BOOTSTRAP-BASED AGGREGATIONS AND THEIR STABILITY IN FEATURE SELECTION

by

Reem Elfatih Salman

A thesis presented to the Faculty of the
American University of Sharjah
College of Arts and Sciences
in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science in
Mathematics

Sharjah, UAE
June 2022

Declaration of Authorship

I declare that this (project/thesis/dissertation) is my own work and, to the best of my knowledge and belief, it does not contain material published or written by a third party, except where permission has been obtained and/or appropriately cited through full and accurate referencing.

Signature **Reem Elfatih Salman**

Date……**08/06/2022**……

Approval Signatures

We, the undersigned, approve the Master's Thesis of Reem Elfatih Salman

Thesis Title: Bootstrap-based Aggregations and their Stability in Feature Selection

Date of Defense: 16-Jun-2022

Name, Title and Affiliation	**Signature**
Dr. Ayman Alzaatreh Associate Professor, Department of Mathematics & Statistics Thesis Advisor	
Dr. Hana Sulieman Professor, Department of Mathematics & Statistics Thesis Co-adviser	
Dr. Stephen Chan Assistant Professor, Department of Mathematics & Statistics Thesis Committee Member	
Dr. Salam Ahmad Dhou Assistant Professor, Department of Computer Science & Engineering Thesis Committee Member	
Dr. Abdul Salam Jarrah Head and Program Coordinator Department of Mathematics & Statistics	
Dr. Hana Sulieman Associate Dean College of Arts and Sciences	
Dr. Mahmoud Anabtawi Dean College of Arts and Sciences	
Dr. Mohamed El-Tarhuni Vice Provost for Research and Graduate Studies Office of Research and Graduate Studies	

Aknowledgments

I would first like to thank my thesis advisors Dr. Ayman Alzaatreh and Dr. Hana Sulieman. Their guidance and advice was always available to me whenever I ran into a trouble spot or had a question about my research or writing. They have consistently allowed this to be my own work, but steered me in the right direction whenever they thought I needed it. I would also like to show gratitude to my committee members, Dr. Stephen Chan and Dr. Salam Dhou, as I am indebted to them for their valuable comments on this thesis. Furthermore, I would like to thank the rest of the faculty at the Department of Mathematics and Statistics for their unceasing encouragement, support and attention throughout my academic years. Last but not least, my appreciation also goes out to the Office of Graduate Studies and Research for their financial support in the form of GTA/GRA and their continuous consideration and aid whenever I needed it.

Finally, I must express my very profound gratitude to my parents and family for providing me with unfailing support and encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Abstract

With the rapid development of technology and the Internet, datasets have grown increasingly larger in size and dimensionality. As a result, feature selection has become a critical reprocessing tool in machine learning applications, as well as the subject of a plethora of research in a variety of fields. However, a common concern in feature selection is that different approaches can give very different results when applied to similar datasets. Aggregating the results of feature selection methods can help resolve this concern and control the diversity of selected feature subsets. In this work, we develop a general framework for the ensemble of different feature selection methods. Based on diversified datasets generated from the original set of observations, we aggregate within and between the importance scores generated by different feature selection techniques. The thesis goes into detail about the framework and its validation on prominent real-world datasets, using experimental analysis to show how aggregating multiple feature selection methods affects the learning algorithm's performance while identifying the optimal and most appropriate feature subset for a given dataset. In further contribution to this field, this thesis also examines the stability of the aggregation process that influences the stability of the feature selection algorithm. Correspondingly, different aggregation approaches are evaluated and compared using datasets from a variety of application fields, in terms of both the classification performance and the stability. The results are meant to emphasize the variations in aggregation approaches and highlight the role of the aggregation procedure in affecting feature selection robustness.

Keywords: Feature selection, aggregation, stability, ensemble learning, bootstrap sampling

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction

With the prevalence of the Internet and the recent technological advancements in database systems and information management, a vast amount of data in all kinds of application domains has become readily available at the touch of a button. Given the increasing volume of this data, traditional means of comprehending and interpreting information have proven inadequate. In turn, machine learning algorithms capable of extracting usable information from massive volumes of redundant data have grown increasingly popular. Among the numerous machine learning approaches, feature selection distinguishes itself by selecting a subset of the most important features while removing unnecessary, redundant, and noisy features. It allows learning algorithms to focus on the most relevant elements of the data, generally resulting in a faster and more accurate learning application. In this thesis, we are interested in enhancing feature selection approaches for machine learning with regards to both their stability and accuracy, through the development of an ensemble feature selection framework.

1.1 Background

The process of learning a set of rules from available data using automated procedures is widely known as machine learning. In inductive machine learning, the development of a learning algorithm allows available data to be analyzed for generalizations on new data. Among the various disciplines associated with machine learning, supervised learning is one of the most active ones [1]. In this field, a predictive model is trained using a collection of observations that include the desired outputs such that, once trained, the model can derive the likeliest output for samples that have yet to be observed. Depending on whether the output associated with the data is of the discrete or continuous type, supervised learning falls under two categories, classification or regression, respectively.

Like other machine learning algorithms, supervised learning largely benefits from bigger sample sizes (i.e. data observations) on which the learning algorithm is trained. However, due to the curse of dimensionality, larger numbers of features are not necessarily productive to the learning process. The curse of dimensionality refers to a slew of issues that occur when dealing with large amounts of data. Datasets containing a large number of features, such as gene microarray datasets, are more likely to have higher amounts of noise, which can lead to errors when learning algorithms are applied. In order to address such problems, feature selection has become a crucial preprocessing

step for the study of high-dimensional datasets [2]. Reducing the data dimensionality using feature selection can enhance performance by lowering model complexity and computational cost or by boosting its prediction accuracy and generalization of the results. The selection of appropriate features might also provide better readability and interpretability of the problem at hand. Given the nature of the relationship between the feature selection technique and the learning algorithm used for creating a prediction model, supervised feature selection approaches can be categorized into three groups: filters, wrappers, and embedded methods. Furthermore, new approaches for combining these existing methods and other machine learning techniques are constantly emerging to address additional challenges associated with feature selection, such as the stability of the feature selection process.

1.2 Problem Statement

As datasets expand in size, both in terms of observations and features, it becomes more difficult to examine their properties or obtain crucial insight into the relationship between features without employing feature selection. However, no single feature selection method outperforms all others across most applications, whether in terms of prediction accuracy or selection stability. In general, a main challenge within machine learning is choosing which feature selection technique to utilize. Accordingly, an understanding of the various types of feature selection algorithms is often needed before an appropriate method selection could be made. For instance, such a choice can be based on massive empirical evaluations of various feature selection methods, or through meta-learning, which involves a systemic study of the best feature selection algorithms in a given problem [3]. Alternatively, other areas of research have been interested in combining different learning models or feature selection methods via a process known as ensemble learning. The premise behind ensemble learning, in general, lies with the idea of combining several models in the hopes of achieving better results than of a singular model. In these ensembles, higher prediction accuracy scores are often not the sole consideration. Since ensemble feature selection frameworks integrate the findings of various feature selection methods, using an ensemble framework is believed to reduce the chances of selecting an unstable subset of features. Feature selection stability or robustness is characterized as the degree of fluctuation in the selected feature subsets given modest changes in the data used to obtain them [4]. As practitioners now recog-

nize the importance of having feature selection results that are resilient to fluctuations in the training data, the research of feature selection stability has received a lot of attention in recent years [5]–[7].

In order to achieve a better balance between predictive performance and stability, this work intends to develop a general framework for a bootstrap ensemble in which the feature selection results are aggregated homogeneously and heterogeneously across a collection of feature selection methods. Moreover, using the proposed ensemble framework, this work aims to analyze both the classification accuracy and stability performance of several aggregation approaches. Currently, while various studies have looked at ensemble feature selection methods based on different aggregations, little research has been done regarding the influence of the aggregation techniques themselves on the stability of the feature selection strategy. To this end, we wish to propose a framework for evaluating the stability of the aggregation techniques and further analyze the scope of this framework in order to understand the impact of the aggregation process.

1.3 Significance of the Research

While numerous feature selection methods exist across machine learning literature, each feature selection technique provides its own assumptions and mechanism for selecting the most important features. Accordingly, small perturbations in the dataset used to obtain the feature subsets might result in enormous changes in the resultant features, especially in high-dimensional datasets encompassing considerable amounts of noise and relatively low sample sizes. For that reason, we believe that investigating the issue of stability in ensemble feature selection, particularly when perturbing feature selection methods, is extremely important. To address this problem, we adopt an ensemble construction in which both data and method perturbation approaches are utilized. Based on diversified datasets generated from the original set of observations, we describe the aggregation of the importance scores generated by multiple feature selection techniques. The thesis intends to investigate in depth the proposed framework, analyze its properties, and assess its validation on prominent real-world datasets. While extensive published work concerns itself with analyzing and comparing ensemble feature selection methods using different aggregations, little work has been devoted to investigating the extent to which the aggregation techniques affects the stability of the ensemble feature selection. Accordingly, this thesis further examines the potential of combining numerous score and

rank-based aggregation rules by establishing an embedded structure for the aggregation procedure within the proposed ensemble bootstrap-based feature selection framework. The findings are intended to show the differences in aggregation techniques and the significance of the aggregating procedure in altering the feature selection stability, hence filling the important gap existing in the literature. The thesis also highlights possible differences in accuracy and stability between the score and rank-based aggregation procedures.

1.4 Thesis Objectives

The aims of this thesis are:

- To investigate and develop an accessible ensemble feature selection framework using bootstrap aggregation with the aim of increasing the robustness of the feature selection process and improving the accuracy of machine learning algorithms.
- To investigate the full potential of utilizing different aggregations, including score-based and rank-based aggregations, within the ensemble bootstrap aggregation framework; and to examine the influence of the aggregation process on the stability of the feature selection algorithm.
- To provide practical insights in terms of identifying an appropriate feature selection framework for a given problem, and recognize aggregation approaches that are more suited to particular application domains than others.

1.5 Thesis Outline

The remainder of this thesis is organized as follows:

Chapter 2 first introduces the background behind the feature selection process. We give an overview of feature selection methods across their three categroies: Filter, Wrapper, and Embedded methods. We also review previous work in the field of ensemble feature selection, with emphasis on the role of the aggregation techniques within the ensemble and the stability of the feature selection process.

In Chapter 3, we propose a boostrap aggregation framework for feature selection, which consists of a homogeneous Within Aggregation Method (WAM) and a heterogeneous Between Aggregation Method (BAM). The frameworks combines multiple feature selection techniques using bootstrap induced diversity. The framework for evaluating the stability influence of the aggregation procedure is also discussed.

Chapter 4 presents the experimental results of applying our proposed ensemble framework on a variety of real datasets from multiple application domains, and compares the accuracy and stability findings of the WAM and BAM algorithms. Furthermore, the effects of using different aggregation techniques on the accuracy and stability of the feature selection are also examined using the homogeneous WAM configuration.

Finally, the conclusion and prospects for future research are presented in Chapter 5.

Chapter 2. Literature Review

A feature is a unique measurable aspect of the observed process; machine learning algorithms may derive rules from available data using a collection of observed features. The process of selecting a subset of relevant features based on particular assessment criteria is known as feature selection. The goal of feature selection is to select as small as possible subset of features which can accurately represent certain data in order to deliver high prediction results while simultaneously limiting the impact of noise and irrelevant features. This chapter provides a thorough review of the existing feature selection methods and establishes the need for the ensemble approach in the selection of important and stable features.

2.1 Feature Selection Methods

In terms of the output, two types of feature selection approaches can be identified. That is, *univariate* methods that evaluate each feature independently, and *multivariate* methods that evaluate subsets of features simultaneously [8]. In multivariate methods, the feature selection procedure begins by choosing a subset of the original features and assessing each feature's value inside the subset. Using this evaluation, some features may be removed or added to the existing subset. Then, a specific assessment criterion is used to determine whether the final subset is satisfactory and can be thus returned to the user. With univariate methods, otherwise known as rankers, the feature selection method returns a ranking of all the features according to another assessment criteria, starting with the most important and ending with the least important features. In this scenario, a threshold must be established in order to select the final feature subset. It should be noted that feature selection is distinct from feature extraction (or feature transformation), which generates new features by combining existing ones. In feature selection, the original meaning of the selected features is preserved, which is desired in many areas.

In addition, depending on the relationship between a feature selection method and the corresponding learning algorithm, three categories of feature selection may be identified. These are filter, wrapper, and embedded methods. For instance, filter methods measure the relevance of each feature with regards to the output class labels independently of the learning algorithm; whereas both wrapper and embedded methods tend to employ the performance of the learning algorithm itself as a selection criteria. Wrapper

methods commonly search the space of all feature subsets, while in embedded methods, the search is guided by the learning algorithm. In order to assess feature subsets or individual features, several assessment criteria have been presented across the literature such as inconsistency rate [9], entropy [10], inference correlation [11], global sensitivity [12], and distance measures [13].

2.1.1 Filter methods

Filter methods largely depend on the general characteristics of the training data and select features as a pre-processing step independent of the learning algorithm. A typical filter algorithm consists of two steps: In the first step, features are ranked based on some criteria; this criteria differs from one technique to another. In the second step, features with the highest rankings are used to build classification and regression models. Filter methods are popular due to their computational efficiency and independence from the model. However, these properties can also be disadvantageous if they contribute to lower prediction accuracy, as the selected susbets may not be optimal for a certain model. Filter techniques can either be univariate or multivariate. Some of the most popular filter methods include: Information Gain [14], Symmetric Uncertainty [15], Chi-Squared [16], ReliefF [17], Minimum Redundancy Maximum Relevance (MRMR) [18], Correlation-Based Feature Selection (CFS) [19], and Consistency-based Filters [20]. More details about some of these techniques are given in [21].

2.1.2 Wrapper methods

Wrapper methods evaluate the feature subset by using the learning algorithm as a black box and the performance of the learning algorithm given the feature subset as the objective function [22]. These methods have the advantage of including the interaction between feature subset search and model selection, as well as the ability to account for feature dependencies. However, wrapper methods are generally more prone to overfitting than filter approaches and are more computationally costly, particularly if the implemented learning algorithm has high computational costs. Wrappers can be categorized into *sequential selection algorithms* and *heuristic search algorithms*. To optimize the objective function, heuristic search algorithms examine different feature subsets by either searching via a search space or producing optimization problem solutions, whereas sequential selection methods begin with an empty set (or full set) and gradually add

(or remove) features until the maximum objective function is achieved. Examples of sequential selection algorithms include Forward Selection, Backwards Selection, Exhaustive selection, and Recursive Selection. One of the most popular heuristic search approaches is the Genetic Algorithm (GA) [23].

