is a limitation that will be addressed by another variant of CFS proposed in the next Section.

We call this entire feature selection process (i.e., the selection of features separately per group based on exhaustive search, and the final merging of the selected features across the groups) the Exhaustive CFS per Group (Exh-CFS-Gr) method. The basic idea of the proposed Exh-CFS-Gr method is summarized in graphical form in Figure 3.2.

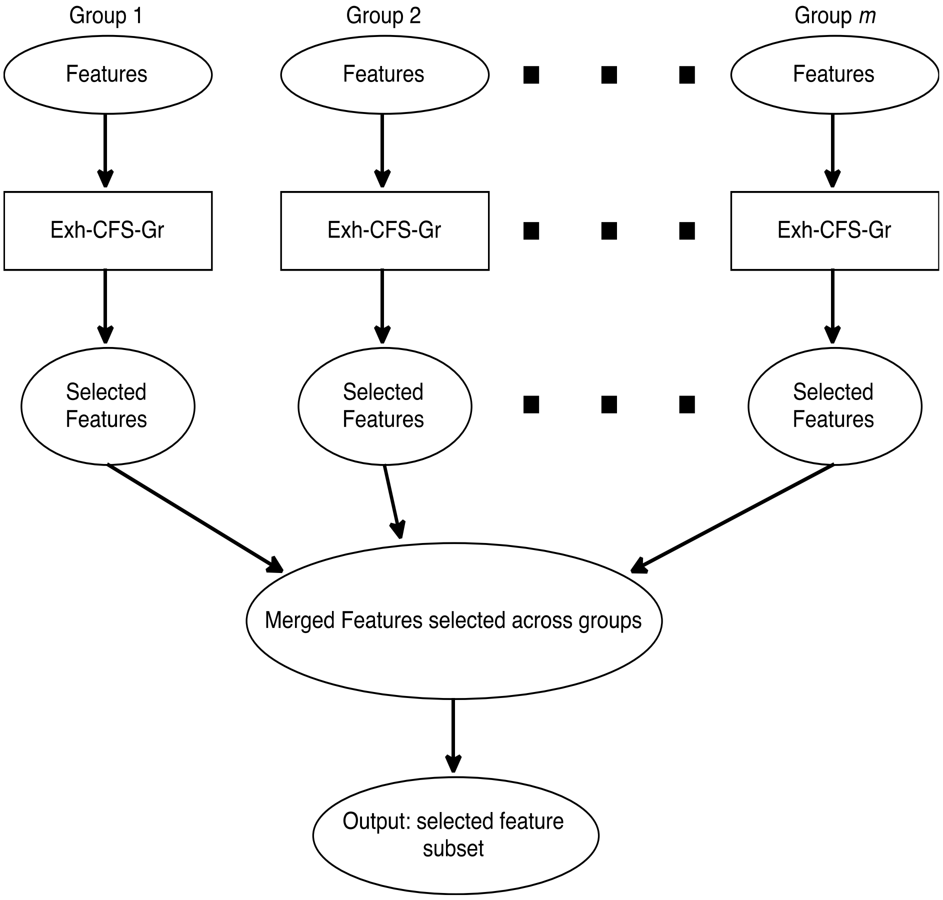

Figure 3.2: The basic idea of the proposed Exh-CFS-Gr method

3.3 Exh-CFS-Gr followed by standard CFS (Exh-CFS-Gr+CFS)

The previously proposed Exh-CFS-Gr method only deals with the redundancy issue within a conceptual feature group, so it does not detect redundancy between features across groups – i.e. redundancy between features derived from different conceptual features. In order to avoid this limitation, we propose another variant of CFS that performs feature selection in two phases.

The first performs feature selection within each group of temporally redundancy features using exhaustive search and then computes the set union operation of all the selected feature subsets, i.e., the first phase simply executes the previously described Exh-CFS-Gr method. In the second phase the algorithm simply applies the standard CFS method to the merged feature subset output by the first phase, in order to select the final set of features across all groups. We call this CFS variant the Exh-CFS-Gr+CFS method.

Note that in the second phase, when standard CFS is applied, the set of input features for the second phase is typically much larger than the set of features within each group. As a result, the set of input features for the second phase normally is not small enough to allow the application of exhaustive search, and indeed this was observed in our datasets, where typically many tens of features are the input for the second phase. Hence, we use a greedy forward search, rather than exhaustive search, to implement the standard CFS in the second phase. In essence, the greedy forward search works as follows. First of all, it initialises the set of selected features with the empty set. Afterwards, it adds one feature at a time (the best feature, according to CFS' Merit function) to the current set of selected features, as long as this improves the Merit value. The whole process of Exh-CFS-Gr+CFS is illustrated in Figure 3.3. Note that the only difference between this Figure and the Figure 3.2 (for Exh-CFS-Gr) is that Figure 3.3 has one extra operation (the application of standard CFS) at the end.

Note also that Exh-CFS-Gr+CFS tends to select a much smaller subset of features than Exh-CFS-Gr, since the former performs an additional feature selection step in the second phase. On the other hand, Exh-CFS-Gr+CFS is of course computationally slower than Exh-CFS-Gr, for the same reason of performing an additional feature selection step.

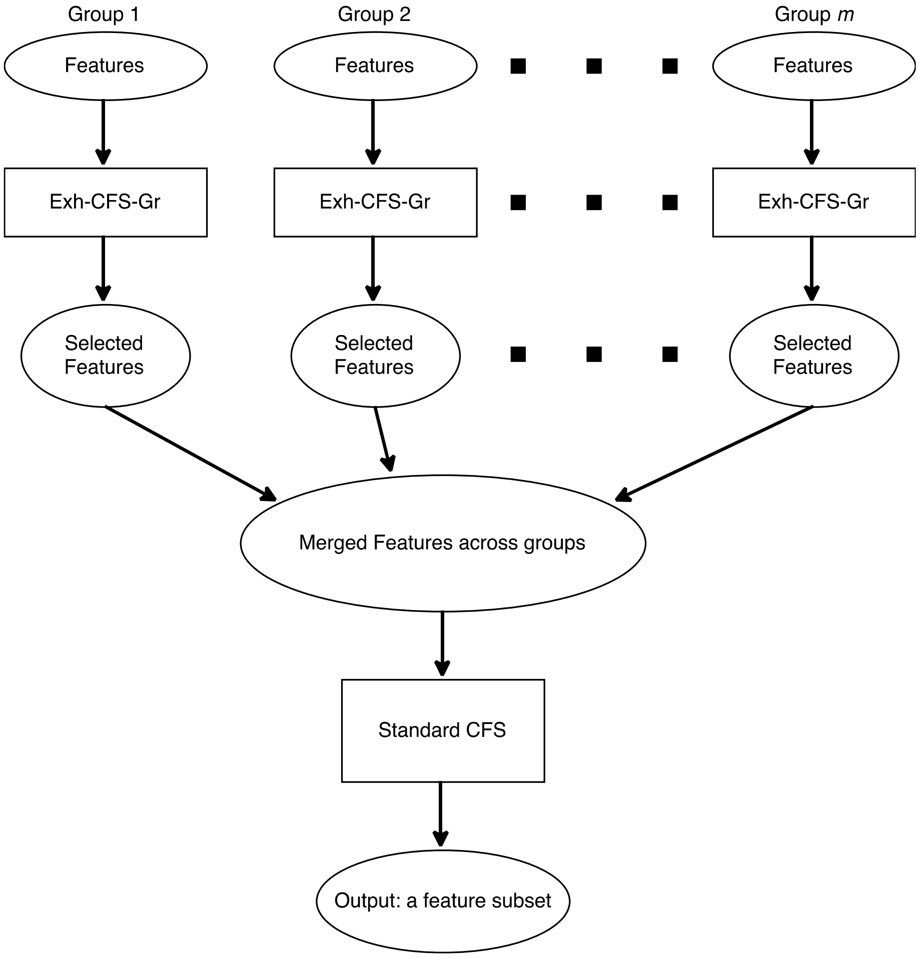

Figure 3.3: The basic idea of the proposed Exh-CFS-Gr+CFS method

3.4 Weighted Redundancy CFS (WR-CFS)

The two variants of CFS proposed in the two previous sections modified only the search method of CFS, whilst using the same Merit function used by the standard CFS. By contrast, in this subsection we propose two variants of CFS that modify the way the Merit function is computed, whilst using the same search method used in the standard CFS.

Before we describe the proposed variants of the Merit function, let us first recall that in the standard CFS the merit of a candidate feature subset S consisting of k features is calculated by the following equation (3.2):

$$Merit_{S_k} = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}} \tag{3.2}$$

where $\overline{\boldsymbol{r_{cf}}}$ denotes the average degree of correlation between a feature *f* and the class variable *c* (averaged over all features in *S*); and $\overline{\boldsymbol{r_{ff}}}$ denotes the average degree of correlation between a pair of features (averaged over all feature pairs in *S*). As discussed in Section 2.2.3, these variables represent the average degrees of feature relevance (for predicting the class) and redundancy between features, respectively.

Note that here the term "correlation" is used in a broad sense, since the CFS version in WEKA, which was used as the basis for our CFS variants, actually measures the correlation between variables using symmetrical uncertainty, rather than Pearson's correlation coefficient. We proceed using the term "*r*" to denote correlation anyway, to be consistent with the literature on CFS.

The core idea of the two CFS variants proposed in this section is to modify how the average degree of redundancy between features in *S* is computed, recognizing that in our context of longitudinal data there are two types of feature redundancy, as follows.

The first redundancy type is the redundancy between the features within the same group of temporally related features, here called temporal redundancy for short. Recall that all features within a group refer to the same conceptual feature (see Section 3.1.3). That is, all features within the same group represent either directly observed values of the same conceptual feature at different time points (waves), in the case of original features, or values calculated for the same conceptual feature considering different ways of representing trends in that feature's value across waves, in the case of the constructed longitudinal features.

The second redundancy type is the redundancy between features belonging to different groups of features. Recall that this redundancy type refers to the redundancy between different conceptual features (unlike the first redundancy type), and is the standard type of feature redundancy considered by CFS and other feature selection algorithms that detect feature redundancy in standard (non-longitudinal) classification datasets.

Note that standard CFS ignores the differences between these two types of feature redundancy; it simply treats all features in the same way, without dividing them into temporally related groups, and without considering the temporal nature of the data. That is, standard CFS would ignore the fact that different features take values measured at or calculated from different time points.

In the two CFS variants proposed in this Section, we assume that the temporal redundancy between a pair of features within the same feature group should be penalized more than the standard (non-temporal) redundancy between a pair of features across different groups. One motivation for this is the fact that, assuming that the classification models or the selected features will be interpreted by users, the selection of two or more features within the same group would represent intuitively redundant and potentially somewhat confusing information for users. For example, if a feature representing the value of a certain type of blood test in wave 6 is selected as a relevant feature for predicting a disease, users may think the values of the same base feature in wave 2 or 4 should be considered intuitively redundant information. By contrast, selecting two different base features, like two different types of blood tests, would not so easily be considered as redundant information by users, as long as the two tests are measuring quite different biomedical properties (which is in general the case in our datasets).

Therefore, our new redundancy equations assign different weight values to a pair of features, depending on whether the two features are in the same group of temporally related features or in different groups. We call this a weighted redundancy (WR) approach, and we call a variant of CFS using this approach WR-CFS. We propose two different versions of this WR approach to define these weights when computing the term $\overline{\boldsymbol{r_{ff}}}$ (the average redundancy over all feature pairs in *S*), producing two variants of WR-CFS, as follows.

The first proposed variant of WR-CFS is called Coarse-grained Weighted Redundancy CFS (Co-WR-CFS), since it is based on a coarse-grained assignment of weights to each type of redundancy, as explained next. The basic idea of Co-WR-CFS is to replace the equation used for computing $\overline{\boldsymbol{r_{ff}}}$ from equation (2.13) – the equation used by standard CFS, where *fp* is the number of feature pairs in *S* – by equation (3.3).

$$\overline{r_{ff}} = \frac{\sum_{i=1}^{k} \sum_{j=1}^{i-1} r_{f_i f_j}}{fp} \tag{3.3}$$

$$\overline{r_{ff}} = w_{wg} \frac{\sum r_{wg}}{fp_{wg}} + w_{ag} \frac{\sum r_{ag}}{fp_{ag}} \tag{3.4}$$

Equation (3.4) splits the computation of the average degree of redundancy into two terms, where r_{wg} and r_{ag} are the correlations between a pair of features within the same group and a pair of features across different groups, respectively, and the summation is over all corresponding feature pairs. Analogously, fp_{wg} and fp_{ag} denote the number of feature pairs within the group and across groups ($fp = fp_{wg} + fp_{ag}$). Hence, after calculating the averages value of redundancy for both types of feature pairs, we weight each of them based on the pre-defined weight values w_{wg} and w_{ag}. Note that these weights are assigned in a coarse-grained way of the average degree of redundancy computed over the corresponding number of features pairs, which is in contrast to the second version of WR-CFS, proposed later. In addition, these weight values have two constraints, where $w_{wg} + w_{ag} = 1.0$ for normalisation purpose and $w_{wg} \geq w_{ag}$ according to the rationale mentioned above.

Note also that, when computing equation (3.4), the number of feature pairs within the group (fp_{wg}) is in general smaller than the number of feature pairs across the groups (fp_{ag}). As a result, recalling that $w_{wg} \geq w_{ag}$, the calculation of the overall degree of redundancy by equation (3.4) over-emphasizes the degree of redundancy associated with the features in the same group, which tends to be a minority of the feature pairs in S. That is, in general, each pair of features within the group will have a greater influence in the value of $\overline{\boldsymbol{r_{ff}}}$ computed by equation (3.4) than each pair of features across groups, simply because there will be typically fewer pairs of the first type.

For instance, suppose that in the candidate feature subset *S* there is just one feature pair within the group, and that pair has a degree of correlation of 1, but there are 15 feature pairs across groups with an average degree of correlation of 0.1. Consider the case where w_{wg} is 0.7 and w_{ag} is 0.3. Using equation (3.4) results in:

Degree of redundancy = 0.7 × 1 + 0.3 × 0.1 = 0.73

In order to mitigate the risk that WR-CFS over-emphasizes the degree of redundancy associated with (usually) a minority of feature pairs, we introduce the second version of WR-CFS, called Fine-grained Weighted Redundancy CFS (Fi-WR-CFS). This version still uses weights subject to the constraints $w_{wg} + w_{ag} = 1.0$ and $w_{wg} \geq w_{ag}$, but it applies the weights to each pair of features separately, in a fine-grained fashion, instead of applying the weights just once to the average values of each type of redundancy in a coarse-grained fashion, as in the first version.

Before specifying the Merit formula for Fi-WR-CFS, first note that equation (3.2) is a simplification of the following equation (3.5) (Hall 1999).

$$\boldsymbol{Merit_{S_k} = \frac{\sum_{i=1}^{k} r_{cf_i}}{\sqrt{k + 2 * \sum_{i=1}^{k}\sum_{j=1}^{i-1} r_{f_i f_j}}}} \tag{3.5}$$

Hence, in the proposed Fi-WR-CFS, the Merit is computed as shown in equation (3.6).

$$Merit_{S_k} = \frac{\sum_{i=1}^{k} r_{cf_i}}{\sqrt{k + 2 * \sum_{i=1}^{k}\sum_{j=1}^{i-1} w_{f_i f_j} * r_{f_i f_j}}} \tag{3.6}$$

where $w_{f_i f_j} = \begin{cases} w_{wg}, & if\ f_i\ and\ f_j\ are\ in\ the\ same\ group \\ w_{ag}, & otherwise \end{cases}$. Alternatively, this can be simplified to equation (3.7),

$$Merit_{S_k} = \frac{\sum_{i=1}^{k} r_{cf_i}}{\sqrt{k + 2 * (w_{wg} * \sum r_{wg} + w_{ag} * \sum r_{ag})}} \tag{3.7}$$

where the summations in the denominator are over the corresponding feature pairs.

Note that equation (3.7) in general de-emphasizes the relative importance of temporal redundancy (versus standard redundancy) in the computation of the Merit function, by comparison with equation (3.4) – used by Co-WR-CFS.

This is because, although equation (3.7) still uses redundancy weights satisfying the constraint $w_{wg} \geq w_{ag}$ (i.e., it assigns a greater weight to temporal redundancy than to standard redundancy), in general the value of the summation $\sum r_{ag}$ may well be greater than the value of the summation $\sum r_{wg}$, because in general there are more pairs of features across groups than pairs of features within the same group, as mentioned earlier. Hence, the relative importance of temporal redundancy (versus standard redundancy) tends to be smaller in equation (3.7), by comparison with in equation (3.4).

In the discussion so far, we referred to the weights w_{wg} w_{ag} in a general way, without discussing how to specify their precise values. We now turn to this issue.

Setting the parameter w_{wg} (or its complement w_{ag}) is not trivial. A common approach to optimize parameters in classification is to use an internal cross-validation procedure (accessing the training set only) to evaluate a set of pre-defined candidate parameter values (w_{wg} values in our case), and then choose the parameter value with the highest predictive accuracy in that internal cross-validation procedure. However, this approach normally assumes the use of a classification algorithm to measure predictive accuracy in the internal cross-validation, and in our case CFS is used as a filter feature selection method (rather than a wrapper one), without running a classification algorithm. Using the Merit function of CFS to evaluate the predictive performance of a set of w_{wg} values via internal cross-validation would not be a fair approach to compare the performance of different w_{wg} values, since different w_{wg} values would effectively implement different evaluation functions. For instance, it is possible that a certain feature subset optimizes the Merit function when, say, w_{wg} = 0.6, whilst another feature subset optimizes the Merit function when, say, w_{wg} = 0.8;

and none of those two feature subsets can be considered the "optimal" feature subset in general, since both were selected based on a Merit function with just a specific value of w_{wg}.

Hence, to mitigate the problem that the selection of the best feature subset depends on the value of w_{wg}, and so to make the feature selection procedure more robust, we propose a method that outputs a set of select features that is obtained by merging the feature subsets selected by using a set of different w_{wg} values – both when using equation (3.4) and when using equation equation (3.7). In particular, we consider the feature subsets selected using five different w_{wg} values, namely w_{wg} = 0.5, 0.6, 0.7, 0.8 and 0.9.

Different merging approaches are possible. In an extreme case, the most inclusive approach would be output the set union of the five selected feature subsets, i.e. to output any feature that was included in any of the five selected feature subsets (for the above five w_{wg} values). However, this would tend to output several features which are not robust (e.g., features selected only for one particular w_{wg} value).

In the opposite extreme case, the strictest (least inclusive) approach would be to output the set intersection of the five feature subsets, i.e., to output only the features that were included in all the five selected feature subsets. However, this would tend to output a small set of features, not outputting features which are robust in general (e.g. features selected for four out of the five w_{wg} values).

As a compromise between these two extreme approaches, in order to get reasonably robust features, we propose to output the features included in at least three of the five selected feature subsets. That is, we run both versions of WR-CFS on the training set five times, with w_{wg} = 0.5, 0.6, 0.7, 0.8 and 0.9; and then output as the set of selected features all features that were selected in at least three of those five runs. Then, the classification algorithm is applied to the reduced training set containing only those selected features in order to build a classification model, and finally the predictive performance of that model is evaluated on the test set.

4 COMPUTATIONAL RESULTS

This chapter provides information about the experimental methodology and analysis of the computational results. This chapter is organised as follows. Section 4.1 explains the experimental methodology used such as cross-validation, the predictive accuracy measure and statistical tests used, etc. Section 4.2 reports the computational results obtained by J48 and the CFS variants proposed in Chapter 3. Section 4.3 reports the analogous results obtained by Naïve Bayes. Each of these two sections reports the predictive accuracy obtained by the corresponding classification algorithm, with and without the proposed CFS variants, as well as the number of features selected by each CFS variant.

A relatively small part of the results reported in this chapter has been included in a peer-reviewed paper that is currently in press (Pomsuwan and Freitas 2017).

4.1 Experimental Methodology

We report results for 10 base datasets created from the raw data in the ELSA database, as described in Chapter 3. Each of these base datasets will be used to produce different datasets varying the class distribution, as will be explained later. Recall that each dataset has a different age-related disease in wave 7 as the class variable to be predicted, whilst all datasets have the same predictive features (derived in general from waves 2, 4, and 6).