2.1.3 Embedded methods

The way feature selection and the learning algorithm interact in embedded methods distinguishes them from other feature selection techniques. In embedded methods, feature selection is incorporated within the training procedure of the learning algorithm; i.e. the learning and feature selection parts cannot be separated. To do this, embedded methods employ an assessment criteria independent from the learning algorithm to determine the best subset for a specified cardinality. The learning algorithm is then used to choose the best feature subset from among the best subsets across all cardinalities. This method has the benefit of including interaction with the classification or regression model while being significantly less computationally costly than wrapper methods. Embedded approaches, in other words, capture feature dependencies and examine not just the link between the input features and the output variable, but also search locally for features that allow for improved local discrimination [2]. A popular example of this is SVM-RFE (Recursive Feature Elimination for Support Vector Machines) [24] and Random Forest Importance Scores [25]. Despite the effectiveness of embedded methods, it should be noted that they make classifier-dependent decisions that may or may not be compatible with other classifiers.

Given that each of the above discussed categories encompass a wide array of algorithms, there is a large corpus of feature selection methods. Most researchers believe that "the optimal technique" does not exist, and their efforts are thus directed at identifying a suitable approach for a given setting. In this regard, several approaches have been developed to cope with large-scale datasets, in the aim of decreasing training time and computational costs while retaining optimal prediction accuracy. Of the three categories listed above (filters, wrappers, and embedded methods), only filters are algorithm-independent. Due to this, filters are computationally simple and quick, and they can efficiently handle extremely large-scale datasets. However, most filters are univariate, which means that they examine each feature independently of other features and

may result in redundancy within the selected feature subsets. To account for these issues, some hybrid filter-wrapper approaches have been proposed in several works [26], [27]. For instance, the dimension of the feature space can be first reduced using a filter technique, after which a wrapper approach can then be used to choose the best feature subset. Alternatively, a more generalized ensemble feature selection framework can be adopted, wherein lies the focus of this work.

2.2 Ensemble Feature Selection

Ensemble methods build on the assumption that the aggregation of multiple models may provide better results than the use of a single model. Given the popularity of ensemble techniques for classification models, the concept of aggregating feature selection methods was recently proposed in [28]. Several works have since embraced this approach, including [29]–[32]. Some entail the use of classifiers, while others do not. Ensemble feature selection frameworks generally consist of two main steps: First, a diversification approach is used to produce different feature selection outputs. Then, an aggregation strategy is employed to combine the produced outputs. Data perturbation and/or method perturbation can be used to achieve diversification. Method perturbation, for instance, introduces diversity by aggregating the results of different feature selection methods. The aim is to construct a subset of relevant features capable of conveying the benefits and drawbacks of all utilized feature selection methods while avoiding the biases of each individual method. This may result in more robust and/or better performing feature subsets [28], but it may also incur a higher processing cost and increase the difficulty of interpreting those results.

For classification models, the most well-known approaches for applying ensemble learning are Bagging and Boosting. These approaches introduce diversity through data perturbation; that is, different results are obtained by introducing variations to the training data. In Bagging, each individual classifier in the ensemble is obtained from a different training set. These varying training sets are constructed through randomly sampling the original data with replacement. In Boosting, a "weak" learning algorithm, is boosted into an arbitrarily "strong" one [33]. By way of example, a boosting algorithm may sequentially obtain a series of classifiers by iteratively updating the training set to compensate for errors made in predictions by earlier classifiers. In other words, boosting would perform random sampling with replacement on weighted data rather

than the original dataset. Ensemble approaches can also be used to minimize several objective functions simultaneously. In their work, Ng *et al*. [34] proposed a multi-objective genetic algorithm NSGA-III with the intention of reducing both training error and ensemble sensitivity. It follows that the ideas used in Bagging and Boosting can be generalized to ensemble feature selection by utilizing feature selection methods instead of classifiers.

In this manner, ensemble feature selection methods can also be categorized into homogeneous and heterogeneous methods [35]. Homogeneous ensembles generally combine the results of the same feature selection methods under different training sets (i.e. data perturbation), whereas heterogeneous ensembles combine the results of different feature selection methods under the same training set (i.e. method perturbation). Overall, both homogeneous and heterogeneous approaches have produced promising results across the literature [28], [36]–[40]. In [41], an analysis and comparison of parallel and serial ensemble combination techniques for homogeneous ensembles concluded that ensemble feature selection outperforms single feature selection in terms of classification accuracy. If the dataset is large or comes with long computational running times, then homogeneous ensembles tend to be recommended due to their lower processing costs [42]. On the other hand, heteregoenus ensembles can be the better option if the dataset is considered small or if the researcher is not sure which choice of feature selection technique would be optimal [42]. In such case, the degree of diversity/similarity between the included feature selection methods needs to be carefully considered for ensuring the best results [43]. In their work, Seijo-Pardo *et al*. [36] built homogeneus and heteregoneus ensembles of five feature selection methods using multiple aggregation methods. The experimental results over seven datasets demonstrate that ensemble feature selection outperforms single methods, and that the difference in classification error between the two ensemble types (homogeneus and heteregoneus) was limited by the dataset size and dimension. In practice, heterogeneous ensembles tend to be the more popular ensemble configuration [29], [30], [44].

Additional considerations for building an ensemble feature selection technique include: the nature of the resampling method, the number/size of the training sets, the aggregation procedure for combining the results of the feature selection methods, the thresholds for selecting the feature subsets, and the learning algorithms on which the se-

lected features are applied [45]. To this end, recent novel techniques in ensemble feature selection include integrating classifiers in the feature selection ensemble [46], introducing a number of automatic feature threshold identifiers [47], [48], assessing the relative importance of each features using several iterations of a genetic algorithm [49], [50], and combining multiple feature-ranking techniques via clustering to select an optimal feature subset [31].

Moreover, in addition to the evaluation of a method's classification performance, the study of the stability of the ensemble feature selection has gained more attention in recent years. Overall, the stability or robustness of any feature selection technique serves as an indicator of its reproducibility power. Accordingly, feature selection stability is particularly valuable for ensuring confidence in the selected feature subsets, especially if additional studies or validations of the selected subsets are costly [28].

On that account, while the use of ensemble techniques generally necessitates higher computational resources, it is possible to create a fast and effective ensemble feature selection framework for dealing with small sample domains at an affordable cost. For instance, previous work suggests that only the least stable feature selection methods truly benefit from a computationally expensive ensemble framework [39], [51], [52]. Furthermore, it was shown that a small number of feature selection methods within an ensemble framework can provide similar or better results than that of a larger ensemble [29].

2.2.1 Aggregation techniques

In order to maximize the effectiveness of ensemble learning, ensemble feature selection frameworks have been tested under multiple considerations. Among them, the aggregation procedure has a particularly substantial impact on the outcome of the ensemble feature selection. To this end, numerous aggregation strategies have been adopted across the literature. In contrast to more complex and computationally expensive techniques, it is possible to combine the homogeneous or heterogeneous feature selection subsets using simple set intersection and/or union [53]. Alternatively, it is possible to combine the label predictions instead, by applying a classification technique based on the different feature selection methods and then combining the obtained labels from each feature selection output [54]. Examples of this include the use of majority votes [55]

and cumulative probability [54]. Many works also adopt the use of ranking filter methods, from which the obtained rankings are then aggregated into a single final ranked list [56]–[58]. In this manner, the most popular rank aggregation techniques include the Borda Methods [59], Stuart [60], Robust Rank Aggregation [61], and SVM-Rank [62]. Other aggregation techniques can also be used to account for interactions between the features or to identify more than one appropriate feature subset [63]. In their paper, Wald *et al*. [64] applied nine rank aggregation techniques on twenty-five feature ranking methods. Their findings highlight the effectiveness of rank aggregation, but indicate no significant differences between the different rank aggregation methods. A similar conclusion was noted across the rank aggregations used in [36]. Meanwhile, another study [65] demonstrates that the similarity between the aggregation methods effectively increases as the selected feature subsets grow larger. In fact, a comparison of the results presents definite clusters of similarity between the different aggregation methods. As such, each cluster can be better-suited for a certain problem domain than others. Alternatively, within the same cluster, use of a simpler aggregation technique, such as the mean aggregation, can be recommended over a complex one [51], [66].

While less commonly used across the literature, score-based aggregation is also another approach for combining the output of multiple feature selection methods. In fact, it was shown that applying the same Arithmetic Mean aggregation technique on both the feature importance scores and the feature ranks can lead to vastly different results [67]. However, there is no clear rule on which of the two is the better option. In text categorization, average rank aggregation outperformed average score aggregation when applied on the same datasets [68]. Yet, in terms of stability, Dernoncourt *et al*. [69] noted that average score aggregation generally resulted in better stability than average rank aggregation.

2.3 Feature Selection Stability

The insensitivity of a feature selection algorithm to changes in the training data and how it impacts the feature selection process is measured by the feature selection stability. In other words, a feature selection method is termed stable if it returns comparable feature rankings over several training sets obtained from the same dataset. Since small changes in the training sample should not have a considerable impact on the obtained feature rankings, stable algorithms are valued for the consistency of their outputs. Correspond-

ingly, it is critical to augment the examination of classification performance with stability analysis in order to guarantee the quality of the feature selection process. In most cases, the best trade-off between the stability and classification performance depends on the dataset itself [28]. However, previous work suggests that homogeneous ensembles can be more stable than their heterogeneous counterparts [54].

It should be noted here that the stability of feature selection techniques is influenced by several factors such as the dataset variation [70], dataset imbalance [71], or feature redundancy [72]. Accordingly, several metrics used to assess the stability of different feature subsets have been introduced and discussed across the literature [5], [73], [74]. For various datasets, these metrics can be utilized for finding a more robust feature subset. In [75], Yang and Mao proposed a multicriterion fusion-based recursive feature elimination (MCF-RFE) algorithm with the goal of improving both classification performance and stability of feature selection results. Moreover, intensive search techniques such as the genetric algorithm have been introduced inside ensembles for improving the feature selection stability [76]. Splitting the input features (depending on their feature extraction techniques) across various classifiers and combining the predictions to arrive at a final conclusion is also recommended in [77]. However, while numerous studies have looked at the stability of ensemble feature selection techniques under countless settings, little work has explored the impact of the aggregation process itself on the stability of the ensemble technique. In [64], using the same mean aggregation approach on both feature importance scores and feature rankings provides radically different outcomes depending on the data characteristics and application domain. In some applications, average score aggregation was found to be more stable than average rank aggregation, whereas average rank aggregation dominated in terms of the classification performance in other applications [68]. Accordingly, it is fairly possible that the stability of the aggregation process directly influences the stability of the feature selection within the ensemble.

Chapter 3. Methodology

The purpose of this chapter is to propose two frameworks for the feature selection ensemble that provide better accuracy of the learning algorithm and higher stability of the feature selection process. The Within Aggregation Method (WAM) is used to aggregate the importance scores within a single feature selection method. The Between Aggregation Method (BAM) is used to aggregate the importance scores between different feature selection methods. Both methods will be assessed and compared using a variety of feature selection techniques and some stability measures. Moreover, to explore the impact of the aggregation procedure on the feature selection ensemble, this chapter outlines the methodology for analyzing the stability behavior of the proposed framework using different aggregation functions. We thoroughly detail each of the approaches used for selecting the most important features, for aggregating the importance scores and ranks, and the classification algorithms used in the experimental analysis.

3.1 Definitions and Notations

Note that throughout this chapter, the terms feature and variable are used interchangeably. Given a dataset $\mathbb{S} \equiv (X, Y)$, with n observations and p features (variables) such that $n, p \in \mathbb{Z}_{>0}$. That is, $X = [x_{ij}]_{n\times p} \in \mathbb{R}^{n,p}$ is the matrix of observations and Y is the target variable (i.e. the rows are the observations and the columns are the features of the dataset). Let x_{ij} denote the observation i of the feature j. In order to predict the target variable Y, the proposed algorithm aims to reduce the number of features in the dataset X.

Now, let $\{V_1, \ldots, V_p\}$ denote the set of features in X and $\{FS_1, \ldots, FS_t\}$ denote the feature selection methods used, where $t \in \mathbb{Z}_{>0}$. For every feature $V_j \in \{V_1, \ldots, V_p\}$, we assume each feature selection method $FS_q \in \{FS_1, \ldots, FS_t\}$ generates a feature importance score $\ell_j \in \mathbb{R}$. Despite being less commonly used in the literature, feature importance scores possess a higher level of detail than the ranks and might be better able to differentiate between the features, particularly in the case of ties. For this reason, this work highlights the use of score-based aggregations in addition to the rank-based aggregations. Accordingly, a normalization technique is implemented for meaningful comparison of the scores derived from different feature selection algorithms.

To produce the desired importance scores; first, the dataset $\mathbb{S}$ is divided into a training dataset $\mathbb{X}$, and a testing dataset $\mathbb{T}$. Here, $\mathbb{X} = X_{\lceil rn \rceil, p}$ and $\mathbb{T} = X_{n-\lceil rn \rceil, p}$ with $0 < r < 1$.

Note that $\lceil rn \rceil$ refers to the Ceiling function (upper bound) of $rn \in \mathbb{R}$. For instance, given the dataset Jasmine with $n = 2984$ observations and $p = 145$ features, then $\{V_1, \ldots, V_{145}\}$ denotes the set of features in Jasmine (Table 4.1). If two-thirds ($r = \frac{2}{3}$) of Jasmine are taken for training, whereas one-third is used for testing. Then, the training data is denoted $\mathbb{X} = X_{1990,145}$ and the testing data is denoted $\mathbb{T} = X_{994,145}$. The aim of this work is to construct an ensemble of feature selection methods in order to reduce the 145 features in Jasmine to only the most important features.

3.2 Bootstrap Aggregation Framework

In this section, we illustrate the proposed bootstrap aggregation procedure for aggregating the feature importance scores inside each FS_q and between different $FS_1, \ldots, FS_t$. The WAM is an ensemble method that aggregates importance scores within a single feature selection method (section 3.2.1), whereas the BAM aggregates importance scores between several feature selection methods (section 3.2.2). In the aggregation step, the proposed WAM and BAM methodologies allow for the implementation of any score or rank-based aggregation strategy such as the L2 norm, Geometric Mean, Robust Rank Aggregation, etc. Thus, we will refer to the aggreagtion technique used in the ensemble framework as AT.