Predictive accuracy was measured by the well-known F-measure, the harmonic mean between Precision and Recall (Japkowicz and Shah 2011), given by equation (4.1),

$$F\text{-}measure = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{4.1}$$

where Precision is the proportion of instances predicted as positive which are really positive and Recall is the proportion of positive instances that were correctly predicted as positive. To compute these measures, each class label (presence or absence of the disease) was considered in turn as the positive class and the reported F-measure is the arithmetic (unweighted) mean of the F-measures for the two class labels.

Each of the proposed variants of the CFS feature selection method was evaluated using two classification algorithms, namely Naïve Bayes and the decision tree induction algorithm J48 – both reviewed in Chapter 2.

All experiments were performed using the WEKA data mining tool (Hall *et al.* 2009), version:3.8.1 and all reported results were obtained by running a well-known 10-fold cross-validation procedure, which works as follows. First, the dataset is randomly partitioned into 10 folds with approximately the same number of instances in each fold. Then, each feature selection method and each classification algorithm is run 10 times, each time using a different fold as the test set and all the other 9 folds as the training set. The reported measure of predictive accuracy is then the average accuracy across the 10 test sets.

4.1.1 Statistical Tests

In this section, all statistical significance tests used to analyse our experimental results are described. We focus on pairwise comparisons for classification models constructed from features selected by different feature selection approaches.

4.1.1.1 The Wilcoxon Signed-Ranks Test

The Wilcoxon signed-ranks test (Wilcoxon 1945) is a non-parametric statistical significance test used in this thesis for comparing the predictive accuracies (more precisely, F-measure values) of two classification models. The main advantage of this test is its non-parametric nature, making no assumption of normal distribution (Japkowicz and Shah 2011), which is a strong assumption made in particular by the alternative paired t-test. Another advantage of the Wilcoxon signed-ranks test is its robustness against outliers, since it is based on the relative ranks of the predictive performances of two models, instead of being based on their raw performance such as the raw F-measure values.

The null-hypothesis for this test is that the medians of two classification models' predictive performances are equal.

There are several steps involved in the Wilcoxon signed-ranks test (Demšar 2006), as follows. To begin, the difference (d_i) between the predictive accuracy of the two classification models is calculated for each i-th dataset, i = 1,…,N, where N is the number of datasets. Next, the differences are ranked according to their absolute values (rank(d_i), i = 1,…,N,), ignoring their signs; in the case of a tie, the corresponding average rank is assigned. Once the data have been prepared, we start to calculate the Wilcoxon signed rank sums. The calculations proceed separately according to equation (4.2) and (4.3) for the positive and negative differences of accuracy, respectively. That is, R^+ denotes the sum of ranks for positive differences and R^- denotes the sum of ranks of negative differences. It should be noted that the differences of 0 have their ranks split evenly among the sums; if there is an odd number of them, one is discarded.

$$R^+ = \sum_{d_i>0} rank(d_i) + \frac{1}{2}\sum_{d_i=0} rank(d_i) \tag{4.2}$$

$$R^- = \sum_{d_i<0} rank(d_i) + \frac{1}{2}\sum_{d_i=0} rank(d_i) \tag{4.3}$$

Afterwards, the smaller of R^+ or R^- is used for the test statistic, T. Let T and $T_{critical}$ be the smaller of the rank sums and the exact critical value respectively. The null hypothesis is rejected if T is greater than or equal to $T_{critical}$, accepted otherwise. In general, the exact value of $T_{critical}$ can be found in a precomputed table (available e.g. in (Bruning and Kintz 1987)) for values of N up to 25. For a larger number of datasets, the distribution of the test statistic can be approximated by a normal distribution, with the following equation for calculating the z-score:

$$z = \frac{T - \frac{1}{4}N(N+1)}{\sqrt{\frac{1}{24}N(N+1)(2N+1)}} \tag{4.4}$$

where T is the test statistic and N is the number of datasets. Subsequently, the null-hypothesis is rejected if z is smaller than the critical value for z-score.

4.1.1.2 Friedman's Test

The Friedman's test is a non-parametric test for determining whether or not there are significant differences in the performance of multiple classification models across multiple datasets (Friedman 1940). Non-parametric means that the test makes no assumption about the dataset having a particular distribution, e.g., the normal distribution. The null hypothesis for the test is that all the classification models have identical predictive accuracy. The alternative hypothesis is that the classification models have different predictive accuracies.

The Friedman's test has six main steps: the first three are involved in data preparation and the rest involve running the test on prepared data. First of all, it ranks the predictive accuracy values of the classification models being compared for each dataset (row) separately. That is, the model with the highest predictive accuracy is assigned a rank of 1. In the case of a tie, the corresponding average rank is assigned to the tied models. Afterwards, the ranks are averaged for each classification model. The next step is to calculate the Friedman's test statistic (χ_F^2) as follows:

$$\chi_F^2 = \frac{12N}{k(k+1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right] \tag{4.5}$$

where N is the number of datasets, k is the number of classification models and R_j is the average ranks of the models being compared. However, as pointed out in (Iman and Davenport 1980), a better statistic can be derived, as shown in equation (4.6),

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2} \tag{4.6}$$

which is distributed according to the F-distribution with (k-1) and (k-1) (N-1) degrees of freedom. If the null hypothesis is rejected, it means that there is a significant difference in predictive performance among the classification models. In this case, we need to apply a post-hoc test to point out which pairs of the models have significantly different performances.

4.1.1.3 The Nemenyi Test

The Nemenyi test is a post-hoc and non-parametric test for determining whether or not there are significant differences in the predictive performance of each pair of classification models across multiple datasets (Nemenyi 1962). Specifically, a post-hoc test is applied if and only if a pre-hoc test (like the Friedman's test) has determined a significant difference. Non-parametric means that the test makes no assumption about the dataset having a particular distribution, e.g., the normal distribution. In addition, the Nemenyi test is a multiple hypothesis test where the number of null hypotheses is the same as the number of pairwise comparisons. The result of the test is determined by the critical difference (CD), computed as shown in equation (4.7),

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}} \tag{4.7}$$

such that if the average rank difference is greater than CD, the null hypothesis is rejected. Note that α is the significance level, and the critical values q_α are based on the Studentized range statistic divided by $\sqrt{2}$. More precisely, given that two classification models have average ranks R_i and R_j, the null hypothesis of the test is that their performance is equivalent. In order to reject the null hypothesis, the difference between R_i and R_j has to be greater than the CD.

4.2 Results for the J48 Decision Tree Induction Algorithm

This section reports the results obtained by the J48 classification algorithm (reviewed in Chapter 2) with different variants of the CFS method used for selecting features in a data pre-processing phase. All these CFS variants were discussed in Chapter 3. This section consists of three subsections. The first one reports results comparing the Exhaustive search-based CFS per Group (Exh-CFS-Gr) method and its extension, namely Exh-CFS-Gr followed by the use of standard CFS (Exh-CFS-Gr+CFS). The second subsection reports results comparing the two Weighted-Redundancy CFS variants, namely Coarse-grained Weighted-Redundancy CFS (Co-WR-CFS) and Fine-grained Weighted-Redundancy CFS (Fi-WR-CFS). Finally, in the third subsection the two best proposed CFS variants (one from each of the first two subsections) are compared with two baseline methods, namely the standard CFS and no feature selection in a pre-processing phase (i.e. passing all features to J48). Furthermore, each analysis includes the variations of class distributions, counting wins and losses, and a suitable statistical test.

In addition, some statistical analysis methods mentioned in Section 4.1.1 were used for validation of the published results. In statistical terms, we attempt to reject the null hypothesis that the classification models with a given feature selection method obtains F-measure values that are the same as the F-measure values obtained with another feature selection method. We used the tests with a significance level of $\alpha = 0.05$, and $N = 35$ (35 datasets) in our experiments.

4.2.1 Results comparing the two proposed CFS variants based on exhaustive search

Table 4.1 reports the F-measure values obtained by J48, after applying the proposed Exh-CFS-Gr and Exh-CFS-Gr+CFS methods in a pre-processing phase, broken down by each dataset – recall that each dataset involves a combination of a disease used as a class variable in wave 7 and a certain class distribution. Note that, for each base dataset associated with a given disease (class variable), several datasets where created by undersampling the majority class, to reduce class imbalance. These produced datasets with different class distributions – varying from the original (very imbalanced) class distribution to the balanced class distribution (with a 1 to 1 ratio between the frequencies of both classes). Each cell of the class distribution column has values in the format "X to Y", where X and Y denote the number of instances belonging to the negative and positive class labels, respectively.

Note also that Table 4.1 reports, for each dataset, only the mean F-measure value over the two class labels, since this is the measure used in the statistical significance analysis, but the detailed values of the precision, recall and F-measure for each class label can be found in the Appendix, Section A. The best F-measure value(s) in each row of the Table – i.e., for each dataset – among the two CFS variants is shown in boldface, and the highest F-measure value(s) for each age-related disease, across all class distributions, is highlighted with an underline.

We also compute the average rank of each of the two CFS variants in the Table, as follows. For each dataset, the CFS variant with the higher (lower) F-measure value is assigned rank 1 (2), and in case of a tie, each CFS variant is assigned the rank 1.5. The average rank of a CFS variant is simply the mean of its rank over the 35 datasets. Hence, the lower its average rank, the better the CFS variant is.

With regard to the results across variations of class distributions, in general the more balanced the class distribution is, the higher the F-measure values achieved by both CFS variants. Specifically, the highest F-measure values were obtained when the class distribution was completely balanced (1 to 1 ratio) in 18 out of 20 cases (10 diseases × 2 CFS variants). The only two exceptions were observed in the results for the class Diabetes, where the completely balanced class distribution yielded the lowest F-measure value for both Exh-CFS-Gr and Exh-CFS-Gr+CFS.

In Table 4.1, it can be seen that Exh-CFS-Gr+CFS obtained slightly better F-measure values than Exh-CFS-Gr on 15 out of the 35 datasets, with 11 losses and 9 ties. In addition, the average rank of Exh-CFS-Gr+CFS (1.44) is lower (better) than that of Exh-CFS-Gr (1.56). In spite of the greater number of wins, by applying the Wilcoxon signed-ranks test the null hypothesis cannot be rejected, with a p-value of 0.39. Hence, the statistical evidence is insufficient to claim that Exh-CFS-Gr+CFS performs better than Exh-CFS-Gr when J48 is used as the classification algorithm.

With regard to the size of the selected feature subsets, Table 4.2 reports the average number and percentage of features selected by each of the two CFS variants (across the 10 cross-validation folds), for each dataset – i.e. each combination of disease and class distribution. Note that the number of features selected by Exh-CFS-Gr is substantially greater than the number of features selected by Exh-CFS-Gr+CFS for all datasets. More precisely, on average across the 35 datasets, Exh-CFS-Gr and Exh-CFS-Gr+CFS select 34.51% and 6.46% of the original features, respectively. This is because, once Exh-CFS-Gr selects a feature subset, the standard CFS is further applied as the second phase of the feature selection process performed by Exh-CFS-Gr+CFS method. Naturally, that second phase tends to remove many features which were previously selected by Exh-CFS-Gr in the first phase. In particular, Exh-CFS-Gr only takes into account the redundancy among the features within the same conceptual group, as mentioned in Section 3.2. Therefore, in the second phase of Exh-CFS-Gr+CFS, the standard CFS tends to eliminate many redundant features across the groups, which results in a clearly smaller feature subset than the subset produced by Exh-CFS-Gr alone.

Although the numbers of features selected by the two CFS variants were very different, the predictive accuracy obtained by J48 was statistically equivalent in both cases, as discussed above. This can be explained by the fact that J48 performs embedded feature selection. Hence, when Exh-CFS-Gr returns a relatively large feature subset containing features with redundancy across groups, J48 can use its embedded feature selection procedure to focus on the most relevant variables.

Moreover, a further analysis of the results has been conducted by using the Pearson's linear correlation coefficient (r) to measure the degree of correlation between the predictive accuracy and the number of selected features for the two feature selection methods compared in this Section, across the 35 datasets. As for Exh-CFS-Gr, the value of r is 0.546 which indicates a moderate positive correlation. Therefore, there is a broad tendency for high predictive accuracies to be obtained with larger feature subset sizes. In contrast, the value of r for Exh-CFS-Gr+CFS is –0.171, which technically indicates a weak negative correlation. Hence, the correlation between the predictive accuracy and the number of features selected by Exh-CFS-Gr+CFS is weak.

Table 4.1: F-measure values obtained by J48 after applying the two CFS variants based on exhaustive search.

Disease	Class Distribution	Exh-CFS-Gr+J48	Exh-CFS-Gr+CFS+J48
HeartAtt	7061 to 435	**0.485**	**0.485**
	1740 to 435	0.565	**0.572**
	870 to 435	**0.589**	0.582
	435 to 435	0.604	**0.610**
Angina	7263 to 233	**0.492**	**0.492**
	932 to 233	0.502	**0.503**
	466 to 233	**0.534**	0.527
	233 to 233	**0.546**	0.539
Stroke	7094 to 402	**0.486**	**0.486**
	1608 to 402	**0.554**	0.551
	804 to 402	**0.594**	0.587
	402 to 402	**0.604**	0.594
Diabetes	6552 to 944	**0.804**	0.799
	3776 to 944	0.800	**0.801**
	1888 to 944	0.786	**0.792**
	944 to 944	0.747	**0.753**
HBP	4438 to 3058	0.652	**0.655**
	3058 to 3058	**0.669**	0.662
Dementia	7360 to 136	**0.495**	**0.495**
	544 to 136	0.541	**0.557**
	272 to 136	0.577	**0.584**
	136 to 136	0.582	**0.592**
Cataract	5344 to 2150	0.632	**0.646**
	4300 to 2150	**0.651**	**0.651**
	2150 to 2150	**0.673**	0.670
Arthritis	4398 to 3098	**0.612**	**0.612**
	3098 to 3098	0.612	**0.617**
Osteoporosis	6796 to 700	**0.476**	**0.476**
	2800 to 700	**0.555**	**0.555**
	1400 to 700	0.590	**0.603**
	700 to 700	**0.617**	0.614
Parkinsons	7433 to 63	**0.498**	**0.498**
	252 to 63	0.495	**0.522**
	126 to 63	**0.572**	0.570
	63 to 63	0.581	**0.584**

Table 4.2: Average numbers and percentages of selected features for the two CFS variants based on exhaustive search.

Disease	Class Distribution	Exh-CFS-Gr	Exh-CFS-Gr+CFS
HeartAtt	7061 to 435	80.0 (36.53%)	24.0 (10.96%)
	1740 to 435	76.7 (35.02%)	21.1 (9.63%)
	870 to 435	70.0 (31.96%)	19.7 (9.00%)
	435 to 435	65.7 (30.00%)	19.5 (8.90%)
Angina	7263 to 233	75.7 (34.57%)	14.0 (6.39%)
	932 to 233	74.7 (34.11%)	22.7 (10.37%)
	466 to 233	70.3 (32.10%)	23.4 (10.68%)
	233 to 233	68.3 (31.19%)	19.3 (8.81%)
Stroke	7094 to 402	77.6 (35.43%)	7.6 (3.47%)
	1608 to 402	75.4 (34.43%)	7.9 (3.61%)
	804 to 402	70.5 (32.19%)	9.0 (4.11%)
	402 to 402	64.6 (29.50%)	12.4 (5.66%)
Diabetes	6552 to 944	98.3 (44.89%)	10.1 (4.61%)
	3776 to 944	96.5 (44.06%)	11.7 (5.34%)
	1888 to 944	93.3 (42.60%)	11.4 (5.21%)
	944 to 944	86.3 (39.41%)	11.0 (5.02%)
HBP	4438 to 3058	98.2 (44.84%)	25.4 (11.60%)
	3058 to 3058	95.3 (43.52%)	24.3 (11.10%)
Dementia	7360 to 136	68.2 (31.14%)	11.8 (5.39%)
	544 to 136	70.7 (32.28%)	14.9 (6.80%)
	272 to 136	71.6 (32.69%)	16.5 (7.53%)
	136 to 136	66.6 (30.41%)	13.4 (6.12%)
Cataract	5344 to 2150	73.2 (33.42%)	7.1 (3.24%)
	4300 to 2150	71.9 (32.83%)	6.6 (3.01%)
	2150 to 2150	69.7 (31.83%)	6.3 (2.88%)
Arthritis	4398 to 3098	80.2 (36.62%)	15.0 (6.85%)
	3098 to 3098	75.5 (34.47%)	18.2 (8.31%)
Osteoporosis	6796 to 700	89.6 (40.91%)	15.6 (7.12%)
	2800 to 700	85.7 (39.13%)	17.5 (7.99%)
	1400 to 700	82.6 (37.72%)	13.9 (6.35%)
	700 to 700	78.7 (35.94%)	13.4 (6.12%)
Parkinsons	7433 to 63	55.9 (25.53%)	7.9 (3.61%)
	252 to 63	57.6 (26.30%)	7.5 (3.42%)
	126 to 63	56.1 (25.62%)	8.0 (3.65%)
	63 to 63	54.2 (24.75%)	7.0 (3.20%)

4.2.2 Results comparing the two proposed CFS variants based on different weights for two types of redundancy

Table 4.3 reports the F-measure values for the pair of proposed CFS variants based on using different weights for the two types of redundany, namely temporal redundancy (among features within the same group of temporally redundant features) and standard (non-temporal) redundancy (among features in different groups), where both CFS variants are run before applying J48 algorithm. Recall that these CFS variants are called Coarse-grained Weighted Redundancy (Co-WR-CFS) and Fine-Grained Weighted Redundancy (Fi-WR-CFS). The Table reports the results for the same 35 datasets used in Table 4.1.

Similarly to the results reported in Table 4.1, overall the F-measure values in Table 4.3 increase for both CFS variants when the class distributions are increasingly more balanced. The only two exceptions are for the diseases Diabetes and Arthritis.

Regarding the predictive performance of J48 when using the two CFS variants in a pre-processing phase, Co-WR-CFS and Fi-WR-CFS achieved a similar number of wins, 13 and 12 respectively, with 10 ties. Furthermore, running the Wilcoxon signed-ranks test also suggested an equivalence in performance, where the null hypothesis is accepted with the p-value of 0.81. In other words, there is no significant difference between Co-WR-CFS and Fi-WR-CFS when they are used with the J48 algorithm.

With regard to the size of the selected feature subsets, Table 4.4 shows the average number and percentage of features selected by each of the two CFS variants based on weighted redundancy (across the 10 cross-validation folds), for each dataset – i.e., each combination of disease and class distribution. The number of features selected by Fi-WR-CFS is greater than the number of features selected by Co-WR-CFS in general. More precisely, on average across the 35 datasets, Fi-WR-CFS and Co-WR-CFS select 9.46% and 5.80% of the original features, respectively.

Nevertheless, the predictive accuracies obtained by J48 when using these two WR CFS variants were statistically equivalent, as discussed above. Again, this can be explained by the fact that J48 performs embedded feature selection.

Moreover, by measuring the Pearson's linear correlation coefficient between predictive accuracy and number of selected features across the 35 datasets, the values of *r* are –0.189 and 0.002 for Co-WR-CFS and Fi-WR-CFS, respectively. Hence, the correlation between the predictive accuracy and the number of selected features is weak for Co-WR-CFS and practically none for Fi-WR-CFS.

Table 4.3: F-measure values obtained by J48 after applying the two CFS variants based on weighted redundancy.