3.2.1 Within Aggregation Method (WAM)

Given a training dataset $\mathbb{X}$, aggregation technique AT, and a feature selection method FS. Let $\mathbb{X}_1, \mathbb{X}_2, \ldots, \mathbb{X}_m$ be bootstrapped samples from $\mathbb{X}$, where $m \in \mathbb{Z}_{>0}$. Then, we apply FS on each $\mathbb{X}_s$, $s = 1, \ldots, m$ to generate feature importance scores $\{\ell_{s1}, \ldots, \ell_{sp}\}$ which corresponds to the set of features $\{V_1, \ldots, V_p\}$. Therefore, a score matrix $\mathbb{L} = [\ell_{sj}] \in \mathbb{R}^{m \times p}$ is generated after applying the feature selection method FS on each bootstrap sample. In $\mathbb{L}$, column j represents the FS importance scores for the feature V_j over the m bootstrap sample datasets, whereas row s represents the FS importance scores for all features $\{V_1, \ldots, V_p\}$ in one bootstrap sample $\mathbb{X}_s$. The final aggregated score for the feature V_j is defined to be the aggregation (via the aggreagtion technique AT) of column j in $\mathbb{L}$. For instance, assuming $AT =$ Arithmetic Mean, we use the notation $a_j = \dfrac{\sum_{s=1}^{m} \ell_{sj}}{m}$ to denote the aggregated scores of V_j. Then, a rank vector $r = (r_1, \ldots, r_p)$, $r_j \in \{1, 2, \ldots, p\}$ is assigned to the feature set $\{V_1, \ldots, V_p\}$ based on the aggregated scores $\{a_1, \ldots, a_p\}$. Based on the rank vector r, the feature set is then sorted

from the most to the least important. Now, based on a threshold parameter, $0 < k \leq 1$, we keep only the most important $100k\%$ of the feature set (determined by the rank vector r). The WAM approach can be used to compare the performance of different feature selection techniques based on various supervised learning methods for a given dataset. The flowchart in Figure 3.1 illustrates the WAM framework. It is clear that both classification and regression problems are comptaible with this framework. Algorithm 1 further details the WAM below.

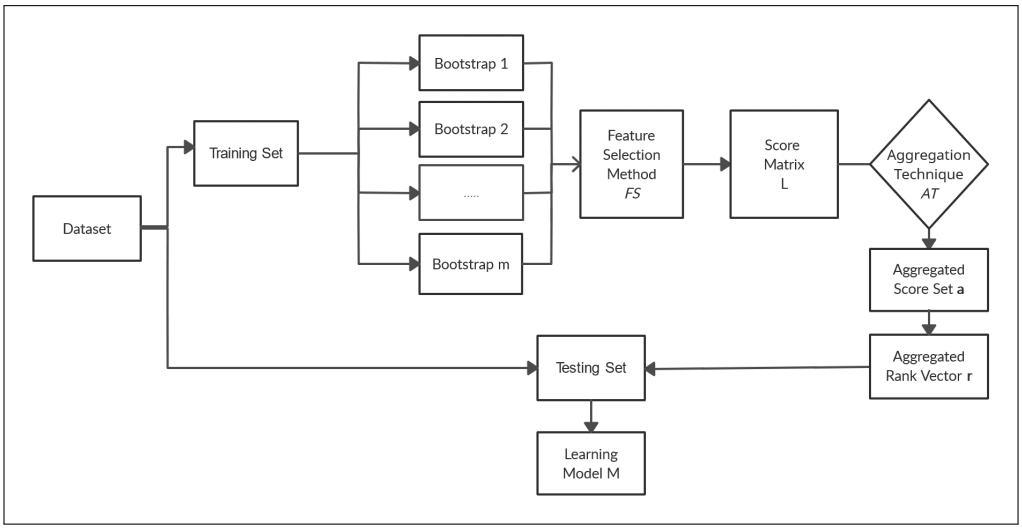

Figure 3.1. Framework for WAM

Algorithm 1 WAM Algorithm

Given a training dataset $\mathbb{X}$ with p features, a testing dataset $\mathbb{T}$, a feature selection method FS, an aggregation technique AT, a threshold parameter k, and a learning algorithm M.

1: For $s = 1, \ldots, m$, generate bootstrap samples, $\mathbb{X}_1, \ldots, \mathbb{X}_m$ of the training dataset $\mathbb{X}$.
2: Based on FS, obtain a features score matrix $\mathbb{L}$.
3: Based on AT, obtain aggregated score set $\{a_1, \ldots, a_p\}$.
4: For the aggregated score set $\{a_1, \ldots, a_p\}$, get the corresponding rank vector $r = (r_1, \ldots, r_p)$.
5: Based on the rank vector r, keep only the top $100k\%$ of the variable set $\{V_1, \ldots, V_p\}$.
6: Based on the selected feature set in (5), use the testing dataset $\mathbb{T}$ and a cross-validation technique to train and test the model M.

3.2.2 Between Aggregation Method (BAM)

Given a training dataset $\mathbb{X}$, feature selection methods $\{FS_1,\ldots,FS_t\}$, and aggregation technique AT. Let $\mathbb{X}_1,\mathbb{X}_2,\ldots,\mathbb{X}_m$ be bootstrapped samples from $\mathbb{X}$, where $m\in\mathbb{Z}_{>0}$. Then, we apply FS_q, $q=1,\ldots,t$ on each $\mathbb{X}_s$, $s=1,\ldots,m$, to generate feature importance scores $\{\ell_{s1},\ldots,\ell_{sp}\}^{(q)}$ which corresponds to the set of features $\{V_1,\ldots,V_p\}$. Therefore, a score matrix $\mathbb{L}^{(q)}=[\ell_{sj}^{(q)}]\in\mathbb{R}^{m\times p}$ is generated after applying the feature selection method FS_q on each bootstrap sample. In $\mathbb{L}^{(q)}$, column j represents the FS_q scores for variable V_j over the m bootstrap sample datasets, whereas row s represents the FS_q importance scores for all features $\{V_1,\ldots,V_p\}$ in one bootstrap sample $\mathbb{X}_s$. To allow for the aggregation of different feature selection outputs, each column in the score matrix $\mathbb{L}^{(q)}$ is normalized using min-max normalization as follows:

$$\vec{\mathbb{L}}^{(q)}=[\vec{\ell}_{sj}^{(q)}],\text{ where }\vec{\ell}_{sj}^{(q)}=\frac{\ell_{sj}^{(q)}-\min\limits_{s}\ell_{sj}^{(q)}}{\max\limits_{s}\ell_{sj}^{(q)}-\min\limits_{s}\ell_{sj}^{(q)}}$$

Then, the aggregation technique AT is used to combine the normalized importance scores across FS_q, $q=1,\ldots t$ into one score matrix $\vec{\bar{L}}=\dfrac{\sum\limits_{q=1}^{t}\vec{\mathbb{L}}^{(q)}}{t}$. For instance, assuming $AT=$ Arithmetic Mean, column j in the score matrix $\vec{\bar{L}}$ represents the average column j between all considered feature selection methods $FS_1,\ldots,FS_t$. Then, the $AT=$ Arithmetic Mean of columns $1,\ldots,p$ in $\vec{\bar{L}}$, is once more used to obtain the final aggregated scores $\{a_1,\ldots,a_p\}$ for the feature set $\{V_1,\ldots,V_p\}$. The rank vector $r=(r_1,\ldots,r_p)$, $r_j\in\{1,2,\ldots,p\}$ is assigned to the feature set $\{V_1,\ldots,V_p\}$ based on the final aggregated scores. The feature set is then sorted from the most to the least important based on the rank vector r. Now, using a threshold parameter, $0<k\leq 1$, we keep only the most important $100k\%$ of the feature set (determined by the rank vector r). The flowchart in Figure 3.2 illustrates the BAM framework. It is clear that both classification and regression problems are comptaible with this framework. Algorithm 2 further details the BAM below.

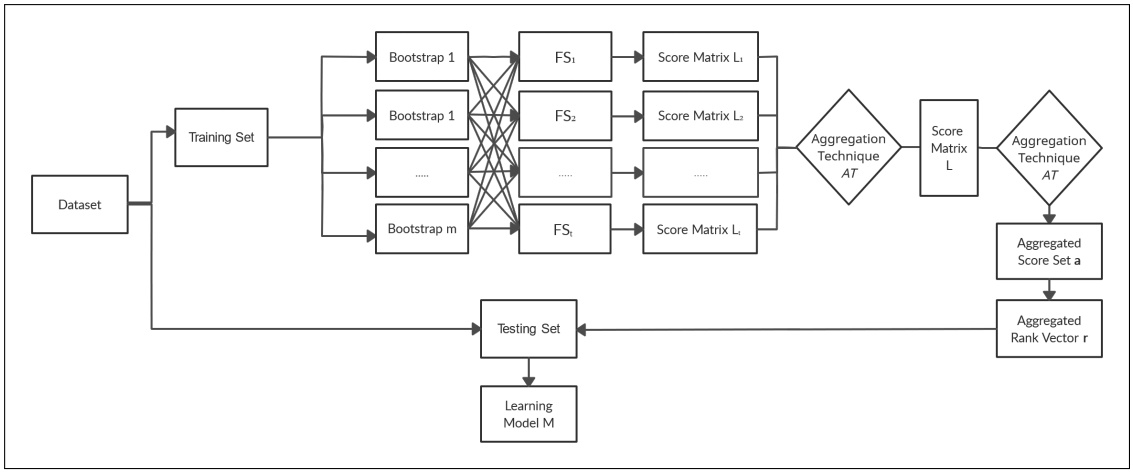

Figure 3.2. Framework for BAM

Algorithm 2 BAM Algorithm

Given a training dataset $\mathbb{X}$ with p features, a testing dataset $\mathbb{T}$, feature selection methods $\{FS_1,\ldots,FS_t\}$, an aggregation technique AT, a threshold parameter k, and a learning algorithm M.

1: For $i=1,\ldots,m$, generate bootstrap samples, $\mathbb{X}_1,\ldots,\mathbb{X}_m$ of the training dataset $\mathbb{X}$.
2: For each feature selection method $FS_q \in \{FS_1,\ldots,FS_t\}$, get features score matrix $\mathbb{L}^{(q)}$.
3: Normalize the score matrices in (2) as $\vec{\mathbb{L}}^{(q)}$, $q=1,\ldots,t$.
4: Use aggregation technique AT to combine the matrices in (3) into one score matrix $\vec{\bar{L}}$.
5: Based on AT, obtain aggregated score set $\{a_1,\ldots,a_p\}$.
6: For the aggregated scores in (5), compute the corresponding rank vector $r=(r_1,\ldots,r_p)$.
7: Based on the rank vector r, keep the top $100k\%$ of the variable set $\{V_1,\ldots,V_p\}$.
8: Based on the selected feature set in (6), use the testing dataset $\mathbb{T}$ and a cross-validation technique to train and test the model M.

3.3 Stability Analysis

As a measure of the sensitivity of the feature selection process, stability assesses the influence of changes in the training data on the output of the feature selection procedure. Since the stability of feature selection is mainly influenced by data variation, we can induce variation in the dataset on which we wish to evaluate the stability through two approaches [4]:

Approach 1 Divide the dataset $\mathbb{S}$ into a training dataset $\mathbb{X}$ and a testing dataset $\mathbb{T}$. Then, multiple training samples $\mathbb{X}_1, \mathbb{X}_2, \ldots, \mathbb{X}_m$ are generated by bootstrapping $\mathbb{X}$ with replacement.

Approach 2 Use m-fold cross-validation to generate different training datasets. For example, take a 5-fold cross-validation procedure. On every iteration, one of these folds is used as a testing dataset $\mathbb{T}$, while the remaining four folds are used as a training dataset $\mathbb{X}$. In this manner, by going through all iterations, one can obtain five training samples $\mathbb{X}_1, \mathbb{X}_2, \ldots, \mathbb{X}_5$ and five testing samples $\mathbb{T}_1, \mathbb{T}_2, \ldots, \mathbb{T}_5$.

In section 3.3.1, using Approach 1, we describe the stability evaluation of the feature selection methods by comparing the similarity of feature selection outputs produced from different bootstrapped samples. Furthermore, by introducing additional data variations to the aggregations using Approach 2, we underline the stability influence of the aggregation techniques themselves within the feature selection ensemble in section 3.3.2.

3.3.1 Feature selection stability

In order to measure the stability of feature rankings for a certain feature selection method, a similarity-based approach can be implemented. This approach depends on the representation language of the produced feature rankings. Given a feature selection method FS, let $\mathbb{X}_1, \mathbb{X}_2, \ldots, \mathbb{X}_m$ be the training samples obtained using Approach 1. Through bootstrapping with replacement, this approach allows us to construct simulated samples without making assumptions about the underlying data distribution. Then, by applying FS on each $\mathbb{X}_s$, $s = 1, \ldots, m$; we obtain any of the following three representations with respect to the feature set $\{V_1, \ldots, V_p\}$ and the sample dataset $\mathbb{X}_s$:

- An importance scores vector $\boldsymbol{\ell}_s = (\ell_{s1}, \ldots, \ell_{sp}), \ell_{sj} \in \mathbb{R}$
- A rank vector $r_s = (r_{s1}, \ldots, r_{sp})$, $r_{sj} \in \{1, 2, \ldots, p\}$
- A subset of features represented by an index vector $w_s = (w_{s1}, \ldots, w_{sp})$, $w_{sj} \in \{1, 0\}$, where 1 indicates feature presence and 0 indicates feature absence.

Naturally, it is possible to transform any feature importance scores vector $\boldsymbol{\ell}$ into a rank vector r by sorting the importance scores. On the other hand, a rank vector r may be converted into an index vector w by selecting the top $100k\%$ features. The most popular approach for assessing the stability of a feature selection method is to simply average the similarity comparisons between each pair of feature rankings produced from different bootstrap samples, as shown below:

$$Stability = \frac{2}{m(m-1)} \sum_{s=1}^{m-1} \sum_{v=s+1}^{m} \Phi(f_s, f_v) \tag{3.1}$$

where $\Phi(f_s, f_v)$ is the similarity measure between a pair of feature rankings from any two training samples $\mathbb{X}_s, \mathbb{X}_v$ $(1 \leqslant s, v \leqslant m)$. Note that the feature rankings (f_s, f_v) can be represented as a pair of importance scores vectors, rank vectors, or index vectors. Moreover, the multiple $\frac{2}{m(m-1)}$ stems from the fact that there are $\frac{m(m-1)}{2}$ possible pairs of feature rankings between the total m samples. Several stability measures have been introduced in the literature, with Jaccard's index being one of the most commonly used [78]. To calculate the stability, feature selection methods can be used to produce importance scores vectors $\{\boldsymbol{\ell}_1, \ldots \boldsymbol{\ell}_m\}$ to be converted into rank vectors $\{r_1, \ldots r_m\}$ or index vectors $\{w_1, \ldots w_m\}$ and used to evaluate the stability using a corresponding similarity measure.