Disease	Class Distribution	Co-WR-CFS+J48	Fi-WR-CFS+J48
HeartAtt	7061 to 435	**0.485**	**0.485**
	1740 to 435	0.560	**0.576**
	870 to 435	**0.593**	0.578
	435 to 435	0.603	**0.608**
Angina	7263 to 233	**0.492**	**0.492**
	932 to 233	0.508	**0.509**
	466 to 233	**0.539**	0.536
	233 to 233	0.546	**0.549**
Stroke	7094 to 402	**0.486**	**0.486**
	1608 to 402	**0.566**	0.556
	804 to 402	0.580	**0.581**
	402 to 402	**0.611**	0.607
Diabetes	6552 to 944	**0.798**	0.797
	3776 to 944	**0.799**	0.797
	1888 to 944	0.793	**0.802**
	944 to 944	0.738	**0.763**
HBP	4438 to 3058	0.635	**0.661**
	3058 to 3058	0.653	**0.662**
Dementia	7360 to 136	**0.495**	**0.495**
	544 to 136	**0.559**	0.547
	272 to 136	**0.588**	0.585
	136 to 136	**0.592**	0.587
Cataract	5344 to 2150	**0.651**	0.647
	4300 to 2150	0.654	**0.657**
	2150 to 2150	**0.671**	**0.671**
Arthritis	4398 to 3098	**0.618**	0.616
	3098 to 3098	**0.614**	0.611
Osteoporosis	6796 to 700	**0.476**	**0.476**
	2800 to 700	0.550	**0.555**
	1400 to 700	0.590	**0.608**
	700 to 700	**0.613**	**0.613**
Parkinsons	7433 to 63	**0.498**	**0.498**
	252 to 63	**0.521**	**0.521**
	126 to 63	**0.570**	**0.570**
	63 to 63	**0.583**	0.582

Table 4.4: Average numbers and percentages of selected features for the two CFS variants based on weighted redundancy.

Disease	Class Distribution	Co-WR-CFS	Fi-WR-CFS
HeartAtt	7061 to 435	15.8 (7.21%)	29.3 (13.38%)
	1740 to 435	15.8 (7.21%)	27.3 (12.47%)
	870 to 435	16.4 (7.49%)	26.4 (12.05%)
	435 to 435	16.9 (7.72%)	25.8 (11.78%)
Angina	7263 to 233	11.9 (5.43%)	19.5 (8.90%)
	932 to 233	19.1 (8.72%)	31.0 (14.16%)
	466 to 233	18.1 (8.26%)	29.8 (13.61%)
	233 to 233	16.7 (7.63%)	25.8 (11.78%)
Stroke	7094 to 402	7.9 (3.61%)	8.3 (3.79%)
	1608 to 402	11.4 (5.21%)	21.1 (9.63%)
	804 to 402	13.5 (6.16%)	23.5 (10.73%)
	402 to 402	14.7 (6.71%)	20.9 (9.54%)
Diabetes	6552 to 944	10.5 (4.79%)	15.4 (7.03%)
	3776 to 944	11.4 (5.21%)	19.7 (9.00%)
	1888 to 944	10.7 (4.89%)	23.9 (10.91%)
	944 to 944	9.6 (4.38%)	25.0 (11.42%)
HBP	4438 to 3058	16.5 (7.53%)	34.1 (15.57%)
	3058 to 3058	17.7 (8.08%)	31.2 (14.25%)
Dementia	7360 to 136	11.5 (5.25%)	17.1 (7.81%)
	544 to 136	16.3 (7.44%)	23.2 (10.59%)
	272 to 136	15.0 (6.85%)	24.7 (11.28%)
	136 to 136	14.1 (6.44%)	21.4 (9.77%)
Cataract	5344 to 2150	8.3 (3.79%)	8.4 (3.84%)
	4300 to 2150	7.8 (3.56%)	8.2 (3.74%)
	2150 to 2150	6.9 (3.15%)	8.1 (3.70%)
Arthritis	4398 to 3098	13.1 (5.98%)	25.1 (11.46%)
	3098 to 3098	13.3 (6.07%)	24.2 (11.05%)
Osteoporosis	6796 to 700	13.1 (5.98%)	26.9 (12.28%)
	2800 to 700	13.4 (6.12%)	24.7 (11.28%)
	1400 to 700	13.2 (6.03%)	22.4 (10.23%)
	700 to 700	12.6 (5.75%)	20.1 (9.18%)
Parkinsons	7433 to 63	7.6 (3.47%)	8.6 (3.93%)
	252 to 63	7.8 (3.56%)	7.8 (3.56%)
	126 to 63	8.2 (3.74%)	8.2 (3.74%)
	63 to 63	7.6 (3.47%)	7.7 (3.52%)

4.2.3 Results comparing four feature selection approaches using J48

This section compares the results obtained by the J48 decision tree algorithm when using four feature selection approaches: comparing two proposed CFS variants – one based on exhaustive search (Exh-CFS-Gr) and the other based on weighted redundancy (Co-WR-CFS) – against the standard CFS method and the baseline approach of giving all input features to J48 – i.e., no feature selection in a pre-processing step. In this section, Exh-CFS-Gr+CFS and Co-WR-CFS were chosen as representatives of proposed CFS variants due to their better performance shown in Subsections 4.2.1 and 4.2.2 respectively.

Accordingly, Table 4.5 reports the F-measure values for the above four CFS variants. We began by calculating the average rank for each variant, as described earlier – recall that the smaller the average rank, the better (higher) the F-measure value of a CFS variant across all datasets in general. The best average rank was jointly obtained by standard CFS and Exh-CFS-Gr+CFS (average rank 2.41), closely followed by Co-MR-CFS (rank 2.49) and no feature selection, i.e. J48 alone (rank 2.69).

Next, the Friedman test was used to determine whether or not there is a significant difference between the average rank of each CFS variant and their mean rank (2.5). The calculated value of F_F is 0.45. With four feature selection approaches and 35 datasets, F_F is distributed according to the F distribution with 4 – 1 = 3 and (4-1) x (35-1) = 102 degrees of freedom. The critical value of $F(3,102)$ for $\alpha = 0.05$ is 2.69. Since F_F (0.45) is smaller than the critical value (2.69), the null-hypothesis cannot be rejected. Hence, there is insufficient statistical evidence against the claim that all the four feature selection approaches obtained equivalent F-measure values.

With regard to the size of the selected feature subsets, Table 4.6 shows the average number and percentage of features selected by each of the three CFS variants (across the 10 cross-validation folds), for each dataset – i.e., each combination of disease and class distribution. The number of features selected by standard CFS and that selected by Exh-CFS-Gr+CFS are almost the same for all datasets.

In order to investigate whether there is a correlation between the predictive accuracy and the number of features selected by each method, we computed the Pearson’s linear correlation coefficient between those two variables across the 35 datasets, for each of the three feature selection methods whose performance is analysed in this section. The computed correlation coefficients are –0.127 for standard CFS and –0.171 for Exh-CFS-Gr+CFS, both indicating a weak negative correlation. Unlike these two methods, Co-WR-CFS selected relatively smaller subsets, in general, with a correlation coefficient of 0.189, indicating a weak positive correlation.

Overall, there is very little correlation between the predictive accuracy and the number of features selected by each of the three CFS variants, despite the fact that all datasets use the same set of original features before applying feature selection. This seems to be due to the fact that, although the features are the same, the datasets have different diseases used as classes, and this leads to a large variation in the predictive relationships between the features and the class variable across datasets. That is, the set of features (and its size) which is relevant for predicting the class variable varies greatly across datasets, as a result of very different diseases being used as classes.

Finally, Table 4.7 and Table 4.8 report the relative frequency (%) of selection for different feature types for the standard CFS and Exh-CFS-Gr+CFS, respectively. The column "General" refers to the age and gender features, the only two features not derived from the Nurse data. In each cell, the relative frequency was calculated as the frequency of selection summed for all features of the corresponding feature type over the maximum possible number of selections, which was the number of features in the feature type times 10 (considering the 10 folds of the cross- validation procedure).

Comparing the relative frequencies of the observed features in waves 2, 4, and 6 that were selected by the two CFS versions, for both the standard CFS (Table 4.7) and Exh-CFS-Gr+CFS (Table 4.8), the relative selection frequency increased monotonically with time (i.e. from wave 2 to wave 4 to wave 6) in 24 of the 35 datasets. The main exceptions were the datasets of Dementia and Parkinson's for all different distributions. This general predominance of selected features from wave 6 can be explained by the fact that features are selected partly based on their ability to predict an age-related disease at wave 7 and intuitively predictions based on features in wave 6 (shorter-term predictions) should be easier and more accurate than predictions based on features in waves 2 and 4 (longer-term predictions).

Regarding the constructed longitudinal features (CLFs), they were rarely selected by both standard CFS and Exh-CFS-Gr+CFS, as can be observed in the column “CLF Total” of Table 4.7 and Table 4.8. This is particularly the case for the feature types diff_w24, mono_w246, up_w24 and up_w26, whose selection frequencies are almost always 0% for both CFS variants. However, the features diff_w46 and diff_w26 were relatively more successful, and in the Dementia dataset, each of these two feature types was selected more often than the original features from wave 6 in two of the four datasets for that disease. These results also hold for both CFs variants. In the other datasets, however, diff_w46 and diff_w26 features were in general selected much less often than the original features from wave 6, the most recent wave.

Table 4.5: F-measure values obtained by J48 after applying different CFS methods

Disease	Class Distribution	J48	standard_CFS +J48	Exh-CFS-Gr+CFS +J48	Co-WR-CFS +J48
HeartAtt	7061 to 435	**0.485**	**0.485**	**0.485**	**0.485**
	1740 to 435	0.568	**0.572**	**0.572**	0.560
	870 to 435	**0.594**	0.579	0.582	0.593
	435 to 435	0.606	0.608	**0.610**	0.603
Angina	7263 to 233	**0.492**	**0.492**	**0.492**	**0.492**
	932 to 233	**0.514**	0.503	0.503	0.508
	466 to 233	0.532	0.525	0.527	**0.539**
	233 to 233	0.545	0.539	0.539	**0.546**
Stroke	7094 to 402	**0.486**	**0.486**	**0.486**	**0.486**
	1608 to 402	0.552	0.551	0.551	**0.566**
	804 to 402	0.572	0.586	**0.587**	0.580
	402 to 402	0.599	0.593	0.594	**0.611**
Diabetes	6552 to 944	**0.800**	0.799	0.799	0.798
	3776 to 944	**0.807**	0.799	0.801	0.799
	1888 to 944	0.798	**0.801**	0.792	0.793
	944 to 944	**0.770**	0.755	0.753	0.738
HBP	4438 to 3058	0.647	**0.657**	0.655	0.635
	3058 to 3058	0.660	0.661	**0.662**	0.653
Dementia	7360 to 136	**0.495**	**0.495**	**0.495**	**0.495**
	544 to 136	0.538	0.557	0.557	**0.559**
	272 to 136	0.584	0.584	0.584	**0.588**
	136 to 136	0.582	**0.592**	**0.592**	**0.592**
Cataract	5344 to 2150	0.631	0.647	0.646	**0.651**
	4300 to 2150	0.645	0.652	0.651	**0.654**
	2150 to 2150	**0.672**	0.670	0.670	0.671
Arthritis	4398 to 3098	0.612	0.617	0.612	**0.618**
	3098 to 3098	0.614	0.615	**0.617**	0.614
Osteoporosis	6796 to 700	**0.476**	**0.476**	**0.476**	**0.476**
	2800 to 700	0.536	**0.555**	**0.555**	0.550
	1400 to 700	0.586	**0.603**	**0.603**	0.590
	700 to 700	0.612	**0.614**	**0.614**	0.613
Parkinsons	7433 to 63	**0.498**	**0.498**	**0.498**	**0.498**
	252 to 63	0.519	**0.522**	**0.522**	0.521
	126 to 63	0.557	**0.570**	**0.570**	**0.570**
	63 to 63	**0.589**	0.584	0.584	0.583

Table 4.6: Average numbers and percentages of selected features for three CFS variants.

Disease	Class Distribution	standard_CFS	Exh-CFS-Gr+CFS	Co-WR-CFS
HeartAtt	7061 to 435	24.0 (10.96%)	24.0 (10.96%)	15.8 (7.21%)
	1740 to 435	21.3 (9.73%)	21.1 (9.63%)	15.8 (7.21%)
	870 to 435	20.3 (9.27%)	19.7 (9.00%)	16.4 (7.49%)
	435 to 435	20.2 (9.22%)	19.5 (8.90%)	16.9 (7.72%)
Angina	7263 to 233	14.1 (6.44%)	14.0 (6.39%)	11.9 (5.43%)
	932 to 233	22.8 (10.41%)	22.7 (10.37%)	19.1 (8.72%)
	466 to 233	24.0 (10.96%)	23.4 (10.68%)	18.1 (8.26%)
	233 to 233	19.5 (8.90%)	19.3 (8.81%)	16.7 (7.63%)
Stroke	7094 to 402	7.9 (3.61%)	7.6 (3.47%)	7.9 (3.61%)
	1608 to 402	8.6 (3.93%)	7.9 (3.61%)	11.4 (5.21%)
	804 to 402	9.4 (4.29%)	9.0 (4.11%)	13.5 (6.16%)
	402 to 402	12.8 (5.84%)	12.4 (5.66%)	14.7 (6.71%)
Diabetes	6552 to 944	10.5 (4.79%)	10.1 (4.61%)	10.5 (4.79%)
	3776 to 944	13.2 (6.03%)	11.7 (5.34%)	11.4 (5.21%)
	1888 to 944	13.2 (6.03%)	11.4 (5.21%)	10.7 (4.89%)
	944 to 944	11.7 (5.34%)	11.0 (5.02%)	9.6 (4.38%)
HBP	4438 to 3058	26.1 (11.92%)	25.4 (11.60%)	16.5 (7.53%)
	3058 to 3058	24.6 (11.23%)	24.3 (11.10%)	17.7 (8.08%)
Dementia	7360 to 136	11.8 (5.39%)	11.8 (5.39%)	11.5 (5.25%)
	544 to 136	15.0 (6.85%)	14.9 (6.80%)	16.3 (7.44%)
	272 to 136	16.5 (7.53%)	16.5 (7.53%)	15.0 (6.85%)
	136 to 136	13.4 (6.12%)	13.4 (6.12%)	14.1 (6.44%)
Cataract	5344 to 2150	7.5 (3.42%)	7.1 (3.24%)	8.3 (3.79%)
	4300 to 2150	7.1 (3.24%)	6.6 (3.01%)	7.8 (3.56%)
	2150 to 2150	6.5 (2.97%)	6.3 (2.88%)	6.9 (3.15%)
Arthritis	4398 to 3098	16.4 (7.49%)	15.0 (6.85%)	13.1 (5.98%)
	3098 to 3098	19.9 (9.09%)	18.2 (8.31%)	13.3 (6.07%)
Osteoporosis	6796 to 700	16.3 (7.44%)	15.6 (7.12%)	13.1 (5.98%)
	2800 to 700	17.5 (7.99%)	17.5 (7.99%)	13.4 (6.12%)
	1400 to 700	14.0 (6.39%)	13.9 (6.35%)	13.2 (6.03%)
	700 to 700	13.4 (6.12%)	13.4 (6.12%)	12.6 (5.75%)
Parkinsons	7433 to 63	8.0 (3.65%)	7.9 (3.61%)	7.6 (3.47%)
	252 to 63	7.5 (3.42%)	7.5 (3.42%)	7.8 (3.56%)
	126 to 63	8.0 (3.65%)	8.0 (3.65%)	8.2 (3.74%)
	63 to 63	7.3 (3.33%)	7.0 (3.20%)	7.6 (3.47%)

Table 4.7: Relative frequency (%) of selection for different feature types, for the standard CFS

Dataset		Original Features				Constructed Longitudinal Features (CLFs)						Total		
Disease	Class Distribution	General	wave 2	wave 4	wave 6	diff_w24	diff_w46	diff_w26	mono_w246	up_w24	up_w46	Original Total	CLF Total	All Total
HeartAtt	7061 to 435	100.00%	10.25%	15.61%	25.00%	0.00%	6.25%	2.27%	0.00%	0.00%	0.00%	18.18%	2.04%	10.96%
HeartAtt	1740 to 435	95.00%	8.25%	14.15%	25.00%	0.00%	1.67%	1.82%	0.00%	0.00%	0.00%	16.94%	0.82%	9.73%
HeartAtt	870 to 435	95.00%	6.50%	16.34%	22.63%	0.00%	0.00%	2.27%	0.00%	0.00%	0.00%	16.36%	0.51%	9.27%
HeartAtt	435 to 435	100.00%	7.00%	13.90%	23.16%	0.00%	0.42%	3.64%	0.00%	0.00%	0.00%	15.95%	0.92%	9.22%
Angina	7263 to 233	50.00%	9.75%	13.90%	9.21%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	11.65%	0.00%	6.44%
Angina	932 to 233	50.00%	11.75%	17.80%	24.21%	0.00%	1.67%	0.91%	0.00%	0.00%	0.00%	18.35%	0.61%	10.41%
Angina	466 to 233	50.00%	11.25%	20.24%	24.74%	0.00%	2.92%	0.45%	0.00%	0.00%	0.00%	19.17%	0.82%	10.96%
Angina	233 to 233	50.00%	6.25%	14.88%	24.21%	0.00%	1.25%	1.82%	0.00%	0.00%	0.00%	15.54%	0.71%	8.90%
Stroke	7094 to 402	50.00%	2.50%	5.37%	8.16%	0.00%	0.00%	2.73%	0.00%	0.00%	0.00%	6.03%	0.61%	3.61%
Stroke	1608 to 402	50.00%	3.00%	3.90%	8.42%	0.00%	0.83%	5.91%	0.45%	0.00%	0.00%	5.79%	1.63%	3.93%
Stroke	804 to 402	50.00%	4.75%	3.17%	10.26%	0.00%	0.42%	5.45%	0.00%	0.00%	0.00%	6.69%	1.33%	4.29%
Stroke	402 to 402	50.00%	3.00%	7.32%	16.32%	0.00%	1.25%	5.00%	0.00%	0.00%	0.00%	9.42%	1.43%	5.84%
Diabetes	6552 to 944	0.00%	6.25%	5.12%	10.53%	0.00%	6.67%	1.36%	0.00%	0.00%	0.00%	7.11%	1.94%	4.79%
Diabetes	3776 to 944	0.00%	5.75%	7.56%	14.47%	0.00%	6.25%	3.64%	0.00%	0.00%	0.00%	9.01%	2.35%	6.03%
Diabetes	1888 to 944	0.00%	5.25%	8.78%	15.00%	0.00%	4.58%	3.18%	0.00%	0.00%	0.00%	9.42%	1.84%	6.03%
Diabetes	944 to 944	5.00%	4.25%	6.34%	14.21%	0.00%	4.58%	3.64%	0.00%	0.00%	0.00%	8.10%	1.94%	5.34%
HBP	4438 to 3058	50.00%	10.25%	18.54%	27.63%	0.00%	4.58%	8.18%	0.00%	0.00%	0.00%	19.17%	2.96%	11.92%
HBP	3058 to 3058	50.00%	9.75%	18.05%	26.32%	0.00%	4.17%	5.91%	0.00%	0.00%	0.00%	18.43%	2.35%	11.23%
Dementia	7360 to 136	50.00%	4.50%	8.78%	6.84%	0.00%	7.92%	4.09%	0.00%	0.00%	0.00%	7.44%	2.86%	5.39%
Dementia	544 to 136	50.00%	10.00%	9.51%	6.32%	0.00%	7.50%	8.64%	0.00%	0.00%	0.00%	9.34%	3.78%	6.85%
Dementia	272 to 136	50.00%	11.00%	10.98%	8.42%	0.00%	5.00%	10.00%	0.00%	0.00%	0.00%	10.83%	3.47%	7.53%
Dementia	136 to 136	50.00%	9.00%	6.59%	9.21%	0.00%	5.42%	5.91%	0.00%	0.00%	0.00%	8.93%	2.65%	6.12%
Cataract	5344 to 2150	50.00%	1.50%	5.37%	8.95%	0.00%	1.25%	0.00%	0.00%	0.00%	0.00%	5.95%	0.31%	3.42%
Cataract	4300 to 2150	50.00%	1.00%	5.85%	7.63%	0.00%	1.25%	0.45%	0.00%	0.00%	0.00%	5.54%	0.41%	3.24%
Cataract	2150 to 2150	50.00%	0.75%	4.15%	8.16%	0.00%	1.25%	0.45%	0.00%	0.00%	0.00%	5.04%	0.41%	2.97%
Arthritis	4398 to 3098	100.00%	2.75%	13.41%	20.53%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	13.55%	0.00%	7.49%
Arthritis	3098 to 3098	100.00%	5.25%	16.59%	22.89%	0.00%	0.83%	0.45%	0.00%	0.00%	0.00%	16.20%	0.31%	9.09%
Osteoporosis	6796 to 700	100.00%	7.00%	12.20%	14.74%	0.00%	0.42%	3.64%	0.00%	0.00%	0.00%	12.73%	0.92%	7.44%
Osteoporosis	2800 to 700	100.00%	6.25%	14.15%	17.37%	0.00%	0.00%	2.73%	0.00%	0.00%	0.00%	13.97%	0.61%	7.99%
Osteoporosis	1400 to 700	95.00%	5.00%	9.02%	15.26%	0.00%	0.42%	2.27%	0.00%	0.00%	0.00%	11.07%	0.61%	6.39%
Osteoporosis	700 to 700	100.00%	4.75%	8.78%	14.21%	0.00%	0.42%	1.82%	0.00%	0.00%	0.00%	10.66%	0.51%	6.12%
Parkinsons	7433 to 63	85.00%	4.25%	4.39%	3.95%	0.00%	2.50%	2.73%	0.00%	3.33%	0.00%	5.54%	1.33%	3.65%
Parkinsons	252 to 63	95.00%	4.75%	4.15%	3.16%	0.00%	0.83%	1.36%	0.91%	3.33%	0.00%	5.54%	0.82%	3.42%
Parkinsons	126 to 63	100.00%	5.50%	4.63%	3.16%	0.00%	0.83%	2.27%	0.00%	0.00%	0.00%	6.03%	0.71%	3.65%
Parkinsons	63 to 63	95.00%	4.75%	4.39%	2.89%	0.00%	0.83%	0.91%	0.91%	0.00%	0.00%	5.54%	0.61%	3.33%

Table 4.8: Relative frequency (%) of selection for different feature types, for Exh-CFS-Gr+CFS

Dataset		Original Features				Constructed Longitudinal Features (CLFs)						Total		
Disease	Class Distribution	General	wave 2	wave 4	wave 6	diff_w24	diff_w46	diff_w26	mono_w246	up_w24	up_w46	Original Total	CLF Total	All Total
HeartAtt	7061 to 435	100.00%	10.25%	15.61%	25.00%	0.00%	6.25%	2.27%	0.00%	0.00%	0.00%	18.18%	2.04%	10.96%
HeartAtt	1740 to 435	95.00%	8.00%	13.90%	25.00%	0.00%	1.67%	1.82%	0.00%	0.00%	0.00%	16.78%	0.82%	9.63%
HeartAtt	870 to 435	95.00%	6.25%	15.12%	23.16%	0.00%	0.00%	1.36%	0.00%	0.00%	0.00%	16.03%	0.31%	9.00%
HeartAtt	435 to 435	100.00%	5.50%	13.66%	23.16%	0.00%	0.42%	3.64%	0.00%	0.00%	0.00%	15.37%	0.92%	8.90%
Angina	7263 to 233	50.00%	9.50%	13.90%	9.21%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	11.57%	0.00%	6.39%
Angina	932 to 233	50.00%	11.50%	17.80%	24.21%	0.00%	1.67%	0.91%	0.00%	0.00%	0.00%	18.26%	0.61%	10.37%
Angina	466 to 233	50.00%	10.75%	19.51%	24.74%	0.00%	2.92%	0.00%	0.00%	0.00%	0.00%	18.76%	0.71%	10.68%
Angina	233 to 233	50.00%	5.75%	14.88%	24.21%	0.00%	1.25%	1.82%	0.00%	0.00%	0.00%	15.37%	0.71%	8.81%
Stroke	7094 to 402	50.00%	2.00%	5.37%	8.16%	0.00%	0.00%	2.27%	0.00%	0.00%	0.00%	5.87%	0.51%	3.47%
Stroke	1608 to 402	50.00%	3.00%	3.90%	8.42%	0.00%	0.83%	3.18%	0.00%	0.00%	0.00%	5.79%	0.92%	3.61%
Stroke	804 to 402	50.00%	4.25%	3.17%	10.26%	0.00%	0.42%	4.55%	0.00%	0.00%	0.00%	6.53%	1.12%	4.11%
Stroke	402 to 402	50.00%	2.50%	7.32%	16.32%	0.00%	1.25%	4.09%	0.00%	0.00%	0.00%	9.26%	1.22%	5.66%
Diabetes	6552 to 944	0.00%	5.25%	5.12%	11.05%	0.00%	6.67%	0.45%	0.00%	0.00%	0.00%	6.94%	1.73%	4.61%
Diabetes	3776 to 944	0.00%	4.50%	6.83%	13.68%	0.00%	6.25%	1.82%	0.00%	0.00%	0.00%	8.10%	1.94%	5.34%
Diabetes	1888 to 944	0.00%	3.75%	7.32%	13.95%	0.00%	4.17%	2.73%	0.00%	0.00%	0.00%	8.10%	1.63%	5.21%
Diabetes	944 to 944	5.00%	3.75%	6.10%	13.42%	0.00%	3.75%	4.09%	0.00%	0.00%	0.00%	7.60%	1.84%	5.02%
HBP	4438 to 3058	50.00%	10.25%	18.05%	27.11%	0.00%	4.17%	7.27%	0.00%	0.00%	0.00%	18.84%	2.65%	11.60%
HBP	3058 to 3058	50.00%	10.00%	18.05%	26.32%	0.00%	4.17%	4.09%	0.00%	0.00%	0.00%	18.51%	1.94%	11.10%
Dementia	7360 to 136	50.00%	4.50%	8.78%	6.84%	0.00%	7.92%	4.09%	0.00%	0.00%	0.00%	7.44%	2.86%	5.39%
Dementia	544 to 136	50.00%	9.75%	9.51%	6.32%	0.00%	7.50%	8.64%	0.00%	0.00%	0.00%	9.26%	3.78%	6.80%
Dementia	272 to 136	50.00%	11.00%	10.98%	8.42%	0.00%	5.00%	10.00%	0.00%	0.00%	0.00%	10.83%	3.47%	7.53%
Dementia	136 to 136	50.00%	9.00%	6.59%	9.21%	0.00%	5.42%	5.91%	0.00%	0.00%	0.00%	8.93%	2.65%	6.12%
Cataract	5344 to 2150	50.00%	1.50%	4.88%	9.21%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	5.87%	0.00%	3.24%
Cataract	4300 to 2150	50.00%	0.50%	5.85%	7.63%	0.00%	0.00%	0.45%	0.00%	0.00%	0.00%	5.37%	0.10%	3.01%
Cataract	2150 to 2150	50.00%	1.00%	3.90%	8.16%	0.00%	0.42%	0.45%	0.00%	0.00%	0.00%	5.04%	0.20%	2.88%
Arthritis	4398 to 3098	100.00%	0.75%	12.93%	19.47%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	12.40%	0.00%	6.85%
Arthritis	3098 to 3098	100.00%	2.50%	16.34%	22.11%	0.00%	0.00%	0.45%	0.00%	0.00%	0.00%	14.96%	0.10%	8.31%
Osteoporosis	6796 to 700	100.00%	6.75%	11.71%	13.95%	0.00%	0.42%	3.18%	0.00%	0.00%	0.00%	12.23%	0.82%	7.12%
Osteoporosis	2800 to 700	100.00%	6.25%	14.15%	17.37%	0.00%	0.00%	2.73%	0.00%	0.00%	0.00%	13.97%	0.61%	7.99%
Osteoporosis	1400 to 700	95.00%	5.00%	9.02%	15.26%	0.00%	0.42%	1.82%	0.00%	0.00%	0.00%	11.07%	0.51%	6.35%
Osteoporosis	700 to 700	100.00%	4.75%	8.78%	14.21%	0.00%	0.42%	1.82%	0.00%	0.00%	0.00%	10.66%	0.51%	6.12%
Parkinsons	7433 to 63	85.00%	4.25%	4.39%	3.68%	0.00%	2.50%	2.73%	0.00%	3.33%	0.00%	5.45%	1.33%	3.61%
Parkinsons	252 to 63	95.00%	4.75%	4.15%	3.16%	0.00%	0.83%	1.36%	0.91%	3.33%	0.00%	5.54%	0.82%	3.42%
Parkinsons	126 to 63	100.00%	5.50%	4.63%	3.16%	0.00%	0.83%	2.27%	0.00%	0.00%	0.00%	6.03%	0.71%	3.65%
Parkinsons	63 to 63	95.00%	4.00%	4.39%	2.89%	0.00%	0.83%	0.91%	0.91%	0.00%	0.00%	5.29%	0.61%	3.20%

4.3 Results for the Naïve Bayes Algorithm

Similar to the previous section, we report the results obtained by Naïve Bayes (NB) when using different CFS variants. Again, we used the statistical tests with a significance level of α = 0.05, and N = 35 (35 datasets) in our experiments. This section is also divided into three subsections, which are analogous to the three subsections of the previous section – i.e., each subsection evaluates a different set of CFS variants.

4.3.1 Results comparing the two proposed CFS variants based on exhaustive search

Table 4.9 reports the F-measure values obtained by NB, after applying the two CFS variants based on exhaustive search, namely Exh-CFS-Gr and Exh-CFS-Gr+CFS methods, breaking down by each disease used as a class variable in wave 7 and by different degrees of class distribution.

Note that Table 4.9 reports, for each dataset, only the mean F-measure value over the two class labels, since this is the measure used in the statistical significance analysis, but the detailed values of the precision, recall and F-measure for each class label can be found in the Appendix, Section B.

Similarly, to the results reported in the previous section (for J48), the results in Table 4.9 for Naïve Bayes show that more balanced class distributions, in general, led to higher F-measure values – with a few exceptions, namely the HeartAtt and Diabetes datasets. Interestingly, for each dataset, the results are consistent across different class distributions, i.e., the best CFS variant is the same for all class distributions. In addition, Exh-CFS-Gr performed considerably better than Exh-CFS-Gr+CFS, with 25 wins, 9 losses and just 1 tie. Furthermore, the average rank of Exh-CFS-Gr (1.27) is lower (better) than that of Exh-CFS-Gr+CFS (1.73). Moreover, with the use of the Wilcoxon signed-ranks test the null hypothesis is rejected with a p-value of 0.05. Hence, there is statistically significant evidence supporting the conclusion that Exh-CFS-Gr performed better than Exh-CFS-Gr+CFS for the NB classification algorithm.

Regarding the number of features selected by each method, recall that we are using CFS variants as filter feature selection methods, independent from the classification algorithm. Hence, the number of features selected by each CFS variant being considered here, in the context of the results for Naïve Bayes, is exactly the number of features reported in Table 4.2, in the context of the results for J48.

As shown in Table 4.2 and discussed earlier, the number of features selected by Exh-CFS-Gr is substantially greater than that of features selected by Exh-CFS-Gr+CFS for all datasets, due to the reason mentioned in Section 4.2.1. Hence, the larger feature subsets selected by Exh-CFS-Gr led to an overall better predictive accuracy than the feature subsets selected by Exh-CFS-Gr+CFS when using Naïve Bayes as the classification algorithm; even though NB is particularly sensitive to redundant features, and so it could benefit from the further removal of redundant features associated with the application of standard CFS in the second phase of Exh-CFS-Gr+CFS. This is because although the standard CFS eliminated redundant features across the groups, it seems it has removed relevant features too, i.e. its feature selection process was too strong, selecting too few features. Actually, in 30 out of the 35 datasets, Exh-CFS-Gr+CFS selected less than 10% of the original number of features, which degraded NB's predictive performance overall. Hence, the predictive performance of NB was better when Exh-CFS-Gr was applied.

Moreover, a further analysis has been conducted by using the Pearson's linear correlation coefficient (r) to measure the correlation between the predictive accuracy and the number of selected features across the 35 datasets. As for Exh-CFS-Gr, the value of r is 0.780, which indicates a strong positive correlation. Therefore, higher predictive accuracies tend to be observed with larger sizes of selected feature subsets. In contrast, the value of r for Exh-CFS-Gr+CFS is 0.035 which indicates a very weak correlation between the predictive accuracy and the number of features selected by Exh-CFS-Gr+CFS.

Table 4.9: F-measure values obtained by NB after applying the two CFS variants based on exhaustive search.

Disease	Class Distribution	Exh-CFS-Gr+NB	Exh-CFS-Gr+CFS+NB
HeartAtt	7061 to 435	**0.616**	0.606
	1740 to 435	**0.628**	0.626
	870 to 435	**0.626**	0.621
	435 to 435	**0.619**	0.615
Angina	7263 to 233	**0.571**	0.544
	932 to 233	**0.575**	0.566
	466 to 233	**0.579**	0.570
	233 to 233	**0.576**	0.561
Stroke	7094 to 402	**0.596**	0.541
	1608 to 402	**0.603**	0.579
	804 to 402	**0.608**	0.585
	402 to 402	**0.610**	0.600
Diabetes	6552 to 944	0.749	**0.791**
	3776 to 944	0.741	**0.780**
	1888 to 944	0.739	**0.767**
	944 to 944	0.733	**0.749**
HBP	4438 to 3058	0.671	**0.689**
	3058 to 3058	0.676	**0.693**
Dementia	7360 to 136	**0.588**	0.563
	544 to 136	**0.610**	0.587
	272 to 136	**0.615**	0.598
	136 to 136	**0.603**	0.589
Cataract	5344 to 2150	0.664	**0.665**
	4300 to 2150	0.663	**0.667**
	2150 to 2150	0.658	**0.676**
Arthritis	4398 to 3098	**0.631**	**0.631**
	3098 to 3098	**0.629**	0.626
Osteoporosis	6796 to 700	**0.616**	0.574
	2800 to 700	**0.618**	0.614
	1400 to 700	**0.618**	0.617
	700 to 700	**0.618**	0.613
Parkinsons	7433 to 63	**0.501**	0.496
	252 to 63	**0.539**	0.509
	126 to 63	**0.559**	0.557
	63 to 63	**0.570**	0.563

4.3.2 Results comparing the two proposed CFS variants based on different weights for two types of redundancy

Table 4.10 reports the F-measure values for the pair of proposed CFS variants based on using different weights for two types of redundancy, namely temporal redundancy (among the same group of temporally redundant features) and standard (non-temporal) redundancy among features in different groups, where both CFS variants are run before applying the NB algorithm. Recall that these CFS variants are called Coarse-grained Weighted Redundancy (Co-WR-CFS) and Fine-Grained Weighted Redundancy (Fi-WR-CFS).

Similarly to Table 4.9, Table 4.10 shows that overall the F-measure values increase for both CFS variants when the class distributions become increasingly more balanced (by undersampling instances of the majority class), with the exception of the results for Diabetes.

Regarding the overall predictive accuracy of the two CFS variants, the Table shows that Fi-WR-CFS achieved higher accuracy for 27 out of 35 datasets, with only 3 losses and 5 ties. After running the Wilcoxon signed-ranks test, it is clear that Fi-WR-CFS significantly outperforms Co-WR-CFS, with a p-value < 0.001.

As shown in Table 4.4 and discussed earlier, in general, the number of features selected by Fi-WR-CFS is greater than that of features selected by Co-WR-CFS for most of the datasets. Since Fi-WR-CFS significantly outperforms Co-WR-CFS in terms of predictive accuracy, the larger feature subset selected by Fi-WR-CFS led to the better predictive accuracy, suggesting that Co-WR-CFS performed a feature selection process that was too strong, selecting too few features. This could be explained by a limitation of Co-WR-CFS, which over-emphasizes the degree of temporal redundancy associated with (usually) a minority of feature pairs, as discussed earlier. As a consequence, it is plausible that some relevant features were also removed by that method. Hence, the performance of NB was better when Fi-WR-CFS was applied.

Moreover, by measuring the Pearson’s linear correlation coefficient, the values of *r* for Co-WR-CFS and Fi-WR-CFS are small (-0.082 and 0.209 respectively). Hence, the correlation between the predictive accuracy and the number of selected features is weak for both these proposed CFS variants.

Table 4.10: F-measure values obtained by NB after applying the two CFS variants based on weighted redundancy.

Disease	Class Distribution	Co-WR-CFS+NB	Fi-WR-CFS+NB
HeartAtt	7061 to 435	0.586	**0.616**
	1740 to 435	0.614	**0.627**
	870 to 435	0.621	**0.622**
	435 to 435	0.607	**0.617**
Angina	7263 to 233	0.537	**0.539**
	932 to 233	0.557	**0.566**
	466 to 233	0.560	**0.578**
	233 to 233	0.564	**0.567**
Stroke	7094 to 402	**0.552**	0.543
	1608 to 402	0.588	**0.596**
	804 to 402	0.601	**0.607**
	402 to 402	0.597	**0.606**
Diabetes	6552 to 944	**0.788**	0.783
	3776 to 944	0.778	**0.785**
	1888 to 944	0.763	**0.766**
	944 to 944	0.748	**0.753**
HBP	4438 to 3058	0.670	**0.689**
	3058 to 3058	0.674	**0.693**
Dementia	7360 to 136	0.565	**0.577**
	544 to 136	0.591	**0.601**
	272 to 136	0.598	**0.601**
	136 to 136	0.591	**0.601**
Cataract	5344 to 2150	0.668	**0.671**
	4300 to 2150	0.667	**0.676**
	2150 to 2150	**0.676**	**0.676**
Arthritis	4398 to 3098	0.624	**0.628**
	3098 to 3098	0.617	**0.624**
Osteoporosis	6796 to 700	0.597	**0.615**
	2800 to 700	0.612	**0.616**
	1400 to 700	**0.613**	0.612
	700 to 700	0.608	**0.611**
Parkinsons	7433 to 63	**0.496**	**0.496**
	252 to 63	**0.513**	**0.513**
	126 to 63	**0.556**	**0.556**
	63 to 63	**0.556**	**0.556**

4.3.3 Results comparing four feature selection approaches using NB

This section compares the results obtained by four feature selection approaches: two proposed CFS variants – one based on exhaustive search (Exh-CFS-Gr) and the other based on weighted redundancy (WR-CFS) – against the standard CFS method and the baseline approach of giving all input features to NB – i.e., no feature selection. We began by calculating the average ranks for each variant, as described earlier – recall that the smaller the average rank, the better (higher) the F-measure value of a CFS variant across all datasets in general. The best average rank was obtained by Exh-CFS-Gr (average rank 1.64), followed by Fi-WR-CFS (rank 2.37), standard CFS (rank 2.54) and no feature selection, i.e. NB alone (rank 3.44).

The Friedman test was used to determine whether or not there is a significant difference between the average ranks of the four feature selection approaches and their mean rank (2.5). The calculated value of F_F is 17.29. With four feature selection approaches and 35 datasets, F_F is distributed according to the F distribution with 4 – 1 = 3 and (4-1) x (35-1) = 102 degrees of freedom. The critical value of $F(3,102)$ for $\alpha = 0.05$ is 2.69. Since, F_F is greater than the critical value, the null hypothesis is rejected. Hence, there is a statistically significant evidence against the claim that all the four feature selection approaches are equivalent.

Therefore, we proceeded with the Nemenyi test, a post-hoc test, for pairwise comparisons. Using the critical value of 2.57 for the two-tailed Nemenyi test, the critical difference is 0.79. As the differences in the average ranks are greater than the critical value, four null-hypotheses can be rejected, as follows.

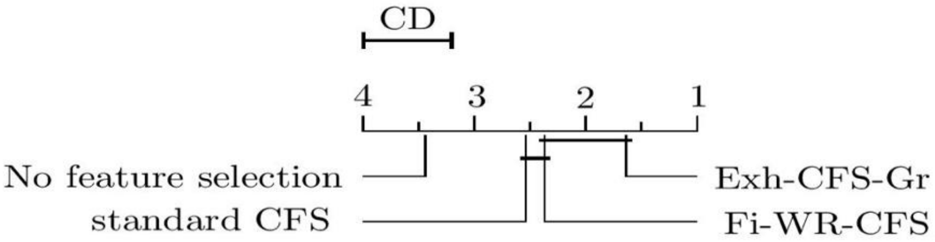

Figure 4.1: The critical diagram for the results of the Nemenyi test

First of all, applying standard CFS improves the predictive accuracy for NB (the average rank difference is: 3.44 – 2.54 = 0.90 > 0.79). Next, the accuracy of pure NB (without feature selection) is significantly worse than that of NB with both proposed methods: Exh-CFS-Gr (3.44 – 1.64 = 1.80 > 0.79) and Fi-WR-CFS (3.44 – 2.37 = 1.07 > 0.79). Moreover, NB with Exh-CFS-Gr performs significantly better than NB with standard CFS (2.54 – 1.64 = 0.90 > 0.79). Lastly, although Exh-CFS-Gr outperforms Fi-WR-CFS, their average rank difference in their predictive performance is not significant (2.37 – 1.64 = 0.73 < 0.79).