3.3.1.1 Score-based stability

In the case of the similarity between two importance score vectors $(\boldsymbol{\ell}_s, \boldsymbol{\ell}_v)$ produced by one of the feature selection methods, the *Pearson's Correlation Coefficient* computes the similarity measure as:

$$\Phi_{PCC}(\boldsymbol{\ell}_s, \boldsymbol{\ell}_v) = \frac{\sum_{j=1}^{p} (\ell_{sj} - \bar{\ell}_{s*})(\ell_{vj} - \bar{\ell}_{v*})}{\sqrt{\sum_{j=1}^{p} (\ell_{sj} - \bar{\ell}_{s*})^2 (\ell_{vj} - \bar{\ell}_{v*})^2}}, \tag{3.2}$$

where the $\boldsymbol{\ell}_s$ refers to the feature importance scores corresponding to the set of features $\{V_1,\ldots,V_p\}$ obtained from $\mathbb{X}_s$. Given a score matrix $\mathbb{L}$, in which the rows correpsond to the importance scores across the training samples $\mathbb{X}_1,\mathbb{X}_2,\ldots,\mathbb{X}_m$ and the columns correspond to the features $\{V_1,\ldots,V_p\}$, then $\boldsymbol{\ell}_s$ refers to the row s in $\mathbb{L}$. Furthermore, $\bar{\ell}_{s*}$ refers to the mean of row s in $\mathbb{L}$. Note that $\Phi_{PCC}(\boldsymbol{\ell}_s,\boldsymbol{\ell}_v)\in[-1,1]$.

In most of feature selection literature, index-based similarity measures tend to be more common due to the popularity of multivariate feature selection methods. However, a recent study, which aims to identify desirable qualities in a stability measure, points out that Pearson's Correlation Coefficient fulfills all desirable properties, yet tends to be overlooked in favour of more complex alternatives [5].

3.3.1.2 Rank-based stability

In the case of the similarity between two rank vectors (r_s,r_v) produced by one of the feature selection methods, there are two possible similarity measures for this evaluation:

i. *Spearman Rank Correlation Coefficient* measures the similarity between the two rank vectors as:

$$\Phi_{SRCC}(r_s,r_v)=1-\frac{6\sum_{j=1}^{p}(r_{sj}-r_{vj})^2}{p(p^2-1)}, \tag{3.3}$$

where r_s is the rank vector that corresponds to the set of features $\{V_1,\ldots,V_p\}$. Here, $\Phi_{SRCC}(r_s,r_v)\in[-1,1]$.

ii. *Canberra Distance* quantifies the similarity between two rank vectors (r_s,r_v) as the absolute difference between them [79]:

$$\Phi_{CD}(r_s,r_v)=\sum_{j=1}^{p}\frac{|r_{sj}-r_{vj}|}{|r_{sj}|+|r_{vj}|}. \tag{3.4}$$

where r_s is the rank vector that corresponds to the set of features $\{V_1,\ldots,V_p\}$. For easier interpretation, Canberra's distance is usually normalized through dividing by p.

3.3.1.3 Index-based stability

Due to the multivariate nature of many feature selection techniques, numerous stability metrics seek to assess the degree of similarity between two feature subsets represented by two index vectors. The most popular similarity measure for this evaluation

is *Jaccard's Index*. Jaccard's index measures the similarity between two finite sets; it is taken as the size of the intersection divided by the size of the union of the two sets. Given the index vectors (w_s, w_v) used to represent the two feature subsets, Jaccard's index is given by:

$$\Phi_{JI}(w_s, w_v) = \frac{|w_s \cap w_v|}{|w_s \cup w_v|} = \frac{|w_s \cap w_v|}{|w_s| + |w_v| - |w_s \cap w_v|}, \tag{3.5}$$

such that $\Phi_{JI}(w_s, w_v) \in [0, 1]$.

3.3.1.4 Average Standard Deviation (ASD)

In addition to similarity-based stability measures, the *Average Standard Deviation* of feature importance scores across all bootstrap samples may be computed for each feature selection technique. The standard deviation, by definition, reflects the dispersion or instability of the feature selection scores over training bootstraps. Similar to the work in [80], we define the ASD for a given feature selection approach as follows:

$$ASD = \frac{1}{p} \sum_{j=1}^{p} SD(c_j), \tag{3.6}$$

where c_j represents the column j in the standardized score matrix $\vec{\mathbb{L}}$. Across the m bootstrap samples, $SD(c_j)$ is the standard deviation of the standardized importance scores for the feature V_j. That is, a low ASD would indicate strong stability, whereas a large ASD would indicate poorer stability.

For the WAM and BAM methodologies, the four discussed similarity measures (Pearson's Correlation Coefficient, Spearman's Rank Correlation Coefficient, Canberra's Distance, and Jaccard's Index) are evaluated to assess the stability of the tested feature selection techniques for each of the possible representation languages. In addition, for every feature selection method, we compute the Average Standard Deviation (ASD) of the feature importance scores across all bootstrap samples.

3.3.2 Influence of the aggregation procedure

In the implemented framework, we combine the two approaches, Approach 1 and Approach 2, to further analyze the stability influence of the aggregation procedure on the ensemble feature selection framework. Given a fixed feature selection method FS; we

integrate the WAM framework, built using Approach 1, within a cross-validation process of Approach 2, as illustrated in Figure 3.3. In other words, an internal WAM framework is used to produce an aggregated feature selection result for each iteration of the cross-validation. Specifically, stratified cross-validation is applied on the dataset $\mathbb{S}$ to produce f-folds. 1 fold is set aside for each iteration (dataset $\mathbb{T}$), while the remaining $f-1$ folds are used for training (dataset $\mathbb{X}$). The training data is subsequently subjected to Algorithm 1 (WAM). Inside each iteration of the cross-validation, the bootstrap samples $\mathbb{X}_1, \ldots, \mathbb{X}_m$ will be created and utilized to obtain an aggregated rank vector r. By going through all iterations, we acquire $\{r_1, \ldots r_f\}$ aggregated rank vectors

Given the aggregated rank vectors $\{r_1, \ldots r_f\}$, the stability of the ensemble feature selection can be evaluated by averaging over the values of any of the similarity measures described in section 3.3.1. In this manner, the similarity scores of all pairs of aggregated feature rankings are computed and then averaged to find a final stability score for the chosen similarity measure. As the WAM is embedded within the cross-validation operation, each output of the WAM creates an aggregated rank vector for each aggregation approach. In accordance, the cross-validation folds (Approach 2) incorporate data variance into these aggregated rank vectors. The ensuing instability is then mostly attributable to the aggregation procedure, because a single feature selection method was fixed across all folds and in each ensemble. Hence, we underline the stability influence of the aggregation techniques themselves within the ensemble feature selection.

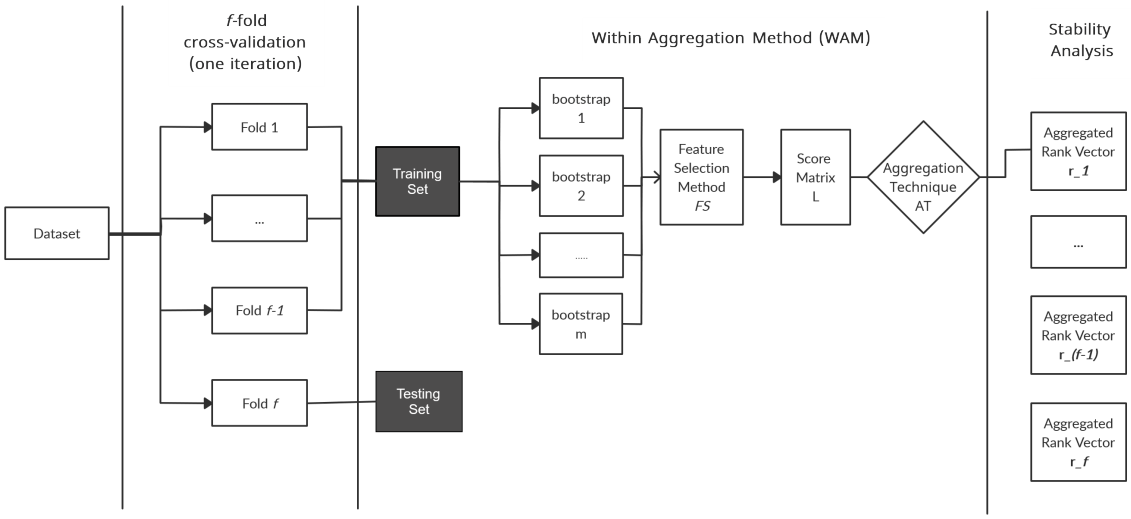

Figure 3.3. Framework for analyzing the stability influence of the aggregation procedure

3.4 Feature Selection Techniques

As described in section 2.1, countless feature selection methods exist in the literature including filter, wrapper, and embedded methods. Due to their computational efficiency and reliable performance, filter methods are used within this thesis to test the proposed methodology. Four of the most popular filter methods used in the experimental design are discussed in this section.

3.4.1 Information Gain (IG)

Information Gain (IG) is one of the most common feature selection methods due to its computational efficiency and ease of understanding. Based on entropy, IG is a symmetrical measure of dependency between two random variables X and Y that measures the information obtained about Y after seeing X, or vice versa. In feature selection, X usually refers to one of the dataset's features, whereas Y refers to the target variable. Moreover, entropy is a measure of the uncertainty of a random variable or the amount of information required to predict its outcome. For simplicity, assume that both X and Y are nominal features, with n and m unique classes for X and Y, respectively. The IG importance score of X is determined by the decrease in the entropy of Y when X is known. Here, the entropy of Y is given by [81]:

$$H(Y) = -\sum_{y} P(y)\log_2(P(y)) \tag{3.7}$$

where $P(y)$ is the probability that an arbitrary sample belongs to class $y \in Y$. On the other hand, the conditional entropy of Y given X, namely $H(Y \mid X)$, represents the uncertainty about Y given the value of X. This is given by:

$$H(Y \mid X = x) = -\sum_{y} \frac{P(x,y)}{P(x)} \log_2\left(\frac{P(x,y)}{P(x)}\right) \tag{3.8}$$

where $P(x,y)$ is the probability that an observation is of class y in a subset x. For nominal random variables, we average $H(Y \mid X = x)$ over all possible values that X may take in order to obtain $H(Y \mid X)$. Accordingly, the Information Gain on knowing X is given by:

$$IG(Y;X) = H(Y) - H(Y \mid X) \tag{3.9}$$

Alternatively, the joint entropy or the sum of the uncertainty contained by the two features can be calculated using:

$$H(X,Y) = -\sum_{x}\sum_{y} P(x,y) \cdot \log_2 \left(P(x,y)\right) \tag{3.10}$$

where $P(x,y)$ is the probability that an observation is of class y in a subset x. In this case, the Information Gain on knowing X can be simply be obtained by:

$$IG(Y;X) = H(X) + H(Y) - H(Y,X) \tag{3.11}$$

For continuous variables, differential entropy replaces the summations in (3.7), 3.8, and (3.10) by integration. IG may be used as a correlation metric since it aims to assess how much information a feature provides about the target variable. The greater the IG, the stronger the relationship between the features X and Y. They are independent if the IG between X and Y is zero. They are dependant if it is more than zero and one variable can supply information about the other. Information Gain can be used with all data types. However, one disadvantages of this method with nominal variables is that it favors features with more distinct values even when they are not more informative (e.g. customer's identification number).

3.4.2 Symmetric Uncertainty (SU)

Symmetric Uncertainty (SU) is a correlation measure between two random variables X and Y, usually an independent feature of the dataset and the target variable. SU, like Information Gain, uses entropy to determine how much information the features contribute. However, the SU criterion adjusts for the inherent bias of IG by dividing it by the sum of X and Y entropies. In other words, SU is defined as:

$$SU(X,Y) = 2 \times \frac{H(X) + H(Y) - H(Y,X)}{H(X) + H(Y)} \tag{3.12}$$

where $H(X)$ and $H(Y)$ are the entropies associated with X and Y, and $H(Y,X)$ is the joint entropy, as defined in 3.7 and 3.10 respectively. Due to the correction factor 2, SU takes values which are normalized in the range $[0,1]$. A value of $SU = 1$ means that the information of one feature completely predicts the other, whereas the value $SU = 0$

indicates that X and Y are uncorrelated.

Generally, SU provides an advantage in overcoming the limitations of Information Gain. However, a weakness of SU is that it is biased towards features with fewer values. In order to examine the interactions of multiple features within the dataset, a generalization of the bivariate measure was introduced in [82]. Like other entropy-based measures, SU works with all data types.

3.4.3 Chi-squared test (CS)

In general, the Chi-squared test is implemented as a way of testing the independence of two nominal features by examining whether the observed distributions are generated by the same underlying distribution. Assume X is a nominal independent feature with r distinct levels and Y is the nominal target variable with c classes. Chi-squared test can be done by applying the following [83]:

$$\chi^2(X) = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{(n_{ij} - \mu_{ij})^2}{\mu_{ij}} \tag{3.13}$$

where n_{ij} is the observed count of observations classified into the j-th class of Y given $x_i \in X$, whereas μ_{ij} indicates the expected frequency. This refers to the expected observations of the j-th class of Y provided there is no relationship between X and Y. Accordingly, μ_{ij} can be calculated using:

$$\mu_{ij} = \frac{n_{*j} n_{i*}}{n} \tag{3.14}$$

where n_{*j} is the number of observations classified into the j-th class of Y, n_{i*} is the number of observations under $x_i \in X$, and n is the total number of observations. If the two features are independent, there should not be a statistically significant difference between the observed frequency and the expected frequency. In order to determine whether the results are statistically significant, the degree of freedom (calculated based on the contingency table size) is used to examine the statistic in the context of the Chi-squared distribution. To apply the Chi-squared test on numeric data, the features need to be discretized.

3.4.4 Minimum Redundancy Maximum Relevance (MRMR)

Minimum Redundancy Maximum Relevance (MRMR) aims to select attributes that are highly correlated with the target variable (maximum relevance), yet show little correlation between the attributes themselves (mininum redundancy). Correlations with the target variable and in-between the features (the optimization criterion) can be measured using IG. The importance of the independent variables X based on the MRMR criterion is defined by [84]:

$$f^{mRMR}(X) = IG(Y;X) - \frac{1}{|S|}\sum_{X_s \in S} IG(X_s;X_i)$$

where Y is the target variable, S is the set of selected features, $|S|$ is the number of selected features, $X_s \in S$ is a particular feature from the feature set S, and features not currently selected are denoted by $X_i \notin S$. Finally, the function $IG(X;Y)$ represents the Information Gain as defined in 3.9 and 3.11. At every step of the MRMR feature selection process, the feature with the highest importance score $max_{X_i \notin S} f^{mRMR}(X_i)$ is added to the selected feature set S. By selecting features that are maximally dissimilar to each other, MRMR reduces feature redundancy while retaining relevant features. However, it is disadvantaged by the sensitivity of its relevance and redundancy measures to outliers. In general, MRMR works with both numeric and nominal inputs, but requires a nominal output.

3.5 Aggregation Techniques

Aggregation refers to the process of combining several values together. Given an input vector of (usually numeric) values, aggregation functions produce a singular output value. The core of aggregation is that the aggregation function's output value should reflect or synthesize "in some way" all individual inputs, depending on the context involved [85]. Due to a vast array of possible aggregation functions, it can be difficult to choose an appropriate aggregation procedure. Optimization-based rank aggregation techniques, for example, are designed to minimize some distance measure to guarantee that the aggregated rank vector is as close as possible to the underlying rank vectors. Other aggregation techniques, such as the Arithmetic Mean, provide simpler aggregations that don't aim to maximize any criterion. In this thesis, we mainly focus on score-based aggregation approaches. While they are less frequent in the literature, score-based

aggregations can provide a higher level of detail than the ranks and be simpler to calculate. Moreover, since the aggregated score vectors can easily be translated into both rank vectors and feature subsets, they are compatible with multiple types of stability metrics. For completeness and further analysis of the methods used, two strictly rank-based aggregation techniques are also included.

3.5.1 Arithmetic Mean (AM)

The Arithmetic Mean (AM) is an aggregation technique that is compatible with both scores and ranks. Given an importance score vector of m values, AM computes the average across input values to determine a final aggregated output score as follows:

$$AM(\boldsymbol{\ell}_j) = \frac{1}{m}\sum_{i=1}^{m} \ell_{ij} \tag{3.15}$$

In Algorithm 1 (WAM), $\boldsymbol{\ell}_j = (\ell_{1j}, \ldots, \ell_{mj})$ is the column j in the score matrix $\mathbb{L}$. That is, the feature importance scores that correspond to feature $V_j \in \{V_1, \ldots, V_p\}$, obtained from $\mathbb{X}_1, \mathbb{X}_2, \ldots, \mathbb{X}_m$.

3.5.2 Geometric Mean (GM)

The Geometric Mean (GM) is another aggregation technique that is compatible with both scores and ranks. Given an importance score vector of m values, GM Calculates the geometric average across importance scores and uses it to determine a final aggregated score as follows: [59]:

$$GM(\boldsymbol{\ell}_j) = \left(\prod_{i=1}^{m} \ell_{ij}\right)^{\frac{1}{m}} \tag{3.16}$$

In Algorithm 1 (WAM), $\boldsymbol{\ell}_j = (\ell_{1j}, \ldots, \ell_{mj})$ is the column j in the score matrix $\mathbb{L}$.

3.5.3 L2 Norm (L2)

In L2 Norm, the importance scores are viewed as an n-dimensional vector, then the Euclidean norm for that vector is calcuated as follows [59]:

$$L2(\boldsymbol{\ell}_j) = \left(\sum_{i=1}^{m} \ell_{ij}^2\right)^{\frac{1}{2}} \tag{3.17}$$

In Algorithm 1 (WAM), $\boldsymbol{\ell}_j = (\ell_{1j}, \ldots, \ell_{mj})$ is the column j in the score matrix $\mathbb{L}$.

3.5.4 Stuart

In Stuart aggreagtion, input rank vectors (for the same ranked feature across experimental samples) are compared to a baseline of random feature ranks. Subsequently, the beta distribution is used to assign significance scores for each feature [60]. Assume $r_j = (r_{1j}, \ldots, r_{mj})$ is the rank vector that corresponds to the feature $V_j \in \{V_1, \ldots, V_p\}$. That is, r_j is derived from the column j in the score matrix $\mathbb{L}$. Furthermore, we normalize the rank vector by dividing by the maximal rank value p, such that the maximal value in the rank vector r_j will be 1. To aggregate the ranks for the feature V_j, let $r_{(1j)}, \ldots, r_{(mj)}$ be a reordering of r_j such that $r_{(1j)} \leq \ldots \leq r_{(mj)}$. Moreover, let $\hat{r}_j$ be a rank vector generated by the null model, i.e. the ranks are sampled from the uniform distribution. We are thus interested in the probability $\hat{r}_{(sj)} \leq r_{(sj)}$, which we denote as $\beta_{s,m}(r_j)$. Since $\hat{r}_{(sj)}$ is the order statistic of m independent random variables uniformly distributed over $[0,1]$, this probability may be evaluated using the beta distribution or taken as a binomial probability under the null model. Finally, we compute the score for ranking the feature V_j using $\rho(r_j) = \min_{s=1,\ldots,m} \beta_{s,m}(r_j)$. Based on the ρ score distribution, p-values can also be calculated for each ρ score as follows [61]:

$$\Pr[X \leq \rho] = 1 - \Pr[\hat{r}_{(1)} \leq 1 - \beta_{m,m}^{-1}(\rho), \ldots, \hat{r}_{(m)} \leq 1 - \beta_{m,1}^{-1}(\rho)] \tag{3.18}$$

where $\hat{r}$ is an observation from the uniform distribution with size m and $\beta_{s,m}^{-1}(\rho)$ is a quantile of $\text{Beta}(s, m-s+1)$ distribution. The p-values are used to decide whether the ranking of a particular feature is statistically significant and to re-rank the features in the final aggregated rank vector.

3.5.5 Robust Rank Aggregation (RRA)

The Robust Rank Aggregation (RRA) is a Stuart variant that uses Bonferroni adjustments to obtain a suitable aggregated vector even when the input rank vectors are inaccurate or irrelevant [61]. First, for each feature V_j, $\rho(r_j)$ is calculated like in Stuart as follows:

$$\rho(r_j) = \min_{s=1,\ldots,m} \beta_{s,m}(r_j),\ \beta_{s,m}(x) := \sum_{\ell=s}^{m} \binom{m}{\ell} x^{\ell}(1-x)^{m-\ell} \tag{3.19}$$

Unlike Stuart, the Bonferroni correction is then applied to determine an upper bound on the corresponding p-value for each score ρ independently. For this purpose, each ρ score is multiplied by the number of input vectors (i.e. the number of bootstraps). In most circumstances, the Bonferroni correction is a suitable compromise between efficiency and precision.

3.6 Classification Learning Algorithms

While the methodology discussed in this chapter so far is applicable to both regression and classification problems, the experimental analysis will solely focus on classification datasets. Hence, the learning algorithms used for the experimental analysis will consist of classification algorithms. In this section, we discuss the most popular classifiers used in this work.

3.6.1 Logistic Regression

Logistic Regression is a commonly used statistical model that uses a logistic (sigmoid) function to represent the likelihood of a class or event occurring. In particular, the sigmoid function is used to map the predicted values into probabilities between 0 and 1 given by:

$$S(Z) = \frac{1}{1+e^{-Z}} = \frac{e^{Z}}{e^{Z}+1}$$

Assume that Z represents a regression function or a linear combination of multiple explanatory features $\{V_1, \ldots, V_p\}$ such that:

$$Z = B_0 + B_1.V_1 + B_2.V_2 \ + \ \ldots \ + \ B_p.V_p$$

Then, the general logistic function can be given by:

$$P(Y=1) = S(Z) = \frac{1}{1+e^{-(B_0+B_1.V_1+B_2.V_2 \ + \ \ldots \ + \ B_p.V_p)}}$$

In binary Logistic Regression, $P(Y=1)$ refers to the probability of the dependent variable Y representing a success/positive class label rather than a failure/negative class label. Moreover, the logit (log odds) function can be defined as the inverse of the logistic

function where:

$$S^{-1}(P(Y=1)) = ln\left(\frac{P(Y=1)}{1-P(Y=1)}\right) = B_0 + B_1.V_1 + B_2.V_2 \ + \ ... \ + \ B_p.V_p$$

We may choose a threshold, such as 0.5, to make the Logistic Regression into a linear classifier. For instance, assuming the classes are 1 and -1, we predict $Y = 1$ when $P(Y = 1) > 0.5$ and $Y = -1$ when $P(Y = 1) < 0.5$. Meanwhile, the Logistic Regression coefficients $B_j, j = 1, \ldots, p$, are commonly estimated using the maximum likelihood estimation approach.

3.6.2 Naive Bayes

Naive Bayes is a probabilistic classifier based on the Bayes theorem [86] that works with both binary and multiclass classification problems. Due to its simplicity, it is a fast machine learning algorithm which can be utilized for large datasets. The classifier is called 'naive' since features are assumed to be class-conditionally independent. In other words, Naive Bayes assumes that the existence of one feature in a class has no bearing on the presence of another one. Moreover, every feature is given the same level of importance. These assumptions work well with the Bayes theorem, which uses information about prior conditions related to a feature to describes its posterior probability. Given an observation vector $x = (x_1, x_2, \ldots, x_p)$ that corresponds to the values of the features $\{V_1, \ldots, V_p\}$ and the class $y \in Y$ (i.e. $x = x_i$ for some row i in the input dataset $\mathbb{S} \equiv (X, Y)$). Then, the Bayes theorem states that:

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)} \tag{3.20}$$

Due to the naive assumption, it becomes possible to rewrite $P(x|y)$ as follows:

$$P(x|y) = P(x_1|y)P(x_2|y)\cdots P(x_p|y)$$

Therefore, Equation 3.20 can be rewritten as:

$$P(y|x) = \frac{P(x_1|y)P(x_2|y)\cdots P(x_p|y)P(y)}{P(x_1)P(x_2)\cdots P(x_p))}$$

As we solve for $P(y)$, $P(x)$ remains a constant. Therefore, the posterior probability can be taken as:

$$P(y \mid x_1, x_2, \ldots, x_n) \propto P(y) \prod_{j=1}^{p} P\left(x_j \mid y\right)$$

At this point, the Naive Bayes classifier aims to choose the class y with the maximum probability. Thus, we find the maximum y value using the following function:

$$y = \text{argmax}_y P(y) \prod_{j=1}^{p} P\left(x_j \mid y\right)$$

By going through this process, the Naive Bayes classifier can produce good results very quickly if the assumptions are met and would not require as much training data as other classifiers.

3.6.3 Random Forest

Random Forests are decision tree ensemble models in which each tree is trained on a random subsample of the dataset to predict the class of the target variable Y. Decision trees work by splitting the datasets into small subgroups using feature-based criteria. This splitting process continues until no additional gains are possible or a predetermined rule is satisfied, such as the tree's maximum depth. The three components of a decision tree are root nodes which represent the entry points to the data, inner nodes which are obtained after splitting a root node, and leaf nodes where the class decision can be made as further splitting is not possible. Moreover, there are several ways for selecting the features used at each split, using impurity criterions such as entropy or Gini's index. The issue with decision trees, however, is that they tend to overfit their training sets, resulting in low bias but large variance.

To minimize the variance of decision trees, Random Forests utilize a majority vote on identically distributed decision trees. In Random Forest, subsamples are obtained from the training dataset with replacement and the output of each decision tree is averaged to increase the predicted accuracy and control over-fitting [87]. While Random Forest models are effective at preventing overfitting, they can be expensive to construct.

3.6.4 Support Vector Machine (SVM)

Assuming that n is the number of observations and p is the number of features, each observation is represented as a point in p-dimensional space in Support Vector Machine (SVM). The goal of SVM is to find the best hyperplane for separating data into different classes of the target variable. As a result, the chosen hyperplane optimize the distance between data points of various classes. This distance is known as the margin. Only the points closest to the hyperplane are used to compute the margin; these points are referred to as the support vectors.

To illustrate, assume that $\mathbf{x_i}^{\mathrm{T}} = (x_{i1}, \ldots x_{ip})$ represents an observation vector (row) in the input dataset $\mathbb{S} \equiv (X, Y)$, where $Y = \{-1, +1\}$ represents a binary class variable. Then, $\mathbf{x_i}$ is positively classified if $y_i = +1$, and negatively classified if $y_i = -1$. Moreover, using some weight vector $\mathbf{w}$ and bias b, the training data can be separated using the hyperplane $\mathbf{w}^{\mathrm{T}}\mathbf{x_i} + b = 0$. Our goal is to find $\mathbf{w}$ and b for the optimal hypberplane. To do this, let the marginal hyperplanes, H_1 and H_2, be taken as:

$$H_1 : \left(\mathbf{w}^{\mathrm{T}}\mathbf{x_i} + b\right) = +1$$

and

$$H_2 : \left(\mathbf{w}^{\mathrm{T}}\mathbf{x_i} + b\right) = -1$$

such that the marginal hyperplanes are passing through the nearest points in each class. Once we subtract the data points to get the distance between the two marginal hyperplanes, we obtain the quantity $\frac{2}{\|\mathbf{w}\|}$. This is known as the margin. We are interested in maximizing this in order to identify the optimal marginal hyperplanes which ensure no data points are misclassified. Alternatively, we maximize the margin such that no point can have a distance to the hyperplane smaller than the margin. In this manner, we find the values of $\mathbf{w}$ and b that maximize the following function [88]:

$$(w^*, b^*) \max \frac{2}{\|\mathbf{w}\|} y_i \begin{cases} +1 \text{ where } \mathbf{w}^{\mathrm{T}}\mathbf{x_i} + b \geq +1 \\ -1 \text{ where } \mathbf{w}^{\mathrm{T}}\mathbf{x_i} + b \leq -1 \end{cases}$$

This can also be expressed as a minimization problem via the following function:

$$(w^*, b^*) \min \frac{\|\mathbf{w}\|^2}{2} + C \sum_{i=1}^{n} \xi_i$$

subject to $y_i\left(\mathbf{w}^{\mathrm{T}}\mathbf{x_i}+b\right)\geq 1-\xi_i$. Note that C is a regularization (penalty) parameter that regulates the trade-off between maximization of the margin and minimization of the training error, and $\xi_i \geq 0, i=1,\ldots,n$, is a slack variable that indicates how much the constraints are violated.

Chapter 4. Experimental Analysis

This chapter highlights both the classification performance and stability behavior results obtained from implementing the WAM and BAM algorithms on different classification problems. In addition, different aggreagtion procedures are tested on the WAM, where their accuracy and stability differences are compared and analyzed across the different feature selection methods.

4.1 Experimental Design

Due to their computational efficiency, four traditional filter feature selection techniques are used in the analysis of the WAM and BAM frameworks. As presented in section 3.5, these methods are: Information Gain (IG), Symmertric Uncertainty (SU), Minimum Redundancy Maximum Relevance (MRMR), and Chi-Square test (CS) (note that numeric features are first discretized using fixed width binning in order to obtain the CS scores). In the experimental analysis, the dataset $\mathbb{S}$ is first divided into a training dataset $\mathbb{X}$, and a testing dataset $\mathbb{T}$, where two thirds of the dataset is used for obtaining the importance scores (i.e. $\mathbb{X}$) and one third is used for testing the model (i.e. $\mathbb{T}$). The training set is bootstrapped 1000 times ($m = 1000$), such that each sample is of the same size as the training set.