The critical diagram with the results of the Nemenyi test is shown in Figure 4.1. As proposed in (Demšar 2006), this diagram reports the average ranks of the methods on the horizontal axis, with the best method (rank 1) at the rightmost position. Below the horizontal axis, there is a horizontal line connecting the lines representing two methods if the difference of average ranks between those two methods was smaller than the critical difference (CD) – whose size is shown at the top of the graph. This indicates that there is no significant difference among those two methods. This diagram shows that Exh-CFS-Gr’s average rank was significantly better than the average rank of two of the three other approaches, the exception was Fi-WR-CFS.

With regard to the size of the selected feature subsets, Table 4.12 shows the average number and percentage of features selected by each of the three CFS variants (across the 10 cross-validation folds), for each dataset – i.e., each combination of disease and class distribution. The number of features selected by the standard CFS and that selected by Fi-WR-CFS are relatively small, compared with that selected by Exh-CFS-Gr.

As mentioned earlier, the results for Exh-CFS-Gr are associated with a strong positive linear correlation coefficient between the predictive accuracy and the number of selected features, with the *r* value of 0.780. In contrast, Fi-WR-CFS and standard CFS produced feature subsets with weak correlations between those two variables (0.209 and 0.081 respectively).

These results are broadly similar (with one exception – see below) to the results obtained with J48, where there is little correlation between the predictive accuracy and the number of features selected by each CFS variant, in general. The exception is that, when using NB, the correlation between predictive accuracy and number of selected features was strong for Exh-CFS-Gr. This is consistent with the fact that, unlike J48, in general NB's predictive performance was substantially negatively affected by the relatively small numbers of features selected by the other CFS variants, in the datasets used in the experiments.

Last but not least, Table 4.13 reports the relative frequency (%) of selection for different feature types for Exh-CFS-Gr. The results in this table can be contrasted with the analogous results for the standard CFS reported in Table 4.7. We focus on comparing these two methods here because Exh-CFS-Gr obtained overall the best results in this Section, whilst the standard CFS is a natural baseline. Recall that in both these tables the column "General" refers to the age and gender features, the only two features not derived from the Nurse data. In each cell, the relative frequency was calculated as the frequency of selection summed for all features of the corresponding feature type over the maximum possible number of selections, which was the number of features in the feature type times 10 (considering the 10 folds of the cross- validation procedure).

Comparing the relative frequencies of features selected by the two CFS variants in the "All Total" column, Exh-CFS-Gr (Table 4.13) selected considerably larger feature subsets for all datasets than the standard CFS (Table 4.7). Intuitively, this can be explained by the fact that, although both Exh-CFS-Gr and standard CFS evaluate the relevance of each candidate feature with respect to the class variable, they use different approaches to evaluate the redundancy among candidate features, as follows. Exh-CFS-Gr evaluates redundancy only among a small group of temporally related features, which are variations (across time) of feature values referring to the same conceptual feature. It does not detect redundancy between features in different groups, so there is less opportunity to remove features based on the redundancy criterion. By contrast, the standard CFS, the baseline, can detect redundancy between potentially any pair of features (since it does not divide features into groups), increasing the opportunity to remove features based on the redundancy criterion.

Note that Exh-CFS-Gr always selects gender and age (100% frequency in the column "General" features), for all datasets. This is because we considered age and gender as separate conceptual features, so that they belong to different conceptual groups, and Exh-CFS-Gr always selects at least one feature from each group. In addition, as shown in Table 4.13, the relative selection frequency of original features increased monotonically with time (i.e., from wave 2 to wave 4 to wave 6) in 24 of the 35 datasets. The main exceptions were the Dementia, Osteoporosis and Parkinson's datasets.

Regarding the constructed longitudinal features (CLFs), as shown in the column "CLF Total" of Table 4.7, standard CFS rarely selected CLFs – their overall relative frequency of selection was below 4% in all 35 datasets. In contrast, for Exh-CFS-Gr (Table 4.13), the relative selection frequencies of CLFs were considerably higher, varying from 8.8% to 26.9% across the datasets. In addition, among the three diff feature types, the feature type diff_w24 (the only diff feature type not involving wave 6 – the most recent wave) had a selection frequency of 0 in all datasets, for both Exh-CFS-Gr and the standard CFS. However, in general diff_w46 and diff_w26 features were selected substantially more often by Exh-CFS-Gr than by standard CFS.

Regarding the feature type mono_w246, which was designed to capture monotonicity patterns in the values of observed features from wave 2 to wave 4 to wave 6, note that this feature type was never selected by standard CFS in 33 datasets, with a selection frequency below 1% in the other two datasets (for Parkinson's disease) (Table 4.7). By contrast, in the results for Exh-CSF-Gr (Table 4.13), the feature type mono_w246 had a dramatically higher selection frequency, above 30% in 23 datasets, and above 50% in 9 datasets. Interestingly, in the four Parkinson's datasets, mono_w246 features had a selection frequency above 90%. These dramatically increased relatively selection frequencies are partly a result of the fact that in general Exh-CSF-Gr selects many more features than standard CFS, as explained earlier.

A similar result was observed for the feature types up_w24 and up_w46, which were only selected by standard CFS in two datasets (both for Parkinson's) with a relative frequency of 3.3%, whilst these two feature types were, in general, selected much more often across the datasets by Exh-CFS-Gr. However, the calculated selection frequencies for these two feature types are less robust values, since each of them includes just two or three CLFs, unlike the much larger numbers of CLFs included in the diff feature type.

Table 4.11: F-measure values obtained by NB after applying different CFS methods

Disease	Class Distribution	NB	standard_CFS +NB	Exh-CFS-Gr +NB	Fi-WR-CFS +NB
HeartAtt	7061 to 435	0.610	0.606	**0.616**	**0.616**
	1740 to 435	0.612	0.627	**0.628**	0.627
	870 to 435	0.608	0.622	**0.626**	0.622
	435 to 435	0.606	**0.620**	0.619	0.617
Angina	7263 to 233	0.563	0.548	**0.571**	0.539
	932 to 233	0.559	0.566	**0.575**	0.566
	466 to 233	0.565	0.572	**0.579**	0.578
	233 to 233	0.559	0.562	**0.576**	0.567
Stroke	7094 to 402	0.596	0.546	**0.596**	0.543
	1608 to 402	0.592	0.574	**0.603**	0.596
	804 to 402	0.594	0.587	**0.608**	0.607
	402 to 402	0.590	0.602	**0.610**	0.606
Diabetes	6552 to 944	0.743	**0.790**	0.749	0.783
	3776 to 944	0.740	**0.786**	0.741	0.785
	1888 to 944	0.738	**0.774**	0.739	0.766
	944 to 944	0.732	**0.760**	0.733	0.753
HBP	4438 to 3058	0.671	**0.690**	0.671	0.689
	3058 to 3058	0.672	**0.693**	0.676	**0.693**
Dementia	7360 to 136	0.604	0.563	**0.588**	0.577
	544 to 136	0.598	0.587	**0.610**	0.601
	272 to 136	0.596	0.600	**0.615**	0.601
	136 to 136	0.589	0.589	**0.603**	0.601
Cataract	5344 to 2150	0.649	0.663	0.664	**0.671**
	4300 to 2150	0.646	0.667	0.663	**0.676**
	2150 to 2150	0.641	**0.677**	0.658	0.676
Arthritis	4398 to 3098	0.614	0.630	**0.631**	0.628
	3098 to 3098	0.614	0.625	**0.629**	0.624
Osteoporosis	6796 to 700	0.602	0.575	**0.616**	0.615
	2800 to 700	0.606	0.614	**0.618**	0.616
	1400 to 700	0.606	0.617	**0.618**	0.612
	700 to 700	0.610	0.613	**0.618**	0.611
Parkinsons	7433 to 63	**0.530**	0.496	0.501	0.496
	252 to 63	0.565	0.509	**0.539**	0.513
	126 to 63	0.555	0.557	**0.559**	0.556
	63 to 63	0.553	0.560	**0.570**	0.556

Table 4.12: Average numbers and percentages of selected features for the three CFS variants.

Disease	Class Distribution	standard_CFS	Exh-CFS-Gr	Fi-WR-CFS
HeartAtt	7061 to 435	24.0 (10.96%)	80.0 (36.53%)	29.3 (13.38%)
	1740 to 435	21.3 (9.73%)	76.7 (35.02%)	27.3 (12.47%)
	870 to 435	20.3 (9.27%)	70.0 (31.96%)	26.4 (12.05%)
	435 to 435	20.2 (9.22%)	65.7 (30.00%)	25.8 (11.78%)
Angina	7263 to 233	14.1 (6.44%)	75.7 (34.57%)	19.5 (8.90%)
	932 to 233	22.8 (10.41%)	74.7 (34.11%)	31.0 (14.16%)
	466 to 233	24.0 (10.96%)	70.3 (32.10%)	29.8 (13.61%)
	233 to 233	19.5 (8.90%)	68.3 (31.19%)	25.8 (11.78%)
Stroke	7094 to 402	7.9 (3.61%)	77.6 (35.43%)	8.3 (3.79%)
	1608 to 402	8.6 (3.93%)	75.4 (34.43%)	21.1 (9.63%)
	804 to 402	9.4 (4.29%)	70.5 (32.19%)	23.5 (10.73%)
	402 to 402	12.8 (5.84%)	64.6 (29.50%)	20.9 (9.54%)
Diabetes	6552 to 944	10.5 (4.79%)	98.3 (44.89%)	15.4 (7.03%)
	3776 to 944	13.2 (6.03%)	96.5 (44.06%)	19.7 (9.00%)
	1888 to 944	13.2 (6.03%)	93.3 (42.60%)	23.9 (10.91%)
	944 to 944	11.7 (5.34%)	86.3 (39.41%)	25.0 (11.42%)
HBP	4438 to 3058	26.1 (11.92%)	98.2 (44.84%)	34.1 (15.57%)
	3058 to 3058	24.6 (11.23%)	95.3 (43.52%)	31.2 (14.25%)
Dementia	7360 to 136	11.8 (5.39%)	68.2 (31.14%)	17.1 (7.81%)
	544 to 136	15.0 (6.85%)	70.7 (32.28%)	23.2 (10.59%)
	272 to 136	16.5 (7.53%)	71.6 (32.69%)	24.7 (11.28%)
	136 to 136	13.4 (6.12%)	66.6 (30.41%)	21.4 (9.77%)
Cataract	5344 to 2150	7.5 (3.42%)	73.2 (33.42%)	8.4 (3.84%)
	4300 to 2150	7.1 (3.24%)	71.9 (32.83%)	8.2 (3.74%)
	2150 to 2150	6.5 (2.97%)	69.7 (31.83%)	8.1 (3.70%)
Arthritis	4398 to 3098	16.4 (7.49%)	80.2 (36.62%)	25.1 (11.46%)
	3098 to 3098	19.9 (9.09%)	75.5 (34.47%)	24.2 (11.05%)
Osteoporosis	6796 to 700	16.3 (7.44%)	89.6 (40.91%)	26.9 (12.28%)
	2800 to 700	17.5 (7.99%)	85.7 (39.13%)	24.7 (11.28%)
	1400 to 700	14.0 (6.39%)	82.6 (37.72%)	22.4 (10.23%)
	700 to 700	13.4 (6.12%)	78.7 (35.94%)	20.1 (9.18%)
Parkinsons	7433 to 63	8.0 (3.65%)	55.9 (25.53%)	8.6 (3.93%)
	252 to 63	7.5 (3.42%)	57.6 (26.30%)	7.8 (3.56%)
	126 to 63	8.0 (3.65%)	56.1 (25.62%)	8.2 (3.74%)
	63 to 63	7.3 (3.33%)	54.2 (24.75%)	7.7 (3.52%)

Table 4.13: Relative frequency (%) of selection for different feature types, for Exh-CFS-Gr

Dataset		Original Features				Constructed Longitudinal Features (CLFs)						Total		
Disease	Class Distribution	General	wave 2	wave 4	wave 6	diff_w24	diff_w46	diff_w26	mono_w246	up_w24	up_w46	Original Total	CLF Total	All Total
HeartAtt	7061 to 435	100.00%	47.00%	50.24%	67.89%	0.00%	12.50%	7.73%	22.73%	60.00%	65.00%	55.54%	13.06%	36.53%
HeartAtt	1740 to 435	100.00%	40.50%	46.83%	63.16%	0.00%	10.00%	9.09%	32.27%	66.67%	90.00%	50.74%	15.61%	35.02%
HeartAtt	870 to 435	100.00%	31.25%	42.68%	58.16%	0.00%	6.25%	7.27%	40.91%	63.33%	95.00%	44.71%	16.22%	31.96%
HeartAtt	435 to 435	100.00%	24.00%	39.27%	55.26%	0.00%	2.92%	6.82%	51.82%	60.00%	80.00%	40.25%	17.35%	30.00%
Angina	7263 to 233	100.00%	45.00%	55.85%	52.89%	0.00%	2.50%	2.73%	38.18%	56.67%	70.00%	52.07%	12.96%	34.57%
Angina	932 to 233	100.00%	39.75%	49.51%	59.74%	0.00%	2.08%	1.36%	41.36%	70.00%	90.00%	50.33%	14.08%	34.11%
Angina	466 to 233	100.00%	32.25%	48.29%	54.47%	0.00%	4.17%	0.45%	45.00%	66.67%	95.00%	45.79%	15.20%	32.10%
Angina	233 to 233	100.00%	30.50%	41.71%	53.42%	0.00%	2.08%	2.73%	50.91%	80.00%	100.00%	42.64%	17.04%	31.19%
Stroke	7094 to 402	100.00%	31.25%	50.98%	67.11%	0.00%	12.08%	22.27%	31.36%	33.33%	50.00%	50.33%	17.04%	35.43%
Stroke	1608 to 402	100.00%	29.50%	48.78%	64.21%	0.00%	12.08%	18.18%	35.91%	43.33%	55.00%	48.10%	17.55%	34.43%
Stroke	804 to 402	100.00%	27.25%	42.20%	61.84%	0.00%	6.25%	16.36%	39.09%	56.67%	70.00%	44.38%	17.14%	32.19%
Stroke	402 to 402	100.00%	20.00%	36.10%	57.11%	0.00%	5.00%	9.09%	50.45%	73.33%	80.00%	38.43%	18.47%	29.50%
Diabetes	6552 to 944	100.00%	56.75%	72.93%	84.47%	0.00%	17.50%	26.82%	4.55%	16.67%	0.00%	71.65%	11.84%	44.89%
Diabetes	3776 to 944	100.00%	55.25%	72.20%	83.95%	0.00%	15.83%	25.00%	4.55%	20.00%	0.00%	70.74%	11.12%	44.06%
Diabetes	1888 to 944	100.00%	50.00%	70.73%	83.42%	0.00%	14.58%	25.91%	4.55%	13.33%	0.00%	68.35%	10.82%	42.60%
Diabetes	944 to 944	100.00%	41.50%	65.12%	79.47%	0.00%	13.33%	19.09%	12.73%	20.00%	0.00%	62.40%	11.02%	39.41%
HBP	4438 to 3058	100.00%	44.50%	76.10%	90.79%	0.00%	20.00%	32.27%	0.00%	26.67%	0.00%	70.66%	12.96%	44.84%
HBP	3058 to 3058	100.00%	39.50%	76.83%	89.21%	0.00%	22.50%	28.18%	0.00%	13.33%	5.00%	68.76%	12.35%	43.52%
Dementia	7360 to 136	100.00%	44.00%	43.41%	30.26%	0.00%	13.75%	18.18%	40.45%	53.33%	75.00%	40.41%	19.69%	31.14%
Dementia	544 to 136	100.00%	46.25%	40.00%	32.37%	0.00%	10.42%	20.45%	50.00%	53.33%	95.00%	40.66%	21.94%	32.28%
Dementia	272 to 136	100.00%	46.75%	41.71%	37.37%	0.00%	6.25%	15.00%	51.82%	46.67%	100.00%	42.98%	20.00%	32.69%
Dementia	136 to 136	100.00%	40.50%	35.12%	36.84%	0.00%	7.50%	10.45%	55.91%	53.33%	100.00%	38.51%	20.41%	30.41%
Cataract	5344 to 2150	100.00%	36.75%	48.29%	73.42%	0.00%	0.00%	11.82%	11.82%	66.67%	80.00%	53.22%	8.98%	33.42%
Cataract	4300 to 2150	100.00%	34.50%	46.59%	72.37%	0.00%	0.00%	9.55%	17.73%	63.33%	80.00%	51.57%	9.69%	32.83%
Cataract	2150 to 2150	100.00%	28.00%	47.56%	69.21%	0.00%	1.67%	4.55%	24.09%	66.67%	100.00%	48.76%	10.92%	31.83%
Arthritis	4398 to 3098	100.00%	28.50%	68.05%	79.74%	0.00%	3.75%	9.55%	18.18%	20.00%	50.00%	59.17%	8.78%	36.62%
Arthritis	3098 to 3098	100.00%	25.25%	64.15%	74.74%	0.00%	3.75%	4.55%	20.91%	33.33%	60.00%	55.21%	8.88%	34.47%
Osteoporosis	6796 to 700	100.00%	64.25%	66.34%	62.37%	0.00%	4.58%	8.18%	33.18%	0.00%	40.00%	64.96%	11.22%	40.91%
Osteoporosis	2800 to 700	100.00%	57.75%	63.66%	61.84%	0.00%	4.58%	7.73%	32.73%	0.00%	50.00%	61.74%	11.22%	39.13%
Osteoporosis	1400 to 700	100.00%	53.50%	61.46%	61.05%	0.00%	3.75%	3.18%	36.36%	0.00%	60.00%	59.34%	11.02%	37.72%
Osteoporosis	700 to 700	100.00%	49.75%	56.59%	58.95%	0.00%	1.67%	2.73%	38.64%	0.00%	85.00%	55.79%	11.43%	35.94%
Parkinsons	7433 to 63	100.00%	23.00%	23.90%	23.95%	0.00%	2.92%	2.73%	90.91%	93.33%	85.00%	24.88%	26.33%	25.53%
Parkinsons	252 to 63	100.00%	24.00%	22.68%	27.11%	0.00%	0.83%	1.36%	95.45%	96.67%	100.00%	25.79%	26.94%	26.30%
Parkinsons	126 to 63	100.00%	23.00%	21.46%	25.79%	0.00%	0.83%	2.27%	94.09%	96.67%	100.00%	24.63%	26.84%	25.62%
Parkinsons	63 to 63	100.00%	22.00%	20.98%	23.68%	0.00%	0.83%	1.36%	92.73%	96.67%	100.00%	23.47%	26.33%	24.75%

5 CONCLUSION

5.1 Summary of Contributions

This thesis has proposed four variants of the CFS (Correlation-based Feature Selection) method adapted to cope with longitudinal classification data, where the values of a variable are repeatedly measured across different time points (called waves). The proposed CFS variants were run in a data pre-processing phase, before running the classification algorithm. By doing so, our goal was to keep the adaptations related to longitudinal data restricted to the pre-processing phase, which has the advantage of enabling any conventional classification algorithm (which ignores the temporal nature of longitudinal data) to be applied to the selected features.

This thesis presents two main contributions. The first one is the creation of the longitudinal datasets used in the experiments, including the construction of features capturing longitudinal information for predicting age-related diseases. The datasets were created from data in the English Longitudinal Study of Ageing (ELSA) database (Marmot *et al.* 2016). The created datasets contain two types of features, namely originally observed features (directly taken from the ELSA database) and the aforementioned constructed longitudinal features, where both feature types occur in three waves. In addition, each individual (observed or constructed) feature was assigned to a group of temporally related features – each group contains observed and constructed features representing variations in the value of a given base feature across the three waves. Besides, each dataset contains a single class variable representing an age-related disease, so that multiple datasets were created for different diseases.

More precisely, we first created 10 datasets, each with a different age-related disease as the class variable. Next, we created variations of these datasets with different class distributions, by performing an undersampling of instances of the majority class, in order to cope with the problem of class imbalance. This produced in total 35 datasets. Note that, although each dataset has a different combination of an age-related disease as the class variable and a class distribution (varying the degree of undersampling), all datasets contain instances representing the same individuals from the ELSA database and the same set of predictive features, representing mainly biomedical information about those individuals.

The second contribution consists of proposing four new variants of the CFS method for selecting features to be used as input by a conventional classification algorithm. These CFS variants take into account (in different ways) the temporal redundancy associated with variations in the value of a given feature across different waves (time points). The four proposed CFS variants were categorised into two types of modifications of the standard CFS method, namely two of the variants modify the standard CFS' search method; whilst the other two variants modify the standard CFS' evaluation function.

The former type of CFS variant includes Exh-CFS-Gr (Exhaustive search-based CFS per group) and Exh-CFS-Gr+CFS (Exhaustive search-based CFS per group followed by standard CFS). The basic idea of both these variants is to use exhaustive search (rather than heuristic search as usual) to select features separately from each group of temporally related features. The use of exhaustive search was made possible by dividing the features into these groups, since within each group the number of features is relatively small. Hence, these two CFS variants exhaustively considered all combinations of features within each group, minimizing the occurrence of temporal redundancy in the feature subset selected from each group. Once this feature selection per group has been done, Exh-CFS-Gr simply merges all features selected across the groups to compute the set of selected features to be used by a classification algorithm. Exh-CFS-Gr+CFS consists of first applying Exh-CFS-Gr and then further applying the standard CFS to the merged set of features selected across all groups. Hence, Exh-CFS-Gr+CFS detects not only temporal feature redundancy within groups (in its first phase, running Exh-CFS-Gr) but also standard (non-temporal) feature redundancy across groups (in its second phase, running standard CFS).

The second type of CFS variant modified the way the merit (evaluation) function is computed, whilst using the same search method used in standard CFS. The two proposed variants of this modification were called Co-WR-CFS (Coarse-grained Weighted-Redundancy CFS) and Fi-WR-CFS (Fine-grained Weighted-Redundancy CFS). Essentially, their merit functions were modified such that the degree of redundancy among the features were weighted unequally, depending on whether the redundancy was between features within the same group of temporally related features (i.e. temporal redundancy) or between features belonging to different groups (conventional, non-temporal redundancy).

These two CFS variants differ in how the weights were assigned. In the case of Co-WR-CFS, a coarse-grained weighting approach was used, where first the average degree of feature redundancy is calculated separately for features within the same group and features across different groups (where the average is over all pairs of features in each case) and then two different weights are assigned to these two average degrees of feature redundancy. By contrast, in the case of Fi-WR-CFS, a fine-grained approach was used, where the redundancy between each pair of features is directly assigned a weight, depending on whether the pair involves features within the same group or in different groups. Note that as explained earlier the coarse-grained approach used by Co-WR-CFS can over-emphasize the degree of redundancy associated with the features in the same group, which tend to be a minority of the feature pairs in a candidate feature subset being evaluated. Fi-WR-CFS reduces this risk, but it has the opposite risk of not emphasizing enough the importance of temporal redundancy involving features within the same group.

Experiments were performed comparing the proposed CFS variants with two baseline approaches, the standard CFS method and the natural baseline of not performing any feature selection in a pre-processing phase, i.e., giving all features to the classification algorithm. In addition, the experiments were performed with two well-known classification algorithms separately, namely the decision-tree induction algorithm J48 (Quinlan 1993; Hall *et al.* 2009) and Naïve Bayes (Sulzmann, Fürnkranz and Hüllermeier 2007). The results reported in this thesis are summarised next from three perspectives, namely the effect of feature selection on the predictive accuracy of the two classification algorithms, the effect of different degrees of undersampling of majority-class instances on the predictive accuracy of the two classification algorithms, and the number and type of features selected by different CFS variants.

First, regarding predictive accuracy, in general there was no statistically significant difference between the predictive accuracies obtained by J48 when different variants of CFS (including standard CFS) were applied in a data pre-processing step, nor any significant difference by comparison with the baseline of no feature selection. This can be explained by the fact that J48 preforms embedded feature selection; so that its predictive accuracy is in principle less sensitive to irrelevant and redundant features, by comparison with Naïve Bayes. Actually, in the case of Naïve Bayes, there was a statistically significant difference in the predictive performances of some CFS variants. In particular, the best CFS variant for Naïve Bayes, namely Exh-CFS-Gr, obtained results significantly better than the results obtained by standard CFS and no feature selection, although there was no significant difference between the results of Exh-CFS-Gr and the second best CFS variant (Fi-WR-CFS).

Second, regarding the effect of different degrees of undersampling of majority-class instances on the predictive accuracy, recall that experiments were performed with increasingly larger amounts of undersampling applied to the training set, up to the point where the number of majority-class instances is reduced to be the same as the number of minority-class instances – i.e., a maximally balanced class distribution. Overall, the results show that the larger the degree of undersampling (i.e. the closer to perfectly balanced the class distribution is), the higher the predictive accuracies of J48 and Naïve Bayes, although sometimes a higher predictive accuracy was achieved with a less balanced class distribution.

Third, regarding the feature subsets selected by different variants of CFS, the number of features selected by Exh-CFS-Gr was substantially greater than that of all other CFS variants by for all datasets. This helps to explain why this feature selection method obtained the best results in the experiments with Naïve Bayes; i.e., it seems that the other CFS variants selected relatively too few features for Naïve Bayes. Regarding the types of features most often selected by the CFS variants, overall, the constructed longitudinal features (CLFs) were selected substantially less often than the features originally observed in the ELSA database, with a few exceptions. Among the different CFS variants, Exh-CFS-Gr selected some CLFs (in particular a CLF designed to capture a monotonic increase or decrease in the value of a feature across three ELSA waves) much more often than the other CFS variants. However, in general Exh-CFS-Gr still selected more originally observed features than CLFs (including that monotonic feature type). Among the originally observed features, broadly speaking (despite several exceptions) the frequency of selection increased from wave 2 to wave 4 to wave 6. This is consistent with the fact that intuitively features from wave 6 are more relevant for predicting an age-related disease in wave 7, given that short-term predictions tend to be more reliable than long-term predictions.

5.2 Future Research Directions

In this work, we have focused on performing experiments with the created longitudinal datasets, which contained both raw (observed) features and CLFs (Constructed Longitudinal Features) synthesised from those raw features. In other words, our aim was to compare the proposed CFS variants against the standard CFS on the created longitudinal datasets. Although our proposed CFS variants achieved higher predictive accuracies than the standard CFS in most cases, there were no experiments conducted to compare the predictive accuracies obtained using the full dataset with both above feature types against the accuracies obtained using only the raw (observed) features. In future work, it would be interesting to perform this kind of experiment, in order to evaluate the effect of the constructed longitudinal features on predictive performance.

Furthermore, the only approach for coping with an imbalanced class distribution used in this work was the undersampling approach, which may have led to loss of relevant information (throwing away many instances of the majority class), especially in a very imbalanced dataset such as Parkinson’s. This was not a significant problem in our experiments, since in general, among dataset variations with different degrees of class imbalance, the best results were obtained with a maximally balanced class distribution (i.e., both classes have the same number of instances), which corresponds to the highest degree of undersampling in our experiments. Nonetheless, it is possible that better results regarding predictive accuracy could be obtained by using another approach to cope with imbalanced class distributions, such as the SMOTE method (Chawla *et al.* 2002), and this could be investigated in future research.

In addition, the CLFs were synthesised from continuous (real-valued) features only, since the proposed approach to create CLFs does not work with categorical features. That is, if a dataset contains only features of categorical type, then there are no CLFs to be created. Therefore, it would be interesting to invent a new sort of CLFs which are synthesised from categorical longitudinal features and are able to capture their temporal information.

Finally, another area for future work involves modifying an existing classification algorithm (or developing a new algorithm) that exploits the temporal information in longitudinal features in a way that works well together with the proposed CFS variants. Actually, in this work, although the proposed CFS variants take into account the temporal information in the longitudinal features in the created datasets, the temporal information associated with the set of selected features is ignored by the conventional classification algorithms used in our experiments. In future work, developing classification algorithms that exploit the temporal information associated with the selected features could improve predictive performance.

REFERENCES

Adam, R.D. (2001). Biology of Giardia lamblia Biology of Giardia lamblia. *Clinical microbiology reviews* **14**:447–469.

Adhikari, S. *et al.* (2015). High-Dimensional Longitudinal Classification with the Multinomial Fused Lasso. *arXiv preprint arXiv:* **1501.07518**:1–21.

Aha, D.W., Kibler, D. and Albert, M.K. (1991). Instance-Based Learning Algorithms. *Machine Learning* **6**:37–66.

Altman, R.B. (2001). Challenges for intelligent systems in biology. *IEEE Intelligent Systems* **16**.

Bache, K. and Lichman, M. (2013). UCI Machine Learning Repository. *University of California Irvine School of Information* [Online]. Available at: https://ergodicity.net/2013/07/.

Baldi, P. and Brunak, S. (2001). *Bioinformatics: The Machine Learning Approach*. MIT Press.

Batista, G.E.A.P.A., Prati, R.C. and Monard, M.C. (2004). A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data. *ACM SIGKDD Explorations Newsletter - Special issue on learning from imbalanced datasets* **6**:20–29.

Bhaskar, H., Hoyle, D.C. and Singh, S. (2006). Machine learning in bioinformatics: A brief survey and recommendations for practitioners. *Computers in Biology* *and Medicine* **36**:1104–1125.

Brazdil, P. and Henery, R. (1994). *Analysis of Results*. Ellis Horwood.

Breiman, L. (2001). Random forests. *Machine Learning* **45**:5–32.

Bruning, J.L. and Kintz, B.L. (1987). *Computational Handbook of Statistic*. Scott, Foresman & Co.

Chawla, N. V. (2009). Data Mining for Imbalanced Datasets: An Overview. In: *Data Mining and Knowledge Discovery Handbook*. Springer US, pp. 875–886.

Chawla, N. V. *et al.* (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research* **16**:321–357.

Chawla, N. V, Japkowicz, N. and Drive, P. (2004). Editorial : Special Issue on Learning from Imbalanced Data Sets. *ACM SIGKDD Explorations Newsletter* **6**:1–6.

Chen, T., Zeng, D. and Wang, Y. (2015). Multiple kernel learning with random effects for predicting longitudinal outcomes and data integration. *Biometrics* **71**:918–928.

Cheung, H. *et al.* (2015). A Longitudinal Support Vector Regression for Prediction of ALS Score. In: *Bioinformatics and Biomedicine (BIBM), 2015 IEEE International Conference on*. IEEE, pp. 1586–1590.

Comfort, A. (1964). *Ageing. The Biology of Senescence.* London: Routledge & Kegan Paul Ltd. Eev. ed.

Cristianini, N. and Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge university press.

De La Iglesia, B. (2013). Evolutionary computation for feature selection in classification problems. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **3**:381–407.

Demšar, J. (2006). Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research* **7**:1–30.

Ding, C. and Peng, H. (2005). Minimum redundancy feature selection from microarray gene expression data. *Journal of bioinformatics and computational biology* **3**:185–205.

Dytham, C. (2011). *Choosing and Using Statistics: A Biologist's Guide*. John Wiley & Sons.

Elomaa, T. and Kääriäinen, M. (2001). An analysis of reduced error pruning. *Journal of Artificial Intelligence Research* **15**:163–187.

Frank, E. (2000). *Pruning Decision Trees and Lists*. University of Waikato, Hamilton, New Zealand.

Freitas, A.A. (2013). Comprehensible Classification Models – a position paper. *ACM SIGKDD Explorations Newsletter* **15**:1–10.

Friedman, M. (1940). A Comparison of alternative tests of significance for the problem of m rankings. *Annals of Mathematical Statistics* **11**:86–92.

Furlanello, C., Merler, S. and Jurman, G. (2006). Combining feature selection and DTW for time-varying functional genomics. *IEEE Transactions on Signal Processing* **54**:2436–2443.

García, S. *et al.* (2013). A survey of discretization techniques: Taxonomy and empirical analysis in supervised learning. *IEEE Transactions on Knowledge and Data Engineering* **25**:734–750.

García, V., Sánchez, J.S. and Mollineda, R.A. (2012). On the effectiveness of preprocessing methods when dealing with different levels of class imbalance. *Knowledge-Based Systems* **25**:13–21.

Goldberg, D.E. (1989). Genetic Algorithms in Search, Optimization and Machine Learning, 1989. *Reading: Addison-Wesley*.

Guo, X. *et al.* (2008). On the class imbalance problem. In: *Proceedings - 4th International Conference on Natural Computation, ICNC 2008*. IEEE, pp. 192–201.

Hall, M. (1999). *Correlation-Based Feature Selection for Machine Learning*. University of Waikato, Hamilton, New Zealand.

Hall, M. *et al.* (2009). The WEKA data mining software. *ACM SIGKDD*

Explorations Newsletter **11**:10–18.

Hall, M.A. (2000). Feature Selection for Discrete and Numeric Class Machine Learning 1 Introduction. *Machine Learning Proc Seventeenth International conference on Machine Learning*:1–16.

He, H. and Garcia, E.A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering* **21**:1263–1284.

Hofer, S.M. and Sliwinski, M.J. (2001). Understanding Ageing. *Gerontology* **47**:341–352.

Iman, R.L. and Davenport, J.M. (1980). Approximations of the critical region of the Friedman statistic. *Communications in Statistics - Theory and Methods* **A9**:571–595.

Japkowicz, N. and Shah, M. (2011). *Evaluating Learning Algorithms: A Classification Perspective*. Cambridge University Press.

Jie, B. *et al.* (2017). Temporally constrained group sparse learning for longitudinal data analysis in Alzheimer’s disease. *IEEE Transactions on Biomedical Engineering* **64**:238–249.

John, G.H.G. and Langley, P. (1995). Estimating Continuous Distributions in Bayesian Classifiers. In: *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence. Montreal, Quebec, Canada.* pp. 338--345.

Jungjit, S. *et al.* (2013). Two extensions to multi-label correlation-based feature selection: A case study in bioinformatics. In: *Proceedings of the 2013 IEEE International Conference on Systems, Man and Cybernetics (SMC-2013)*. IEEE Computer Society Conference Publishing Services, pp. 1519–1524.

Jungjit, S. and Freitas, A.A. (2015). A New Genetic Algorithm for Multi-Label Correlation-Based Feature Selection. In: *Proceedings of the 23rd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*. Presses universitaires de Louvain, pp. 285–290.

Kohavi, R. (1995). A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. *International Joint Conference on Artificial Intelligence* **14**:1137–1143.

Lexell, J., Taylor, C.C. and Sjöström, M. (1988). What is the cause of the ageing atrophy?. Total number, size and proportion of different fiber types studied in whole vastus lateralis muscle from 15- to 83-year-old men. *Neurological Sciences* **84**:275–294.

Li, J. *et al.* (2016). Feature Selection: A Data Perspective. *arXiv preprint arXiv:1601.07996*.

Liu, H. *et al.* (2002). Discretization: An enabling technique. *Data Mining and Knowledge Discovery* **6**:393–423.

Liu, H. and H.M. (1998). *Feature Selection for Knowledge Discovery and Data Mining*. Springer US.

Liu, P. *et al.* (2010). Classifying skewed data streams based on reusing data. In: *Computer Application and System Modeling (ICCASM), 2010 International Conference on*. IEEE, pp. V4--90.

Longadge, R., Dongre, S.S. and Malik, L. (2013). Class imbalance problem in data mining: review. *International Journal of Computer Science and Network* **2**:83–87.

Lou, Q. and Obradovic, Z. (2012). Analysis of temporal high-dimensional gene expression data for identifying informative biomarker candidates. In: *Proceedings - IEEE International Conference on Data Mining, ICDM*. IEEE, pp. 996–1001.

Marmot, M. *et al.* (2016). *English Longitudinal Study of Ageing: Waves 0-7, 1998-2015* [Online]. Available at: https://www.elsa-project.ac.uk/.

Minhas, S. *et al.* (2015). Early alzheimer’s disease prediction in machine learning setup: Empirical analysis with missing value computation. In: *International Conference on Intelligent Data Engineering and Automated Learning*. Springer, pp. 424–432.

Narendra, P.M. and Fukunaga, K. (1977). A Branch and Bound Algorithm for Feature Subset Selection. *IEEE Transactions on Computers* **C-26**:917–922.

Nemenyi, P. (1962). Distribution-free multiple comparisons. *Biometrics* **18**:263.

Peng, H., Long, F. and Ding, C. (2005). Feature selection based on mutual information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **27**:1226–1238.

Polikar and Robi (2006). Ensemble based systems in decision making. *Circuits and systems magazine, IEEE*:21–45.

Pomsuwan, T. and Freitas, A.A. (2017). Feature selection for the classification of longitudinal human ageing data. In: *Proceedings of ICDM ARIAL 2017: 1st Workshop on Data Mining for Aging; Rehabilitation and Independent Assisted Living (ARIAL); in Conjunction with the IEEE International Conference on Data Mining (ICDM'2017)*. IEEE Computer Society Press.

Quinlan, J. (1993). C4. 5: programs for machine learning. *San Mateo, CA: Morgan Kaufmann*.

Quinlan, J.R. (1986). Induction of Decision Trees. *Machine Learning* **1**:81–106.

Radovic, M. *et al.* (2017). Minimum redundancy maximum relevance feature selection approach for temporal gene expression data. *BMC Bioinformatics* **18**:9.

Ribeiro, C.E. *et al.* (2017). A revision and analysis of the comprehensiveness of the main longitudinal studies of human aging for data mining research. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* **7**:e1202.

Rokach, L. (2016). Decision forest: Twenty years of research. *Information Fusion* **27**:111–125.

Rokach, L. (2010). Ensemble-based classifiers. *Artificial Intelligence Review* **33**:1–39.

Su, J. and Zhang, H. (2006). A Fast Decision Tree Learning Algorithm. In: *Proceedings of the 21st National Conference on Artificial Intelligence*. pp. 500–505.

Sulzmann, J.-N., Fürnkranz, J. and Hüllermeier, E. (2007). On Pairwise Naive Bayes Classifiers. In: *Proceedings of the 2007 European Conference on Machine Learning*. Springer, pp. 371–381.

Tabachnick, B.G. and Fidell, L.S. (2013). *Using Multivariate Statistics*. Pearson Education.

Touw, W.G. *et al.* (2013). Data mining in the life science swith random forest: A walk in the park or lost in the jungle? *Briefings in Bioinformatics* **14**:315–326.

United Nations, Department of Economic and Social Affairs and Population Division (2015). World Population Ageing 2015. *(ST/ESA/SER.A/390)*.

Wang, L., Wang, Y. and Chang, Q. (2016). Feature selection methods for big data bioinformatics: A survey from the search perspective. *Methods* **111**:21–31.

Wilcoxon, F. (1945). Individual Comparisons by Ranking Methods. *Biometrics Bulletin* **1**:80.

Xue, J.H. and Titterington, D.M. (2008). Comment on ‘on discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes’. *Neural Processing Letters* **28**:169–187.

Yang, J. and Honavar, V. (1998). Feature subset selection using a genetic algorithm. *IEEE Intelligent Systems* **13**:44–49.

APPENDIX

The appendix is organized as follows. Section A reports the detailed results of predictive accuracy for the J48 decision tree induction algorithm. Section B shows the detailed results of predictive accuracy for the Naïve Bayes algorithm. Section C reports the most relevant features selected by J48 for each disease class.

In Sections A and B, by "detailed results of predictive accuracy" it is meant the separate values of precision, recall and F-measure for each of the two class labels (presence or absence of disease). Each of these sections reports these results in six tables, one for each of six different feature selection approaches, namely the baseline approach of not performing any feature selection in a pre-processing phase (i.e. giving all features to the classification algorithm), standard Correlation-based Feature Selection (CFS), and the four CFS variants proposed in this work. Hence, these detailed tables of results complement the summarized tables of predictive accuracy results reported in Chapter 4, where the analysis of the results was performed in terms of the average F-measure value across the two class labels for each dataset.