By applying the WAM and BAM frameworks as described in sections 3.2 and 3.3 on all bootstrap samples, we obtain an aggregated score set $\{a_1, \ldots, a_p\}$ and a corresponding rank vector $r = (r_1, \ldots, r_p)$. Based on the rank vector r, we select only the top $100k\%$ of the variable set $\{V_1, \ldots, V_p\}$ and test their classification performance on the previously segmented testing set $\mathbb{T}$. The classifiers used for this step are Logistic Regression, Naive Bayes, Random Forest, and SVM, described in section 3.7. The testing data is also divided into 5 folds for stratified K-fold cross-validation, where stratification is used to ensure that each fold encompasses equal proportions of the target classes. Moreover, the metric used for determining the classification performance is the *Accuracy* score (i.e. the proportion of correct predictions). By removing different percentages of features from the classifier depending on the feature rankings, we can observe the optimal percentage of features that can be used for obtaining a reasonable accuracy. In the experimental analysis, ten different k thresholds are used in the testing step, resulting in subsets containing the top $\{10\%, 20\%, \ldots, 100\%\}$ of the total features. Here, $100k\%$ where $k = 1.0$, refers to the baseline model where all features are used

and none of the feature selection methods are implemented. In addition, to provide a more general picture of the classification performance of the two ensembles, a weighted average of the accuracy scores is taken across the feature selection methods and for each classifier. In this Weighted Average Accuracy $(WAcc)$, each accuracy score is weighted by the percentage of removed features such that:

$$WAcc = \frac{\sum_{i=1}^{d} W_i Acc_i}{\sum_{i=1}^{d} W_i} \tag{4.1}$$

where the weights $W_i = \{0.1, 0.2, \ldots, 0.9\}$ are the nine threshold values ($d = 9$), excluding $k = 1.0$. Note that the weights are taken to be $1 - k$, rather than k, in order to give more weight to feature selections methods that result in good accuracy at higher feature reduction percentages. Furthermore, due to its simplicity and effectiveness, the Arithmetic Mean (AM) [51], [66] is used for aggregating the feature importance scores obtained from the tested feature selection methods in the WAM and BAM frameworks. In other words, $AT = \{AM\}$ for comparing WAM and BAM in the first part of the experimental analysis. Then, using the WAM ensemble framework, different aggregation methods are assessed and compared in terms of their accuracy and stability behavior.

In the process of examining the influence of the aggregation procedure, 1000 bootstrap samples are created to aggregate the feature importance scores for each feature selection method using the five aggregation strategies: Arithmetic Mean (AM), Geometric Mean (GM), L2 Norm (L2), Robust Rank Aggregation (RRA), and Stuart aggregation. To assess the classification accuracy of the feature selection, a 5-fold cross-validation process is then implemented using the Naive Bayes classifier. Moreover, the bootstrap samples are embedded within a 100-fold cross-validation approach to compute the stability. On each iteration, 99 folds are used to generate a training set and one fold is used to generate a testing set. By going through all iterations, the stability of the same feature selection ensemble under different aggregation techniques is evaluated using the final 100 aggregated rank vectors. This procedure is followed across the different score-based (AM, GM, L2) and rank-based (RRA, Stuart) aggregations. Due to their contrasting performance in the WAM and BAM ensembles, the two filter techniques Information Gain (IG) and Minimum Redundancy Maximum Relevance (MRMR) are used to analyze the stability influence of the different aggregation procedures.

The experimental framework was performed in the open-source statistical programming language R. The experimental environment was Windows 10, 64-bit, 16 GB RAM, Intel(R) Xeon E-2124 (3.30GHz). The evaluation involved twelve classification problems with real datasets from different application domains, ten of which are binary and two with multiclass target variables. Note that features of near-zero variance were removed prior to the analysis.

4.2 Experimental Datasets

Our experimental analysis consists of the datasets illustrated in Table 4.1. The datasets have been acquired from various sources pertaining to different classification problems, including a binarization of multiclass problems, ML challanges, and datasets from the UCI Machine Learning Repository. As seen in the Table 4.1, the datasets contain both numerical and nominal values with various dimensionalities and numbers of observations. Thus, they provide an interesting benchmark for the investigation of the proposed framework's behavior and its characteristics.

The datasets Jasmine, Philippine, and Ada are all part of the ChaLearn AutoML Challenge Series (2015–2018) [89], which consists of six rounds of a machine learning competition with increasing levels of difficulty and computational constraints. The data was chosen to represent a wide range of application domains including biology, medicine, and ecology, among others. While not in their original variable-length representations, the classification involves processing of text, speech, and video data; this data was preprocessed into numerical features.

The Scene dataset [90] consists of 2400 images from both personal and the COREL stock image collections. Spatial color moments in Luv space were recorded as features. The image was divided into 49 blocks using a 7x7 grid after being converted to Luv space. The first and second moments (mean and variance) of each band were calculated, which correspond to a low-resolution image and inexpensive texture features. As a result, each image has a 294-dimension feature vector. The classification task was taken to be the identification of a scene as Urban. Similarly, the Image dataset [91] consists of 2,000 natural scene images belonging to the classes desert, mountains, sea, sunset, and trees. While some of the photographs were taken from the COREL image collection, others were gathered via the Internet. The binary classification task was the identification of the sea in the images.

The Musk dataset [92] describes 102 compounds, 39 of which human experts have determined to be musks and the remaining 63 to be non-musks. A single molecule can take on a variety of shapes due to the rotational nature of bonds. Therefore, the dataset was constructed by generating 6,598 conformations from each of the molecules' low-energy conformations. Then, a feature vector that characterizes each precise shape, or conformation, of the molecules was retrieved. The objective is to tell whether new compounds will be musks or not.

The Optidigits dataset [92] consists of normalized bitmaps of handwritten digits preprocessed by NIST software. The number of pixels was counted in each block of a 32x32 bitmap that was partitioned into 4x4 nonoverlapping blocks. With each element being an integer between 0 and 16, this produced an 8x8 input matrix. Meanwhile, the features in the Satellite dataset [92] were taken from satellite observations. Each image, specifically, was captured using four different light wavelengths: two visible (green and red) and two infrared. The binary dataset's objective is to categorize the image into the observed region's soil category (Normal vs Anamoly). Red soil, gray soil, damp gray soil, and very damp gray soil were used to define the 'Normal' class. 'Anomalies' were drawn from the two semantically distinct groups "cotton crop" and "soil with vegetation stubble."

The Splice dataset [92] consists of 60 variables, each of which represents a group of DNA nucleotides. The aim is to determine whether the middle of the sequence is a splice junction and if so, what kind it is. A splice junction is a location in a DNA sequence where "superflous" DNA is eliminated during the production of proteins. Exon/intron (EI) and intron/exon (IE) sites are the two different kinds of splice junctions found in DNA sequences; exon is the section of the sequence that is kept, whereas intron is the component of the sequence that is spliced out.

Finally, Indian Pines [93] is a multiclass dataset for segmenting hyperspectral images. The input data consists of 145x145 pixel hyperspectral bands covering a single landscape in Indiana, United States. The data collection contained 220 spectral reflectance bands for each pixel that represent various portions of the electromagnetic spectrum. The classes include land-use types for alfalfa, corn, grass, hay, oats, soybeans, trees, and wheat. Similarly, the Semeion multiclass dataset [92] consists of 1593 handwritten digits (0-9) from 80 individuals that were scanned and enlarged to a 16x16

size image. Each of the 256 variables describes a single pixel and its associated value.

Table 4.1. Datasets Description

Dataset name and source	No. observations	No. Features	No. Classes	Dimensionality*
Jasmine[1]	2984 (1492/1492)	145	2	0.048592
Image[2]	2000 (1420/580)	140	2	0.070000
Scene[3]	2407 (1976/431)	295	2	0.122559
Musk[4]	6598 (5581/1017)	170	2	0.025765
Philippine[1]	5832 (2916/2916)	309	2	0.052984
Ionosphere[4]	351 (126/225)	34	2	0.096866
Optdigits[2]	5620 (572/5048)	64	2	0.011388
Satellite[2]	5100 (75/5025)	37	2	0.007255
Ada[1]	4147 (1029/3118)	49	2	0.011816
Splice[2]	3190 (1535/1655)	61	2	0.019436
Indian Pines[2]	9144	221	8	0.024168
Semeion[4]	1593	257	10	0.161330

* Dimensionality is the ratio of features to number of observations. Superscripts indicate the data sources as follows:
[1] automl.chalearn.org, [2] www.openml.org, [3] mulan.sourceforge.net, [4] archive.ics.uci.edu.

4.3 Classification Performance Results

The accuracy scores (percentage of true predictions) after applying the WAM and the BAM on each dataset are shown in Figures 4.1-4.12. The WAM findings are represented by the curves corresponding to the different feature selection methods: IG, SU, MRMR, and CS. The BAM findings, on the other hand, are shown in each panel by a single BAM curve. The four plots for each dataset illustrate the four classifiers that were used: Logistic Regression, Naive Bayes, Random Forest, and SVM. Note that for Figures 4.11-4.12, Logistic Regression is not included due to being multiclass datasets. Overall, the accuracy values, which are averaged across the 5-folds in the cross-validation, are displayed against ten distinct $100k\%$ thresholds in the testing stage to demonstrate the classification performance for different feature subsets. In addition, the Weighted Average Accuracy values in Table 4.3 present a more general overview of the WAM and BAM performances.

Conversely, the accuracy scores after applying the WAM to each dataset using different aggregation strategies are shown in Figures 4.13-4.24. These WAM findings are represented by the curves corresponding to the various aggregation techniques: AM, GM, L2, RRA, and Stuart aggregation. The four plots for each dataset illustrate the four feature selection approaches that were used: IG, SU, CS, and MRMR. The accuracy values, which are averaged across the 5-folds in the cross-validation, are displayed against

ten distinct $100k\%$ thresholds in the testing stage to demonstrate the classification performance for different feature subsets.

The running times for WAM and BAM, respectively, are also presented in Table 4.2. In the experiments, WAM takes an average of around 3,030 seconds to run, whereas BAM averages at nearly 3,012 seconds. Since BAM requires an additional aggregation step across the multiple feature selection algorithms, it is significantly slower than WAM across all datasets (Wilcoxon-test p-value $= 0.0002$). Overall, the computational costs of the ensemble frameworks are mostly controlled by the feature selection methods utilized and the dataset composition on which it is used (see Table 4.1).

Table 4.2. Computational running times for WAM and BAM frameworks

Dataset name	**WAM (seconds)**	**BAM (seconds)**
Jasmine	1838	1840
Image	1816	1843
Scene	4277	4332
Musk	4307	4379
Philippine	7595	7601
Ionosphere	377	377
Optdigits	1112	1117
Satalite	786	799
Ada	698	698
Splice	1009	1019
Indian Pines	8744	8753
Semeion	698	698

In the following two subsections, we analyze and compare the performance of the WAM and BAM algorithms, in terms of their classification accuracy and their identification of optimal feature subsets (subsection 4.3.1). In subsection 4.3.2, we also analyze and compare the classification performance of the different aggregation techniques within the WAM framework.

4.3.1 Comparison of the classification performance for WAM and BAM

As can be seen in Figures 4.1-4.12, the best classifier for most datasets appears to be the Random Forest classifier averaging at around 88% accuracy throughout the different experiments. In contrast, SVM achieves largely better accuracy scores in the Image, Scene, and Optidigts datasets (see Figures 4.2-4.3 and Figure 4.7). However, the Naive Bayes classifier, in particular, seems to be the most influenced by the ensemble framework.

In the majority of datasets, we generally observe that both ensemble feature selection frameworks enhance the baseline ($k = 1.0$) classification performance to some degree. The efficiency of the WAM is particularly apparent when using the Naive Bayes classifier, where the accuracy values exhibit a steeper increase than that of other classifiers across multiple datasets. For example, in the Ionosphere dataset in Figure 4.6, the Naive Bayes accuracy climbs from roughly 0.81 at baseline to nearly 0.90. In the Satellite dataset in Figure 4.8, it climbs from around 0.89 baseline accuracy to nearly 0.99. Similarly, it can be observed that BAM also improves the baseline accuracy of the model, particularly under the Logistic Regression classifier. For example, in the Jasmine dataset in Figure 4.1, BAM improves the Logistic Regression accuracy from approximately 0.75 at baseline to nearly 0.9. In the Philippine dataset in Figure 4.5, the Logistic Regression accuracy climbs from around 0.68 at baseline to over 0.72 under the BAM.

Overall, the experimental results reveal that the ensembling of bootstrap samples and aggregating of feature importance scores within and between feature selection methods helps improve the baseline classification performance and/or allows for removing a large proportion of insignificant features without compromising the accuracy (e.g. Image, Musk, and Satellite datasets). However, while the WAM and BAM produce comparable results, the aggregated feature selection methods under WAM appear to somewhat outperform the BAM in terms of the maximum accuracy scores. In particular, for most of the datasets, at least one feature selection method aggregated under WAM has produced better accuracy values than those obtained by the BAM. In the Philippine and Scene datasets, for example, the aggregated CS clearly outperforms BAM as seen in Figure 4.5 and Figure 4.3, respectively. In some datasets, such as Jasmine in Figure 4.1

or Ionosphere in Figure 4.6, the aggregated SU yields better accuracy values, whereas in others such as Optidigits in Figure 4.7 and Splice in Figure 4.10, the aggregated MRMR outperforms the other feature selection methods. Similar patterns can be observed across the rest of the binary datasets, where the aggregated IG findings exhibit the highest correlation with the BAM results and the aggregated MRMR exhibit the least correlation. Alternatively, in the multiclass datasets, both the WAM and BAM frameworks produce overlapping curves across most classifiers in Figures 4.11-4.12. This might be attributed to the consensus of the other feature selection techniques in the multi-class datasets, given that the WAM simply averages the outcomes of the other methods. On a more general level, the BAM appears to be the middle-of-the-pack strategy when compared with the WAM in terms of its classification performance.

However, when compared comprehensively, using the Weighed Average Accuracy scores in Table 4.3, the distinction between the WAM and BAM becomes less prominent. Although the best weighted average scores (bolded across the rows in Table 4.3) are still mainly dominated by the WAM methods, the BAM still produces relatively good performance in a number of datasets such as Musk, Splice, and Semeion. Moreover, the BAM continues to be the middle-best performing strategy even in the datasets in which a feature selection approach under WAM outperforms it. Statistically, a repeated-measures ANOVA reveals that differences in the accuracy scores between the feature selection algorithms are significant under the Logistic Regression and SVM classifiers (Bonferroni-adjusted p-values 0.013 and 0.0372 respectively). When further investigated, the significant pairwise differences were generally attributed to a difference between MRMR and each of the other feature selection methods (IG, SU, CS, BAM) under SVM (adjusted p-values 0.044, 0.04, 0.019, and 0.047, respectively). In addition, there were significant differences between the BAM and MRMR methods under the Logistic Regression classifier (adjusted p-value 0.009), highlighting the interaction between both the classification algorithm and the utilized feature selection within the ensemble.