A.DETAILED RESULTS FOR THE J48 ALGORITHM

Appendix

Table A.1: predictive accuracies obtained from J48 without feature selection

Disease	Class Distribution	Class label = 0 (No)			Class label = 1 (Yes)		
		Precision	Recall	F-measure	Precision	Recall	F-measure
HeartAtt	7061 to 435	0.942	1.000	0.970	0.000	0.000	0.000
	1740 to 435	0.949	0.964	0.956	0.208	0.152	0.175
	870 to 435	0.959	0.870	0.912	0.159	0.398	**0.227**
	435 to 435	0.971	0.698	0.813	0.120	0.664	0.203
Angina	7263 to 233	0.969	1.000	0.984	0.000	0.000	0.000
	932 to 233	0.970	0.985	0.977	0.068	0.034	0.046
	466 to 233	0.972	0.893	0.931	0.058	0.206	**0.090**
	233 to 233	0.978	0.585	0.732	0.043	0.584	0.080
Stroke	7094 to 402	0.946	1.000	0.972	0.000	0.000	0.000
	1608 to 402	0.950	0.973	0.962	0.180	0.104	0.132
	804 to 402	0.958	0.877	0.916	0.131	0.326	**0.187**
	402 to 402	0.974	0.670	0.794	0.105	0.679	0.181
Diabetes	6552 to 944	0.928	0.987	0.957	0.840	0.472	0.605
	3776 to 944	0.937	0.978	0.957	0.781	0.544	0.642
	1888 to 944	0.947	0.953	0.950	0.659	0.632	**0.645**
	944 to 944	0.962	0.885	0.922	0.487	0.760	0.594
HBP	4438 to 3058	0.686	0.815	0.745	0.631	0.459	0.531
	3058 to 3058	0.746	0.643	0.690	0.568	0.682	**0.620**
Dementia	7360 to 136	0.982	1.000	0.991	0.000	0.000	0.000
	544 to 136	0.984	0.974	0.979	0.078	0.118	0.094
	272 to 136	0.989	0.835	0.906	0.052	0.493	**0.095**
	136 to 136	0.990	0.754	0.856	0.042	0.581	0.078
Cataract	5344 to 2150	0.771	0.872	0.818	0.528	0.357	0.426
	4300 to 2150	0.789	0.828	0.808	0.512	0.449	0.479
	2150 to 2150	0.852	0.666	0.748	0.462	0.711	**0.560**
Arthritis	4398 to 3098	0.650	0.832	0.730	0.604	0.362	0.453
	3098 to 3098	0.684	0.666	0.675	0.543	0.564	**0.554**
Osteoporosis	6796 to 700	0.907	1.000	0.951	0.000	0.000	0.000
	2800 to 700	0.910	0.976	0.941	0.202	0.060	0.093
	1400 to 700	0.927	0.871	0.898	0.212	0.337	0.260
	700 to 700	0.959	0.601	0.739	0.162	0.749	**0.266**
Parkinsons	7433 to 63	0.992	1.000	0.996	0.000	0.000	0.000
	252 to 63	0.992	0.941	0.966	0.018	0.127	0.031
	126 to 63	0.994	0.766	0.865	0.017	0.476	**0.033**
	63 to 63	0.997	0.597	0.747	0.017	0.810	**0.033**

Appendix

Table A.2: predictive accuracies obtained from J48 with the standard CFS

Disease	Class Distribution	Class label = 0 (No)			Class label = 1 (Yes)		
		Precision	Recall	F-measure	Precision	Recall	F-measure
HeartAtt	7061 to 435	0.942	1.000	0.970	0.000	0.000	0.000
	1740 to 435	0.949	0.968	0.958	0.223	0.149	0.179
	870 to 435	0.956	0.875	0.914	0.146	0.347	**0.205**
	435 to 435	0.973	0.689	0.806	0.119	0.685	0.203
Angina	7263 to 233	0.969	1.000	0.984	0.000	0.000	0.000
	932 to 233	0.969	0.983	0.976	0.040	0.021	0.028
	466 to 233	0.972	0.881	0.924	0.051	0.197	**0.081**
	233 to 233	0.976	0.597	0.741	0.042	0.549	0.078
Stroke	7094 to 402	0.946	1.000	0.972	0.000	0.000	0.000
	1608 to 402	0.950	0.979	0.964	0.192	0.087	0.120
	804 to 402	0.960	0.889	0.923	0.152	0.351	**0.212**
	402 to 402	0.970	0.700	0.813	0.105	0.622	0.180
Diabetes	6552 to 944	0.928	0.988	0.957	0.845	0.467	0.602
	3776 to 944	0.931	0.983	0.956	0.807	0.495	0.613
	1888 to 944	0.944	0.961	0.953	0.692	0.607	**0.647**
	944 to 944	0.964	0.857	0.908	0.440	0.776	0.561
HBP	4438 to 3058	0.696	0.808	0.748	0.636	0.487	0.552
	3058 to 3058	0.745	0.649	0.694	0.571	0.677	**0.620**
Dementia	7360 to 136	0.982	1.000	0.991	0.000	0.000	0.000
	544 to 136	0.985	0.971	0.978	0.100	0.176	**0.128**
	272 to 136	0.988	0.880	0.931	0.063	0.434	0.109
	136 to 136	0.991	0.731	0.842	0.043	0.654	0.081
Cataract	5344 to 2150	0.785	0.849	0.816	0.530	0.423	0.470
	4300 to 2150	0.791	0.839	0.814	0.529	0.448	0.485
	2150 to 2150	0.857	0.641	0.733	0.451	0.734	0.559
Arthritis	4398 to 3098	0.653	0.835	0.733	0.612	0.370	0.461
	3098 to 3098	0.692	0.636	0.663	0.536	0.598	**0.565**
Osteoporosis	6796 to 700	0.907	1.000	0.951	0.000	0.000	0.000
	2800 to 700	0.912	0.972	0.941	0.251	0.090	0.132
	1400 to 700	0.933	0.859	0.895	0.226	0.399	**0.288**
	700 to 700	0.959	0.611	0.746	0.165	0.744	0.269
Parkinsons	7433 to 63	0.992	1.000	0.996	0.000	0.000	0.000
	252 to 63	0.992	0.962	0.977	0.024	0.111	**0.040**
	126 to 63	0.995	0.764	0.864	0.019	0.540	0.037
	63 to 63	0.997	0.601	0.750	0.016	0.778	0.032

Table A.3: predictive accuracies obtained from J48 with Exh-CFS-Gr

Disease	Class Distribution	Class label = 0 (No)			Class label = 1 (Yes)		
		Precision	Recall	F-measure	Precision	Recall	F-measure
HeartAtt	7061 to 435	0.942	1.000	0.970	0.000	0.000	0.000
	1740 to 435	0.948	0.966	0.957	0.205	0.140	0.166
	870 to 435	0.958	0.864	0.909	0.151	0.391	**0.218**
	435 to 435	0.970	0.703	0.815	0.119	0.653	0.202
Angina	7263 to 233	0.969	1.000	0.984	0.000	0.000	0.000
	932 to 233	0.969	0.989	0.979	0.036	0.013	0.019
	466 to 233	0.973	0.883	0.926	0.058	0.223	**0.092**
	233 to 233	0.978	0.575	0.724	0.043	0.597	0.080
Stroke	7094 to 402	0.946	1.000	0.972	0.000	0.000	0.000
	1608 to 402	0.951	0.972	0.961	0.183	0.109	0.137
	804 to 402	0.962	0.887	0.923	0.159	0.376	**0.223**
	402 to 402	0.976	0.658	0.786	0.105	0.711	0.184
Diabetes	6552 to 944	0.929	0.988	0.958	0.852	0.477	0.611
	3776 to 944	0.934	0.979	0.956	0.780	0.521	**0.625**
	1888 to 944	0.943	0.953	0.948	0.648	0.602	0.624
	944 to 944	0.960	0.856	0.905	0.431	0.755	0.548
HBP	4438 to 3058	0.698	0.784	0.738	0.618	0.507	0.557
	3058 to 3058	0.759	0.639	0.694	0.574	0.705	**0.633**
Dementia	7360 to 136	0.982	1.000	0.991	0.000	0.000	0.000
	544 to 136	0.984	0.974	0.979	0.082	0.125	**0.099**
	272 to 136	0.988	0.864	0.922	0.055	0.426	0.097
	136 to 136	0.990	0.753	0.855	0.042	0.581	0.078
Cataract	5344 to 2150	0.771	0.876	0.820	0.533	0.353	0.424
	4300 to 2150	0.792	0.832	0.812	0.522	0.456	0.487
	2150 to 2150	0.857	0.650	0.739	0.456	0.730	**0.561**
Arthritis	4398 to 3098	0.653	0.819	0.726	0.597	0.381	0.465
	3098 to 3098	0.686	0.645	0.665	0.535	0.581	**0.557**
Osteoporosis	6796 to 700	0.907	1.000	0.951	0.000	0.000	0.000
	2800 to 700	0.912	0.972	0.941	0.252	0.090	0.133
	1400 to 700	0.928	0.877	0.902	0.220	0.337	0.266
	700 to 700	0.960	0.608	0.745	0.166	0.757	**0.272**
Parkinsons	7433 to 63	0.992	1.000	0.996	0.000	0.000	0.000
	252 to 63	0.991	0.988	0.990	0.000	0.000	0.000
	126 to 63	0.995	0.756	0.859	0.019	0.556	**0.037**
	63 to 63	0.997	0.600	0.749	0.016	0.762	0.031

Table A.4: predictive accuracies obtained from J48 with Exh-CFS-Gr+CFS

Disease	Class Distribution	Class label = 0 (No)			Class label = 1 (Yes)		
		Precision	Recall	F-measure	Precision	Recall	F-measure
HeartAtt	7061 to 435	0.942	1.000	0.970	0.000	0.000	0.000
	1740 to 435	0.949	0.968	0.958	0.223	0.149	0.179
	870 to 435	0.957	0.873	0.913	0.148	0.359	**0.210**
	435 to 435	0.973	0.689	0.807	0.120	0.690	0.205
Angina	7263 to 233	0.969	1.000	0.984	0.000	0.000	0.000
	932 to 233	0.969	0.983	0.976	0.040	0.021	0.028
	466 to 233	0.972	0.883	0.925	0.052	0.202	**0.083**
	233 to 233	0.976	0.597	0.741	0.042	0.549	0.078
Stroke	7094 to 402	0.946	1.000	0.972	0.000	0.000	0.000
	1608 to 402	0.950	0.978	0.964	0.188	0.090	0.121
	804 to 402	0.960	0.893	0.926	0.155	0.346	**0.214**
	402 to 402	0.970	0.702	0.815	0.106	0.622	0.181
Diabetes	6552 to 944	0.928	0.988	0.957	0.846	0.466	0.601
	3776 to 944	0.932	0.983	0.957	0.807	0.502	0.619
	1888 to 944	0.942	0.959	0.951	0.677	0.590	**0.630**
	944 to 944	0.963	0.857	0.907	0.438	0.771	0.558
HBP	4438 to 3058	0.694	0.807	0.747	0.634	0.484	0.549
	3058 to 3058	0.746	0.653	0.696	0.573	0.677	**0.621**
Dementia	7360 to 136	0.982	1.000	0.991	0.000	0.000	0.000
	544 to 136	0.985	0.971	0.978	0.100	0.176	**0.128**
	272 to 136	0.988	0.880	0.931	0.063	0.434	0.109
	136 to 136	0.991	0.731	0.842	0.043	0.654	0.081
Cataract	5344 to 2150	0.785	0.849	0.816	0.529	0.421	0.469
	4300 to 2150	0.790	0.840	0.814	0.528	0.445	0.483
	2150 to 2150	0.856	0.641	0.733	0.451	0.733	**0.558**
Arthritis	4398 to 3098	0.652	0.822	0.727	0.599	0.378	0.463
	3098 to 3098	0.695	0.628	0.660	0.535	0.608	**0.569**
Osteoporosis	6796 to 700	0.907	1.000	0.951	0.000	0.000	0.000
	2800 to 700	0.912	0.972	0.941	0.251	0.090	0.132
	1400 to 700	0.933	0.860	0.895	0.226	0.399	**0.289**
	700 to 700	0.959	0.611	0.746	0.165	0.744	0.269
Parkinsons	7433 to 63	0.992	1.000	0.996	0.000	0.000	0.000
	252 to 63	0.992	0.962	0.977	0.024	0.111	**0.040**
	126 to 63	0.995	0.764	0.864	0.019	0.540	0.037
	63 to 63	0.997	0.601	0.750	0.016	0.778	0.032

Table A.5: predictive accuracies obtained from J48 with Co-WR-CFS

Disease	Class Distribution	Class label = 0 (No)			Class label = 1 (Yes)		
		Precision	Recall	F-measure	Precision	Recall	F-measure
HeartAtt	7061 to 435	0.942	1.000	0.970	0.000	0.000	0.000
	1740 to 435	0.948	0.967	0.957	0.195	0.131	0.157
	870 to 435	0.959	0.864	0.909	0.154	0.402	**0.223**
	435 to 435	0.972	0.679	0.799	0.115	0.678	0.197
Angina	7263 to 233	0.969	1.000	0.984	0.000	0.000	0.000
	932 to 233	0.969	0.988	0.978	0.053	0.021	0.031
	466 to 233	0.973	0.899	0.934	0.066	0.223	**0.102**
	233 to 233	0.978	0.582	0.730	0.043	0.592	0.081
Stroke	7094 to 402	0.946	1.000	0.972	0.000	0.000	0.000
	1608 to 402	0.951	0.976	0.964	0.220	0.119	0.155
	804 to 402	0.960	0.867	0.911	0.135	0.366	**0.197**
	402 to 402	0.977	0.664	0.791	0.109	0.729	0.190
Diabetes	6552 to 944	0.928	0.988	0.957	0.844	0.465	0.600
	3776 to 944	0.931	0.983	0.956	0.807	0.496	0.614
	1888 to 944	0.942	0.960	0.951	0.681	0.592	**0.633**
	944 to 944	0.961	0.842	0.897	0.409	0.761	0.532
HBP	4438 to 3058	0.683	0.787	0.731	0.603	0.469	0.528
	3058 to 3058	0.745	0.617	0.675	0.555	0.693	**0.616**
Dementia	7360 to 136	0.982	1.000	0.991	0.000	0.000	0.000
	544 to 136	0.985	0.972	0.978	0.106	0.176	**0.132**
	272 to 136	0.989	0.870	0.926	0.062	0.463	0.109
	136 to 136	0.992	0.708	0.826	0.041	0.676	0.077
Cataract	5344 to 2150	0.789	0.844	0.816	0.531	0.440	0.481
	4300 to 2150	0.793	0.836	0.814	0.529	0.459	0.492
	2150 to 2150	0.853	0.655	0.741	0.456	0.720	**0.558**
Arthritis	4398 to 3098	0.653	0.835	0.733	0.613	0.371	0.462
	3098 to 3098	0.694	0.620	0.655	0.531	0.612	**0.569**
Osteoporosis	6796 to 700	0.907	1.000	0.951	0.000	0.000	0.000
	2800 to 700	0.911	0.977	0.943	0.243	0.073	0.112
	1400 to 700	0.929	0.860	0.893	0.211	0.363	0.267
	700 to 700	0.956	0.633	0.762	0.167	0.716	**0.271**
Parkinsons	7433 to 63	0.992	1.000	0.996	0.000	0.000	0.000
	252 to 63	0.992	0.959	0.975	0.022	0.111	**0.037**
	126 to 63	0.995	0.762	0.863	0.019	0.540	0.036
	63 to 63	0.997	0.598	0.748	0.016	0.778	0.032

Appendix

Table A.6: predictive accuracies obtained from J48 with Fi-WR-CFS

Disease	Class Distribution	Class label = 0 (No)			Class label = 1 (Yes)		
		Precision	Recall	F-measure	Precision	Recall	F-measure
HeartAtt	7061 to 435	0.942	1.000	0.970	0.000	0.000	0.000
	1740 to 435	0.949	0.964	0.957	0.223	0.168	0.191
	870 to 435	0.956	0.860	0.906	0.138	0.363	0.200
	435 to 435	0.972	0.690	0.807	0.119	0.683	**0.203**
Angina	7263 to 233	0.969	1.000	0.984	0.000	0.000	0.000
	932 to 233	0.969	0.983	0.976	0.053	0.030	0.038
	466 to 233	0.973	0.889	0.929	0.060	0.223	**0.095**
	233 to 233	0.978	0.592	0.737	0.044	0.592	0.083
Stroke	7094 to 402	0.946	1.000	0.972	0.000	0.000	0.000
	1608 to 402	0.951	0.974	0.962	0.191	0.109	0.139
	804 to 402	0.960	0.875	0.916	0.139	0.356	**0.200**
	402 to 402	0.976	0.662	0.789	0.107	0.716	0.186
Diabetes	6552 to 944	0.928	0.987	0.957	0.841	0.465	0.599
	3776 to 944	0.932	0.981	0.956	0.793	0.502	0.615
	1888 to 944	0.946	0.960	0.953	0.689	0.617	**0.651**
	944 to 944	0.963	0.874	0.916	0.467	0.765	0.580
HBP	4438 to 3058	0.699	0.810	0.750	0.642	0.494	0.559
	3058 to 3058	0.746	0.651	0.696	0.573	0.679	**0.621**
Dementia	7360 to 136	0.982	1.000	0.991	0.000	0.000	0.000
	544 to 136	0.984	0.972	0.978	0.087	0.147	**0.110**
	272 to 136	0.988	0.879	0.930	0.063	0.441	**0.110**
	136 to 136	0.991	0.723	0.836	0.041	0.640	0.077
Cataract	5344 to 2150	0.789	0.836	0.811	0.520	0.443	0.479
	4300 to 2150	0.798	0.824	0.811	0.524	0.481	0.502
	2150 to 2150	0.854	0.653	0.740	0.456	0.723	**0.559**
Arthritis	4398 to 3098	0.657	0.815	0.727	0.600	0.395	0.476
	3098 to 3098	0.690	0.625	0.656	0.530	0.601	**0.563**
Osteoporosis	6796 to 700	0.907	1.000	0.951	0.000	0.000	0.000
	2800 to 700	0.913	0.967	0.939	0.241	0.101	0.143
	1400 to 700	0.935	0.852	0.892	0.228	0.423	**0.296**
	700 to 700	0.957	0.627	0.758	0.167	0.724	0.271
Parkinsons	7433 to 63	0.992	1.000	0.996	0.000	0.000	0.000
	252 to 63	0.992	0.959	0.975	0.022	0.111	**0.037**
	126 to 63	0.995	0.762	0.863	0.019	0.540	0.036
	63 to 63	0.997	0.592	0.743	0.016	0.778	0.031