On the other hand, in terms of selecting the optimal $100k\%$ threshold based on the accuracy values, the WAM and BAM produce nearly consistent results. Across the Jasmine, Scene, and Splice datasets, removing the least significant features results in a comparable improvement in the performance of most classifiers in Figure 4.1, Figure 4.3 and Figure 4.10, respectively. Although the aggregated feature selection techniques

reveal some variable patterns depending on the amount of data retained, the overall trend exhibits a stable or improved accuracy performance followed by a sharp decrease in the overall accuracy once the number of used features falls below a certain threshold. In the Philippine and Jasmine datasets in Figure 4.5 and Figure 4.1, there is a clear trend in which the classification accuracy decreases drastically around the 40% line, demonstrating that removing more than 60% of the features reduces the trained model's performance significantly. In the Scene and Optidigits datasets in Figure 4.3 and Figure 4.7, most classifiers agree that roughly half of the top features should be retained. Likewise, the optimal feature reduction threshold is nearly 80% of the features in the Satellite and Ada datasets in Figure 4.8 and Figure 4.9, whereas it is approaching 20% in the mutliclass datasets in Figure 4.11 and Figure 4.12. That is to say, the optimal feature reduction percentage appears to be dependent on the dataset used. It's also worth noting that the results of the experimental analysis demonstrate that the performance of a specific feature selection approach is similarly data dependent. None of the utilized feature selection methods produces the best accuracy values across all datasets. However, given that highly significant features are expected to retain comparable ranks throughout different feature selections, there is still some clear overlap in accuracy values between the feature selection methods. In fact, the relatively good performance of the weighted average accuracy scores across Table 4.3 indicate that both the WAM and BAM succeed in identifying many of these features.

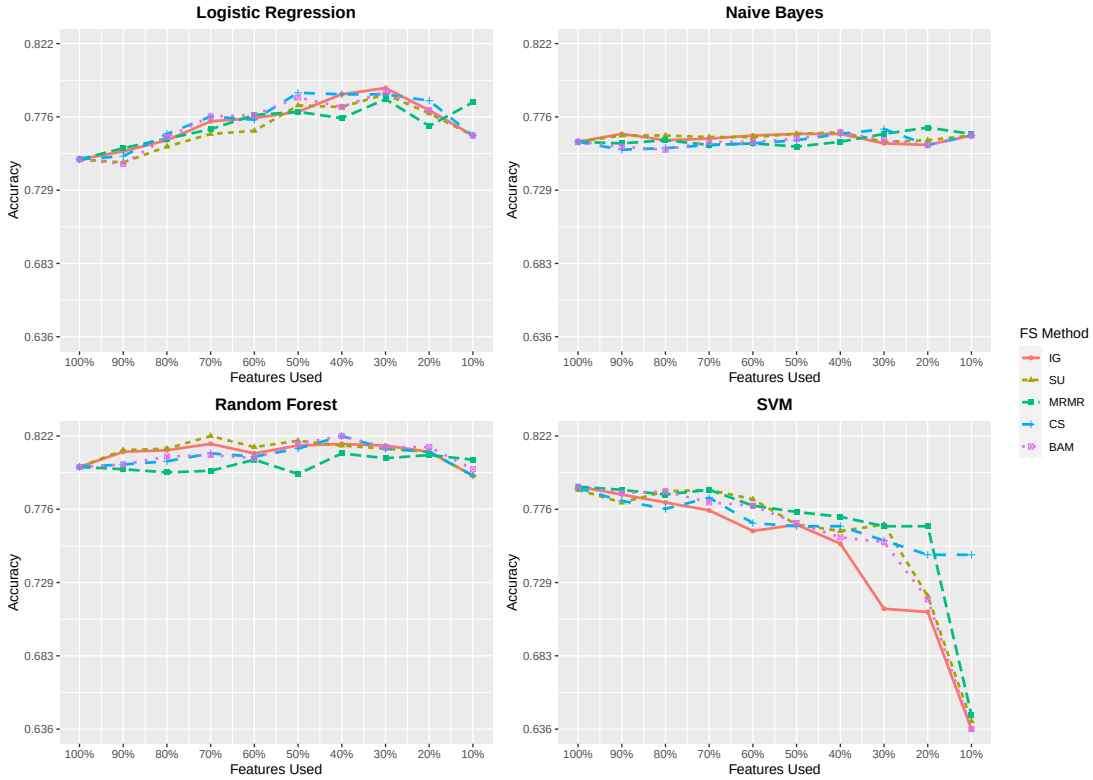

Figure 4.1. Jasmine dataset classification results (by FS)

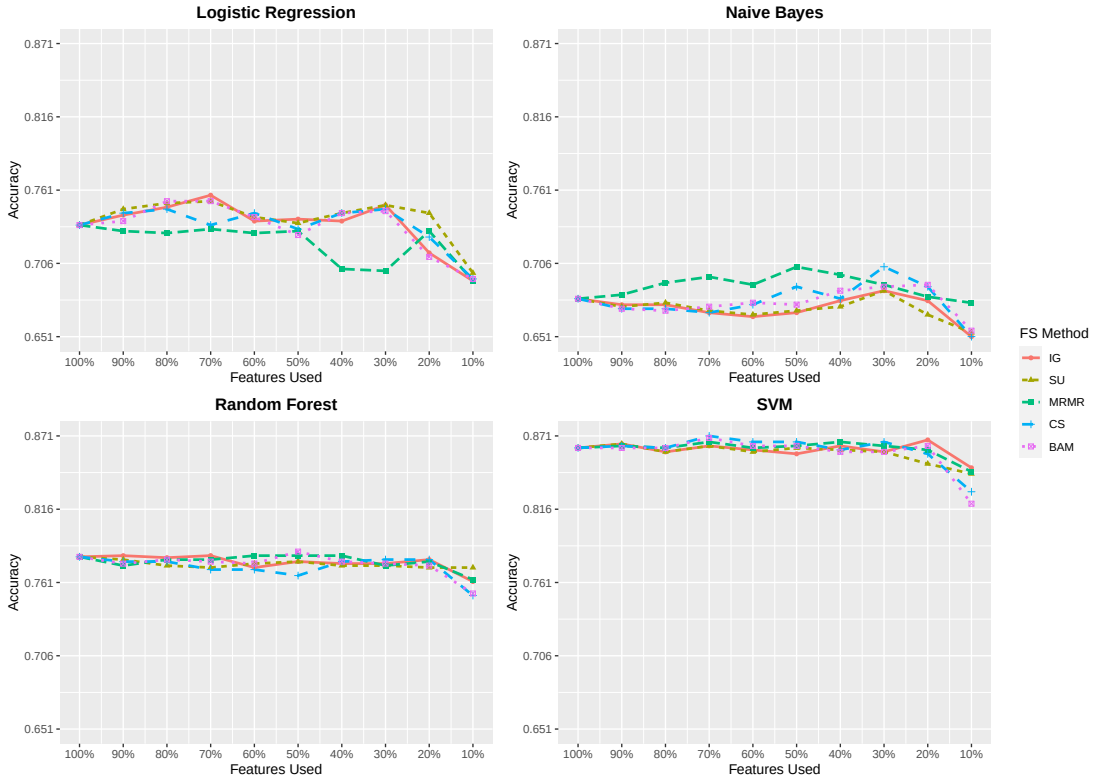

Figure 4.2. Image dataset classification results (by FS)

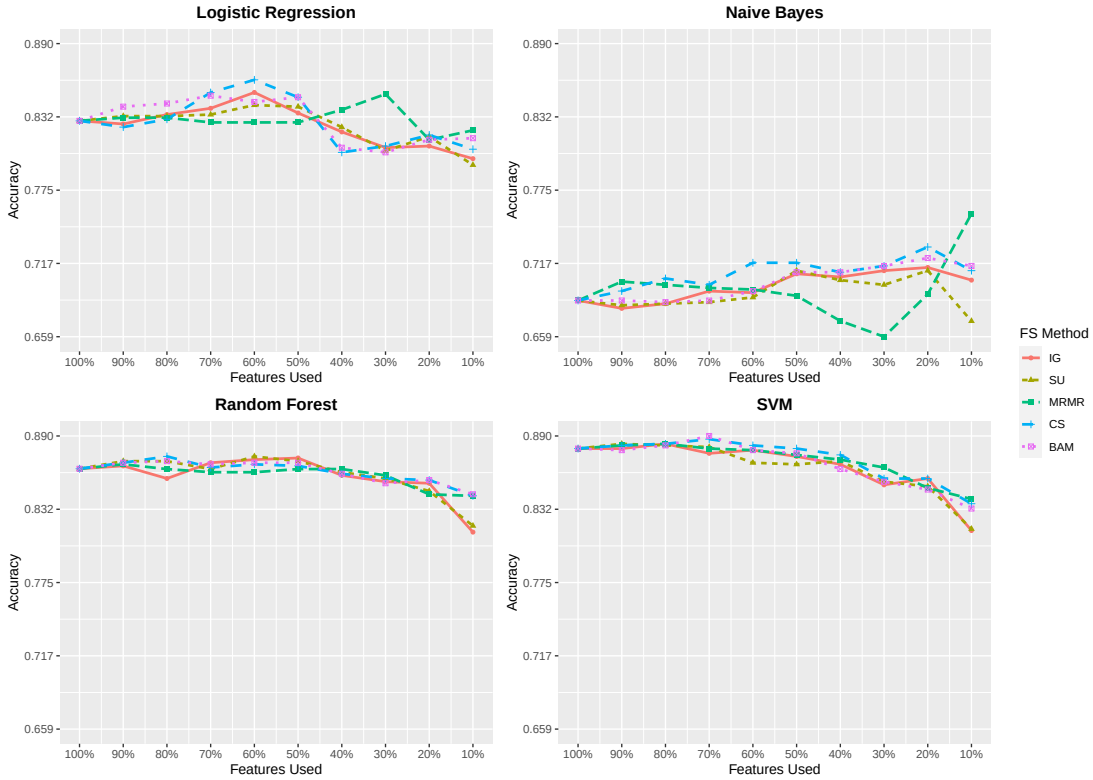

Figure 4.3. Scene dataset classification results (by FS)

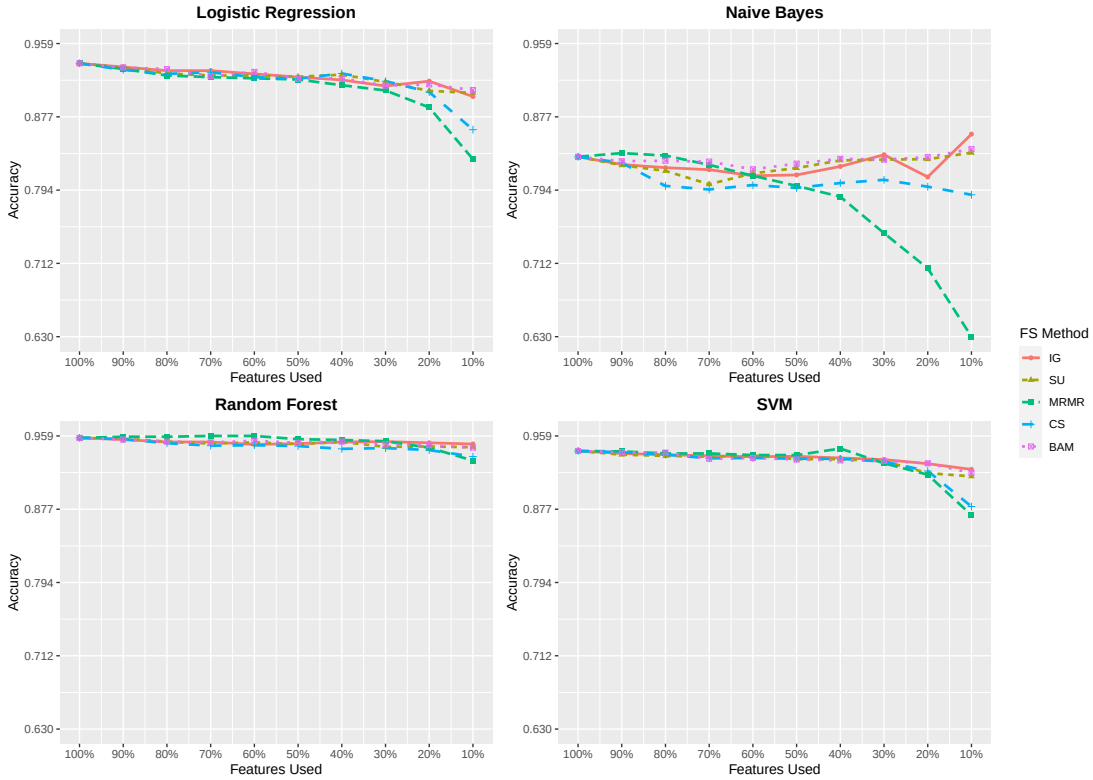

Figure 4.4. Musk dataset classification results (by FS)

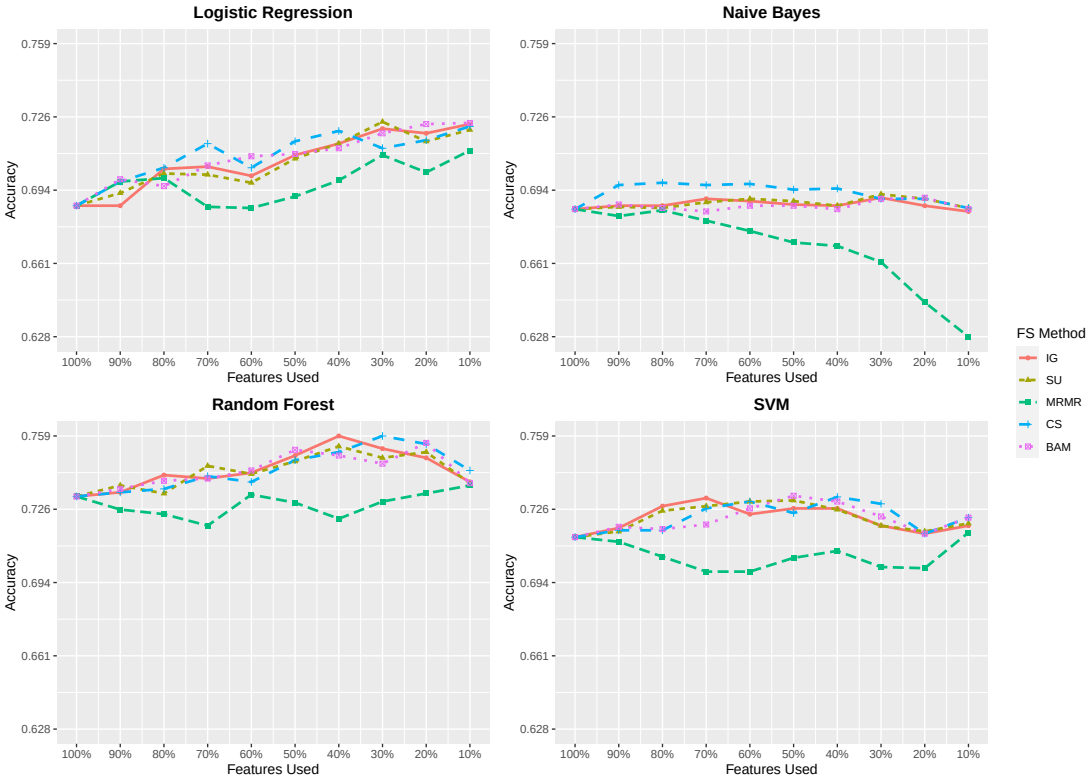

Figure 4.5. Philippine dataset classification results (by FS)