B. DETAILED RESULTS FOR NAÏVE BAYES ALGORITHM

Appendix

Table B.1: predictive accuracies obtained from NB without feature selection

Disease	Class Distribution	Class label = 0 (No)			Class label = 1 (Yes)		
		Precision	Recall	F-measure	Precision	Recall	F-measure
HeartAtt	7061 to 435	0.967	0.784	0.866	0.140	0.572	0.225
	1740 to 435	0.967	0.802	0.877	0.147	0.554	**0.233**
	870 to 435	0.968	0.773	0.860	0.136	0.579	0.220
	435 to 435	0.969	0.745	0.842	0.128	0.609	0.212
Angina	7263 to 233	0.977	0.817	0.890	0.066	0.403	**0.114**
	932 to 233	0.977	0.772	0.863	0.059	0.442	0.103
	466 to 233	0.978	0.776	0.866	0.062	0.459	0.109
	233 to 233	0.978	0.728	0.835	0.055	0.489	0.098
Stroke	7094 to 402	0.968	0.776	0.861	0.120	0.540	**0.197**
	1608 to 402	0.967	0.770	0.857	0.116	0.535	0.191
	804 to 402	0.968	0.749	0.844	0.114	0.570	0.190
	402 to 402	0.968	0.727	0.831	0.107	0.580	0.181
Diabetes	6552 to 944	0.957	0.864	0.908	0.435	0.729	**0.545**
	3776 to 944	0.960	0.848	0.901	0.417	0.752	0.536
	1888 to 944	0.961	0.838	0.896	0.406	0.767	0.531
	944 to 944	0.958	0.840	0.895	0.401	0.745	0.522
HBP	4438 to 3058	0.745	0.692	0.717	0.594	0.656	**0.623**
	3058 to 3058	0.743	0.699	0.720	0.598	0.649	0.622
Dementia	7360 to 136	0.991	0.826	0.901	0.060	0.596	**0.108**
	544 to 136	0.991	0.817	0.896	0.056	0.581	0.101
	272 to 136	0.991	0.785	0.876	0.050	0.610	0.092
	136 to 136	0.991	0.742	0.849	0.043	0.625	0.080
Cataract	5344 to 2150	0.831	0.656	0.734	0.439	0.669	**0.530**
	4300 to 2150	0.833	0.636	0.721	0.430	0.684	0.528
	2150 to 2150	0.825	0.654	0.729	0.432	0.654	0.520
Arthritis	4398 to 3098	0.707	0.557	0.623	0.517	0.673	**0.585**
	3098 to 3098	0.705	0.570	0.630	0.520	0.661	0.582
Osteoporosis	6796 to 700	0.950	0.637	0.762	0.161	0.677	0.260
	2800 to 700	0.954	0.619	0.751	0.161	0.710	**0.263**
	1400 to 700	0.955	0.602	0.739	0.158	0.726	0.260
	700 to 700	0.958	0.596	0.735	0.160	0.747	**0.263**
Parkinsons	7433 to 63	0.993	0.835	0.907	0.014	0.286	0.027
	252 to 63	0.994	0.828	0.904	0.021	0.444	**0.041**
	126 to 63	0.994	0.784	0.876	0.017	0.444	0.033
	63 to 63	0.994	0.684	0.811	0.014	0.540	0.028

Appendix

Table B.2: predictive accuracies obtained from NB with standard CFS

Disease	Class Distribution	Class label = 0 (No)			Class label = 1 (Yes)		
		Precision	Recall	F‑measure	Precision	Recall	F‑measure
HeartAtt	7061 to 435	0.960	0.895	0.926	0.186	0.389	0.252
	1740 to 435	0.967	0.847	0.903	0.178	0.538	**0.268**
	870 to 435	0.970	0.789	0.870	0.151	0.609	0.242
	435 to 435	0.975	0.712	0.823	0.130	0.699	0.219
Angina	7263 to 233	0.971	0.985	0.978	0.152	0.086	0.110
	932 to 233	0.977	0.838	0.902	0.072	0.391	**0.121**
	466 to 233	0.979	0.776	0.866	0.065	0.489	0.115
	233 to 233	0.979	0.679	0.802	0.052	0.554	0.096
Stroke	7094 to 402	0.949	0.988	0.968	0.200	0.055	0.086
	1608 to 402	0.956	0.926	0.941	0.163	0.254	**0.199**
	804 to 402	0.963	0.830	0.892	0.128	0.440	**0.199**
	402 to 402	0.972	0.710	0.821	0.112	0.644	0.191
Diabetes	6552 to 944	0.938	0.966	0.952	0.704	0.555	0.621
	3776 to 944	0.942	0.956	0.949	0.658	0.590	**0.622**
	1888 to 944	0.949	0.929	0.939	0.568	0.651	0.607
	944 to 944	0.956	0.892	0.923	0.487	0.714	0.579
HBP	4438 to 3058	0.743	0.758	0.751	0.639	0.619	0.629
	3058 to 3058	0.754	0.735	0.745	0.629	0.652	**0.641**
Dementia	7360 to 136	0.985	0.971	0.978	0.108	0.191	0.138
	544 to 136	0.988	0.928	0.957	0.086	0.368	**0.139**
	272 to 136	0.989	0.895	0.940	0.077	0.478	0.133
	136 to 136	0.990	0.824	0.899	0.053	0.529	0.096
Cataract	5344 to 2150	0.815	0.772	0.793	0.499	0.565	0.530
	4300 to 2150	0.822	0.757	0.789	0.496	0.593	0.540
	2150 to 2150	0.846	0.701	0.767	0.479	0.683	**0.563**
Arthritis	4398 to 3098	0.696	0.687	0.691	0.563	0.573	0.568
	3098 to 3098	0.700	0.643	0.670	0.546	0.609	**0.576**
Osteoporosis	6796 to 700	0.926	0.844	0.883	0.187	0.347	0.243
	2800 to 700	0.953	0.671	0.788	0.175	0.676	**0.278**
	1400 to 700	0.958	0.629	0.760	0.169	0.734	0.275
	700 to 700	0.965	0.542	0.695	0.154	0.811	0.259
Parkinsons	7433 to 63	0.992	0.994	0.993	0.000	0.000	0.000
	252 to 63	0.992	0.952	0.972	0.014	0.079	0.024
	126 to 63	0.994	0.851	0.917	0.021	0.381	**0.040**
	63 to 63	0.995	0.685	0.811	0.015	0.571	0.029

Table B.3: predictive accuracies obtained from NB with Exh-CFS-Gr

Disease	Class Distribution	Class label = 0 (No)			Class label = 1 (Yes)		
		Precision	Recall	F-measure	Precision	Recall	F-measure
HeartAtt	7061 to 435	0.966	0.830	0.893	0.161	0.529	0.246
	1740 to 435	0.971	0.807	0.881	0.161	0.605	**0.255**
	870 to 435	0.973	0.769	0.859	0.148	0.651	0.241
	435 to 435	0.973	0.727	0.832	0.133	0.678	0.222
Angina	7263 to 233	0.977	0.871	0.921	0.083	0.365	**0.135**
	932 to 233	0.980	0.778	0.867	0.067	0.498	0.118
	466 to 233	0.981	0.750	0.850	0.065	0.545	0.117
	233 to 233	0.982	0.685	0.807	0.058	0.605	0.106
Stroke	7094 to 402	0.965	0.829	0.892	0.135	0.470	**0.210**
	1608 to 402	0.970	0.766	0.856	0.123	0.580	0.203
	804 to 402	0.973	0.731	0.835	0.119	0.642	0.201
	402 to 402	0.975	0.699	0.814	0.114	0.687	0.196
Diabetes	6552 to 944	0.954	0.882	0.917	0.463	0.706	**0.559**
	3776 to 944	0.955	0.869	0.910	0.440	0.714	0.544
	1888 to 944	0.958	0.852	0.902	0.420	0.740	0.536
	944 to 944	0.961	0.831	0.891	0.395	0.765	0.521
HBP	4438 to 3058	0.727	0.750	0.738	0.619	0.591	0.605
	3058 to 3058	0.736	0.736	0.736	0.617	0.617	**0.617**
Dementia	7360 to 136	0.988	0.900	0.942	0.072	0.419	**0.123**
	544 to 136	0.991	0.855	0.918	0.069	0.581	**0.123**
	272 to 136	0.992	0.820	0.898	0.063	0.654	0.115
	136 to 136	0.992	0.769	0.866	0.050	0.662	0.094
Cataract	5344 to 2150	0.831	0.714	0.768	0.474	0.639	**0.544**
	4300 to 2150	0.833	0.702	0.762	0.467	0.650	**0.544**
	2150 to 2150	0.839	0.664	0.741	0.450	0.682	0.542
Arthritis	4398 to 3098	0.704	0.659	0.681	0.556	0.607	0.580
	3098 to 3098	0.706	0.638	0.670	0.548	0.623	**0.583**
Osteoporosis	6796 to 700	0.952	0.684	0.796	0.179	0.667	**0.282**
	2800 to 700	0.957	0.646	0.771	0.173	0.719	0.279
	1400 to 700	0.961	0.606	0.743	0.166	0.761	0.273
	700 to 700	0.965	0.570	0.717	0.161	0.800	0.268
Parkinsons	7433 to 63	0.992	0.987	0.989	0.010	0.016	0.012
	252 to 63	0.993	0.853	0.918	0.017	0.302	0.032
	126 to 63	0.994	0.771	0.869	0.017	0.476	**0.033**
	63 to 63	0.995	0.654	0.790	0.016	0.651	0.031

Table B.4: predictive accuracies obtained from NB with Exh-CFS-Gr+CFS

Disease	Class Distribution	Class label = 0 (No)			Class label = 1 (Yes)		
		Precision	Recall	F-measure	Precision	Recall	F-measure
HeartAtt	7061 to 435	0.960	0.895	0.926	0.186	0.389	0.252
	1740 to 435	0.967	0.848	0.904	0.178	0.533	**0.267**
	870 to 435	0.970	0.789	0.870	0.150	0.605	0.241
	435 to 435	0.973	0.710	0.821	0.127	0.685	0.214
Angina	7263 to 233	0.971	0.985	0.978	0.144	0.077	0.101
	932 to 233	0.977	0.838	0.902	0.072	0.391	**0.121**
	466 to 233	0.979	0.772	0.864	0.064	0.485	0.113
	233 to 233	0.979	0.678	0.801	0.052	0.554	0.096
Stroke	7094 to 402	0.948	0.987	0.967	0.182	0.050	0.078
	1608 to 402	0.957	0.928	0.942	0.171	0.261	**0.207**
	804 to 402	0.963	0.830	0.891	0.127	0.435	0.196
	402 to 402	0.972	0.711	0.821	0.111	0.637	0.189
Diabetes	6552 to 944	0.938	0.967	0.952	0.708	0.556	**0.623**
	3776 to 944	0.940	0.955	0.947	0.647	0.578	0.611
	1888 to 944	0.946	0.928	0.937	0.560	0.636	0.595
	944 to 944	0.955	0.882	0.917	0.463	0.709	0.560
HBP	4438 to 3058	0.743	0.758	0.750	0.638	0.619	0.628
	3058 to 3058	0.755	0.734	0.744	0.629	0.654	**0.641**
Dementia	7360 to 136	0.985	0.971	0.978	0.108	0.191	0.138
	544 to 136	0.988	0.928	0.957	0.086	0.368	**0.139**
	272 to 136	0.989	0.895	0.940	0.076	0.471	0.131
	136 to 136	0.990	0.824	0.899	0.053	0.529	0.096
Cataract	5344 to 2150	0.816	0.774	0.795	0.502	0.567	0.533
	4300 to 2150	0.823	0.755	0.788	0.495	0.596	0.541
	2150 to 2150	0.845	0.701	0.766	0.478	0.681	**0.562**
Arthritis	4398 to 3098	0.696	0.688	0.692	0.564	0.574	0.569
	3098 to 3098	0.702	0.642	0.671	0.547	0.612	**0.577**
Osteoporosis	6796 to 700	0.926	0.848	0.885	0.187	0.340	0.241
	2800 to 700	0.953	0.671	0.788	0.175	0.676	**0.278**
	1400 to 700	0.958	0.631	0.761	0.170	0.733	0.276
	700 to 700	0.965	0.542	0.695	0.154	0.811	0.259
Parkinsons	7433 to 63	0.992	0.994	0.993	0.000	0.000	0.000
	252 to 63	0.992	0.952	0.972	0.014	0.079	0.024
	126 to 63	0.994	0.851	0.917	0.021	0.381	**0.040**
	63 to 63	0.995	0.685	0.811	0.016	0.587	0.030

Table B.5: predictive accuracies obtained from NB with Co-WR-CFS

Disease	Class Distribution	Class label = 0 (No)			Class label = 1 (Yes)		
		Precision	Recall	F-measure	Precision	Recall	F-measure
HeartAtt	7061 to 435	0.954	0.934	0.944	0.197	0.262	0.225
	1740 to 435	0.965	0.847	0.902	0.167	0.497	**0.250**
	870 to 435	0.970	0.785	0.868	0.149	0.611	0.240
	435 to 435	0.971	0.703	0.816	0.121	0.664	0.205
Angina	7263 to 233	0.970	0.989	0.979	0.135	0.056	0.079
	932 to 233	0.976	0.821	0.892	0.063	0.378	**0.109**
	466 to 233	0.978	0.752	0.850	0.057	0.468	0.102
	233 to 233	0.980	0.648	0.780	0.052	0.597	0.095
Stroke	7094 to 402	0.949	0.989	0.968	0.222	0.055	0.088
	1608 to 402	0.961	0.872	0.915	0.145	0.383	**0.211**
	804 to 402	0.969	0.763	0.854	0.121	0.575	0.200
	402 to 402	0.972	0.690	0.807	0.106	0.647	0.182
Diabetes	6552 to 944	0.938	0.966	0.951	0.698	0.554	**0.618**
	3776 to 944	0.938	0.957	0.948	0.654	0.563	0.605
	1888 to 944	0.942	0.935	0.939	0.572	0.603	0.587
	944 to 944	0.950	0.897	0.922	0.482	0.669	0.561
HBP	4438 to 3058	0.732	0.726	0.729	0.607	0.614	0.611
	3058 to 3058	0.744	0.702	0.723	0.600	0.650	**0.624**
Dementia	7360 to 136	0.985	0.970	0.977	0.109	0.199	**0.141**
	544 to 136	0.989	0.894	0.939	0.072	0.441	0.123
	272 to 136	0.990	0.856	0.918	0.064	0.529	0.113
	136 to 136	0.990	0.779	0.872	0.047	0.596	0.088
Cataract	5344 to 2150	0.827	0.743	0.783	0.490	0.613	0.545
	4300 to 2150	0.829	0.733	0.778	0.484	0.623	0.545
	2150 to 2150	0.844	0.705	0.768	0.480	0.677	**0.561**
Arthritis	4398 to 3098	0.695	0.662	0.678	0.550	0.587	0.568
	3098 to 3098	0.703	0.593	0.643	0.527	0.645	**0.580**
Osteoporosis	6796 to 700	0.935	0.806	0.866	0.195	0.457	0.274
	2800 to 700	0.949	0.709	0.811	0.182	0.627	**0.282**
	1400 to 700	0.958	0.615	0.749	0.165	0.737	0.269
	700 to 700	0.963	0.536	0.689	0.151	0.801	0.254
Parkinsons	7433 to 63	0.992	0.994	0.993	0.000	0.000	0.000
	252 to 63	0.992	0.951	0.971	0.016	0.095	0.028
	126 to 63	0.994	0.850	0.917	0.021	0.381	**0.040**
	63 to 63	0.995	0.683	0.810	0.015	0.556	0.029

Appendix

Table B.6: predictive accuracies obtained from NB with Fi-WR-CFS

Disease	Class Distribution	Class label = 0 (No)			Class label = 1 (Yes)		
		Precision	Recall	F-measure	Precision	Recall	F-measure
HeartAtt	7061 to 435	0.963	0.879	0.919	0.186	0.448	**0.263**
	1740 to 435	0.969	0.827	0.893	0.169	0.570	0.261
	870 to 435	0.971	0.775	0.862	0.147	0.628	0.238
	435 to 435	0.974	0.717	0.826	0.130	0.685	0.218
Angina	7263 to 233	0.972	0.968	0.970	0.103	0.116	0.109
	932 to 233	0.978	0.802	0.881	0.066	0.433	0.114
	466 to 233	0.981	0.755	0.853	0.066	0.536	**0.117**
	233 to 233	0.981	0.673	0.798	0.054	0.584	0.099
Stroke	7094 to 402	0.949	0.985	0.966	0.182	0.060	0.090
	1608 to 402	0.965	0.830	0.893	0.136	0.470	**0.211**
	804 to 402	0.972	0.740	0.840	0.120	0.627	0.202
	402 to 402	0.974	0.692	0.809	0.111	0.679	0.191
Diabetes	6552 to 944	0.940	0.958	0.949	0.662	0.575	0.616
	3776 to 944	0.949	0.939	0.944	0.605	0.646	**0.625**
	1888 to 944	0.956	0.899	0.926	0.503	0.713	0.590
	944 to 944	0.958	0.873	0.913	0.455	0.737	0.562
HBP	4438 to 3058	0.743	0.756	0.750	0.637	0.620	0.628
	3058 to 3058	0.753	0.738	0.746	0.631	0.649	**0.640**
Dementia	7360 to 136	0.986	0.961	0.974	0.110	0.257	**0.154**
	544 to 136	0.989	0.893	0.939	0.078	0.485	0.134
	272 to 136	0.990	0.853	0.917	0.064	0.544	0.115
	136 to 136	0.991	0.799	0.885	0.054	0.618	0.099
Cataract	5344 to 2150	0.823	0.767	0.794	0.505	0.590	0.544
	4300 to 2150	0.830	0.757	0.792	0.504	0.614	0.553
	2150 to 2150	0.841	0.714	0.772	0.483	0.665	**0.560**
Arthritis	4398 to 3098	0.701	0.657	0.678	0.553	0.602	0.576
	3098 to 3098	0.704	0.625	0.662	0.541	0.626	**0.580**
Osteoporosis	6796 to 700	0.952	0.686	0.797	0.179	0.664	**0.282**
	2800 to 700	0.957	0.634	0.763	0.169	0.724	0.274
	1400 to 700	0.961	0.584	0.726	0.159	0.767	0.264
	700 to 700	0.965	0.542	0.694	0.154	0.807	0.258
Parkinsons	7433 to 63	0.992	0.994	0.993	0.000	0.000	0.000
	252 to 63	0.992	0.951	0.971	0.016	0.095	0.028
	126 to 63	0.994	0.850	0.917	0.021	0.381	**0.040**
	63 to 63	0.995	0.684	0.810	0.015	0.556	0.029

Appendix

C. The Most Relevant Features Selected by J48 for Each Disease Class

Table C.1 shows the most relevant feature selected by the decision tree induction algorithm J48 for each disease class. Recall that the most relevant feature in a decision tree is the root node feature, since this is used to classify all instances. In Table C.1, for each disease, the feature shown in the last column is the root node feature in the decision tree leading to the highest F-measure value, among all decision trees built by J48 for all feature selection approaches and across all class distributions for that disease, as reported earlier in Table 4.5. The definition of the features can be found in Table 3.1. The second column of Table C.1 reports the feature selection approach that produced the corresponding decision tree, where "None" indicates that the "no feature selection" approach was used in a data pre-processing phase (i.e. J48 used all features). For instance, for the disease Heart Attack, we can observe in Table 4.5 that the highest F-measure value (0.61) was obtained by Exh-CFS-Gr+CFS in the class distribution with a ratio of 1 to 1 for the two class labels. Hence, the root node feature reported for this disease in the last column of Table C.1 is the one in the decision tree built when using Exh-CFS-Gr+CFS in a data pre-processing phase and using that class distribution in the training set. For the diseases Dementia and Osteoporosis, there are three and two (respectively) feature selection approaches which are tied in terms of the highest F-measure value. Despite that, there is just one reported feature, since the same feature was consistently selected as the root node for all of the tied models.

Interestingly, for 5 out of the 10 diseases, the most relevant feature overall was "w6indager", which represents the age of the individual (at wave 6). This is not very surprising, considering that all 10 diseases used as classes in our experiments are age-related diseases. However, considering that there is a large number of biomedical variables being used as features, it is perhaps somewhat surprising that age is the most relevant feature in half of the datasets. In addition, another simple feature, "w6indsex", representing the gender of the individual, was selected by J48 as the most relevant feature for predicting osteoporosis, which is consistent with the fact that osteoporosis is known to be more common in females than in males, overall.

Appendix

Regarding the other four diseases, the most relevant features selected by J48 were features directly extracted from the Nurse data in ELSA: in two cases a feature from wave 6 (the most recent wave), for Diabetes and Arthritis; in one case a feature from wave 4, for Angina; and finally in the last case a feature from wave 2, for High Blood Pressure. Hence, none of the constructed longitudinal features was selected by J48 as the most relevant feature for classification.

Table C.1: Feature selected by J48 for the root node of the decision tree, in the tree with highest predictive accuracy for each disease.

Disease	Feature selection approach	Root node feature
Heart attack	Exh-CFS-Gr + CFS	w6indager
Angina	Co-WR-CFS	w4clotb
Stroke	Co-WR-CFS	w6indager
Diabetes	None	w6hba1c
High blood pressure	Exh-CFS-Gr + CFS	w2sysval
Dementia	Standard CFS, Exh-CFS-Gr + CFS, Co-WR-CFS	w6indager
Cataract	None	w6indager
Arthritis	Co-WR-CFS	w6mmgsd_me
Osteoporosis	Standard CFS, Exh-CFS-Gr + CFS	indsex
Parkinson’s disease	None	w6indager