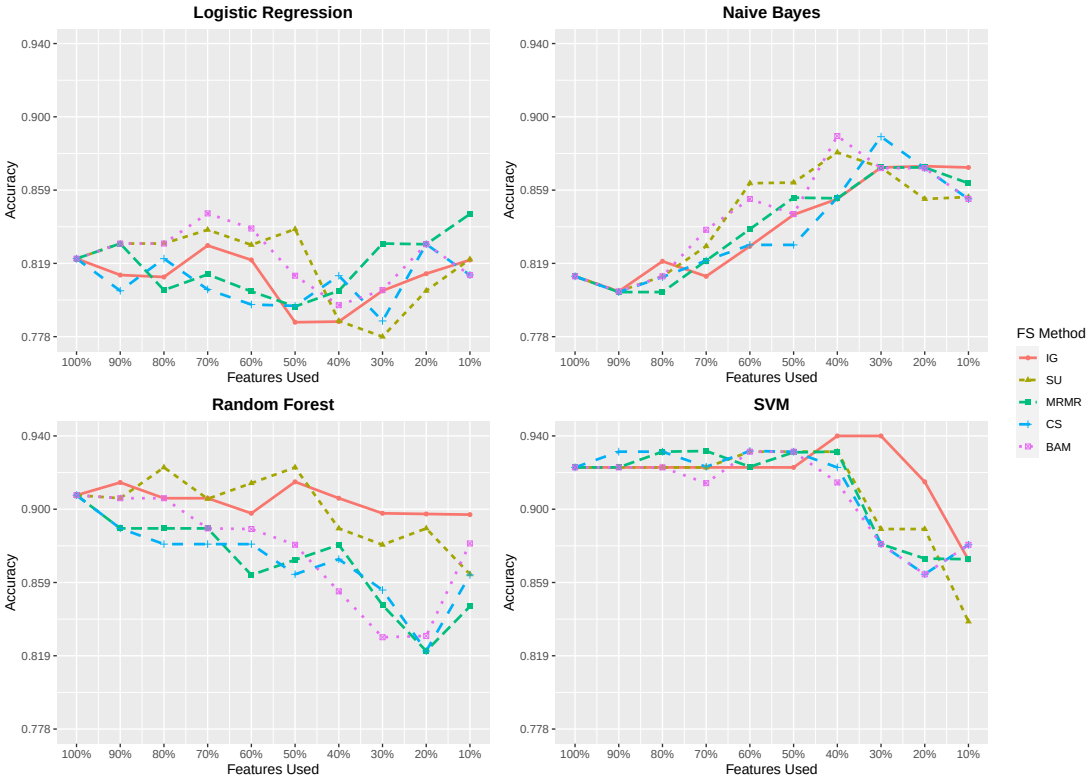

Figure 4.6. Ionosphere dataset classification results (by FS)

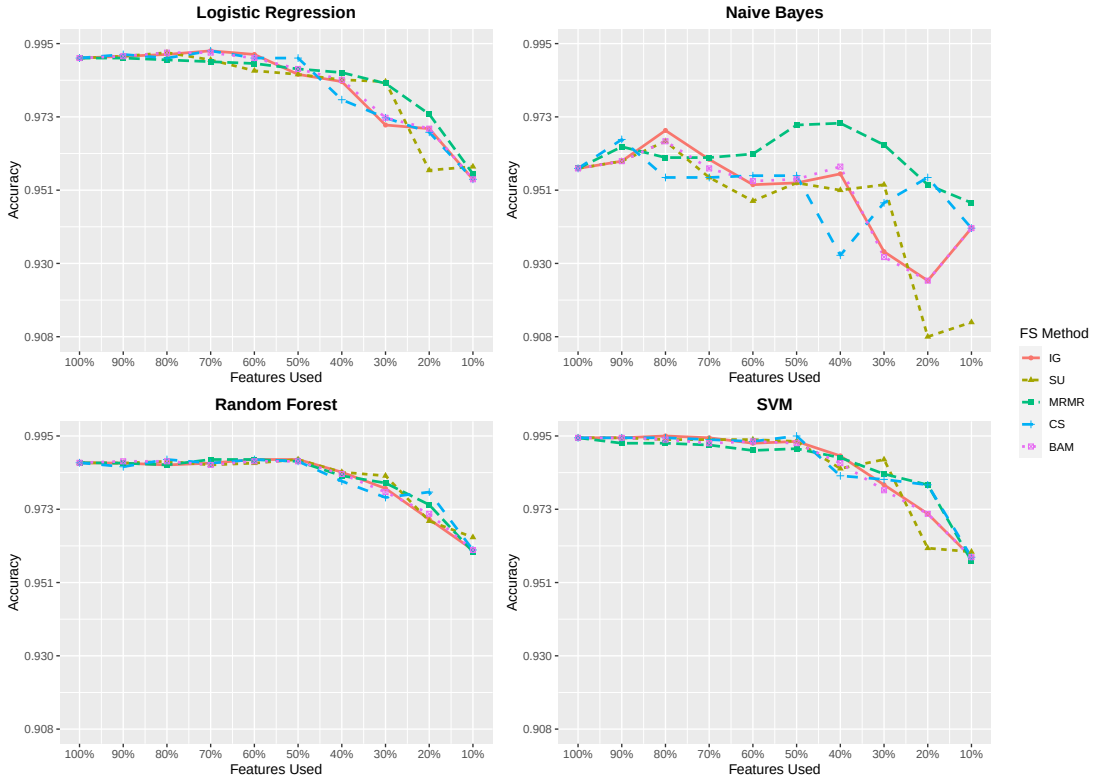

Figure 4.7. Optdigits dataset classification results (by FS)

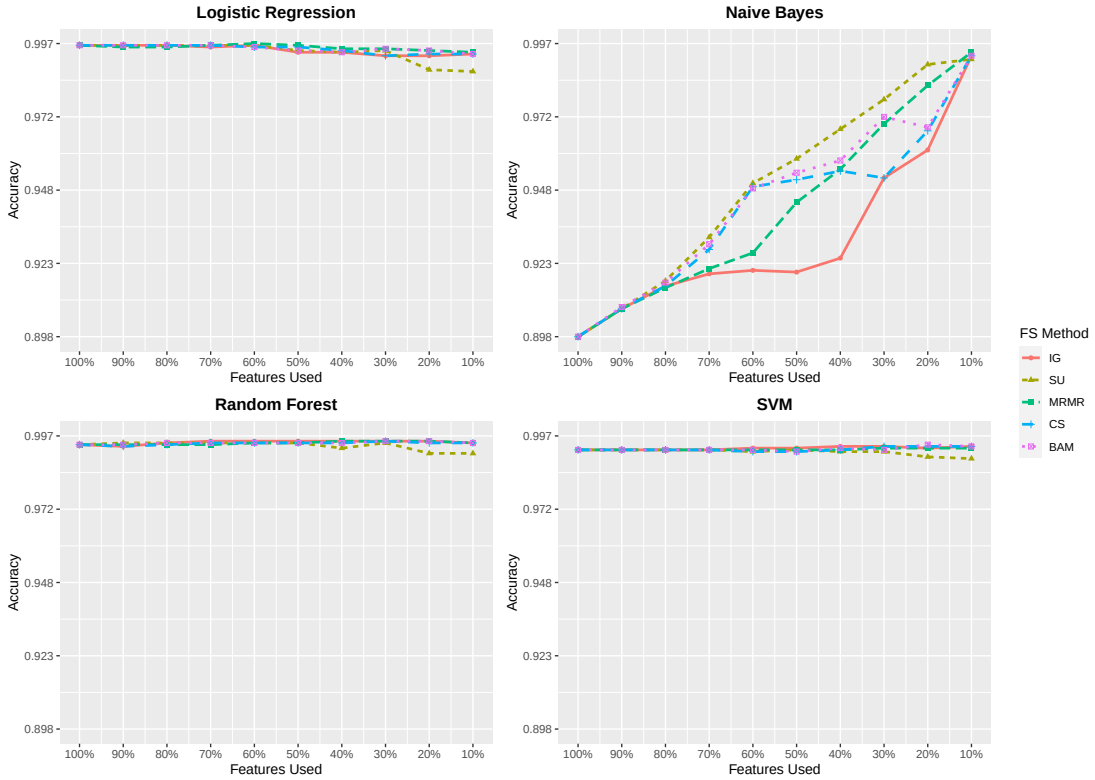

Figure 4.8. Satellite dataset classification results (by FS)

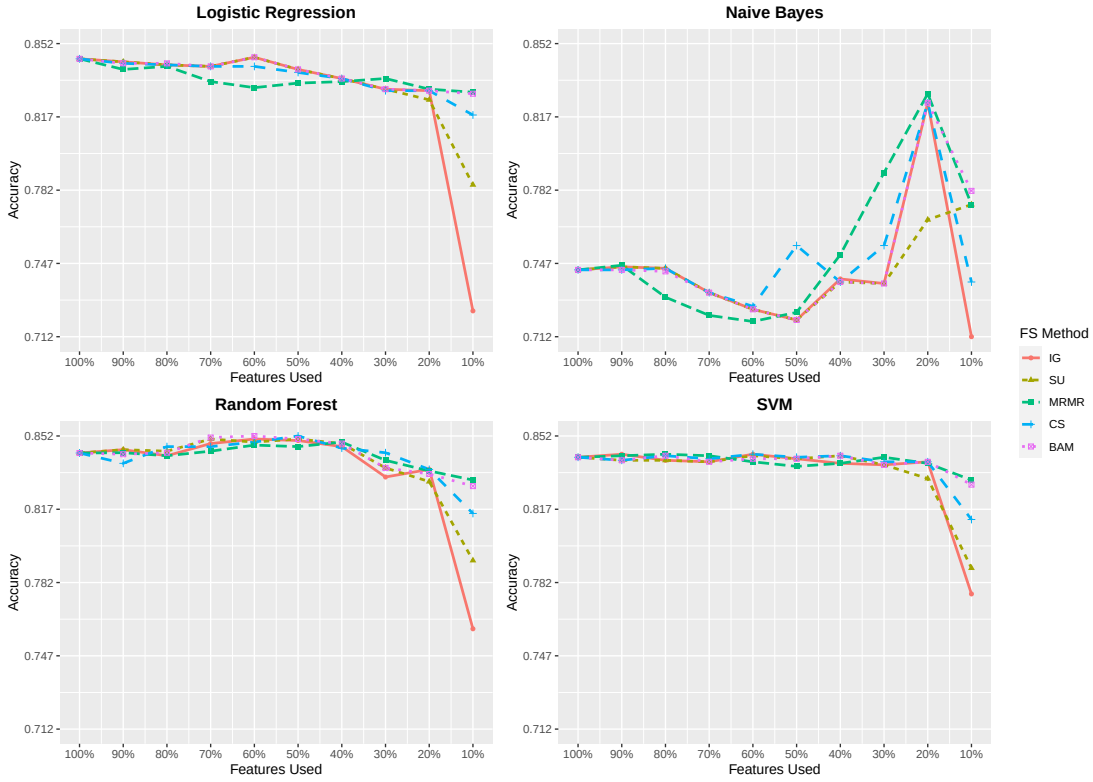

Figure 4.9. Ada dataset classification results (by FS)

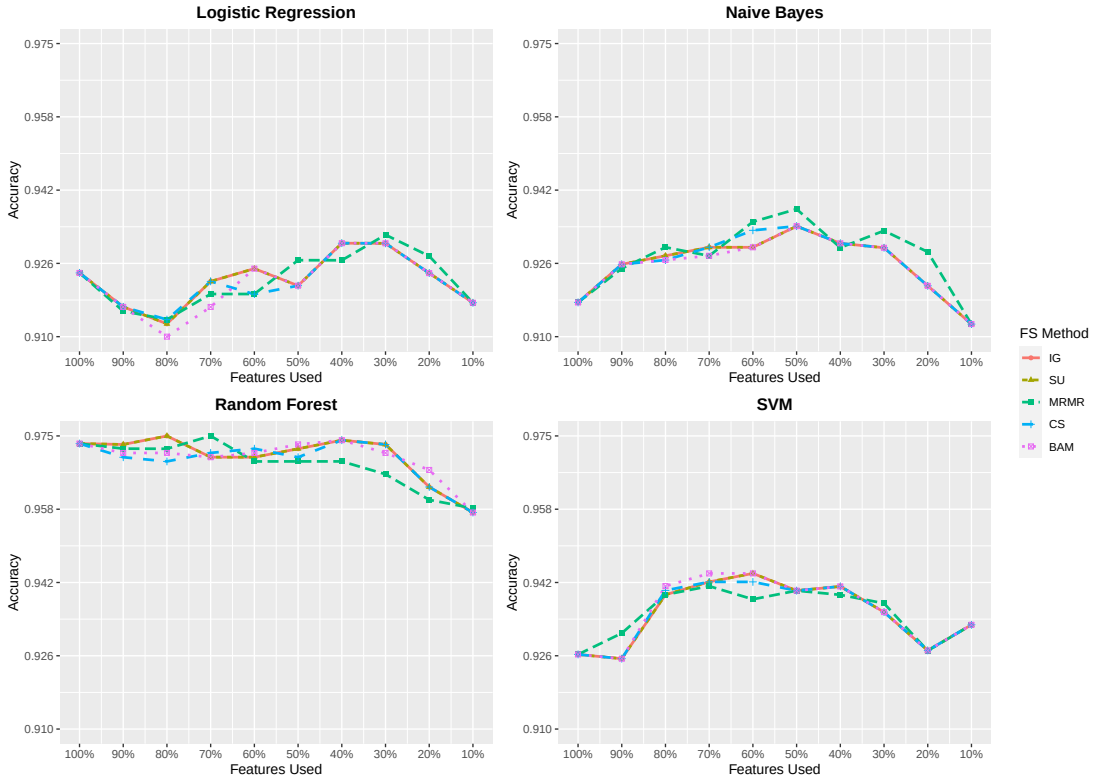

Figure 4.10. Splice dataset classification results (by FS)

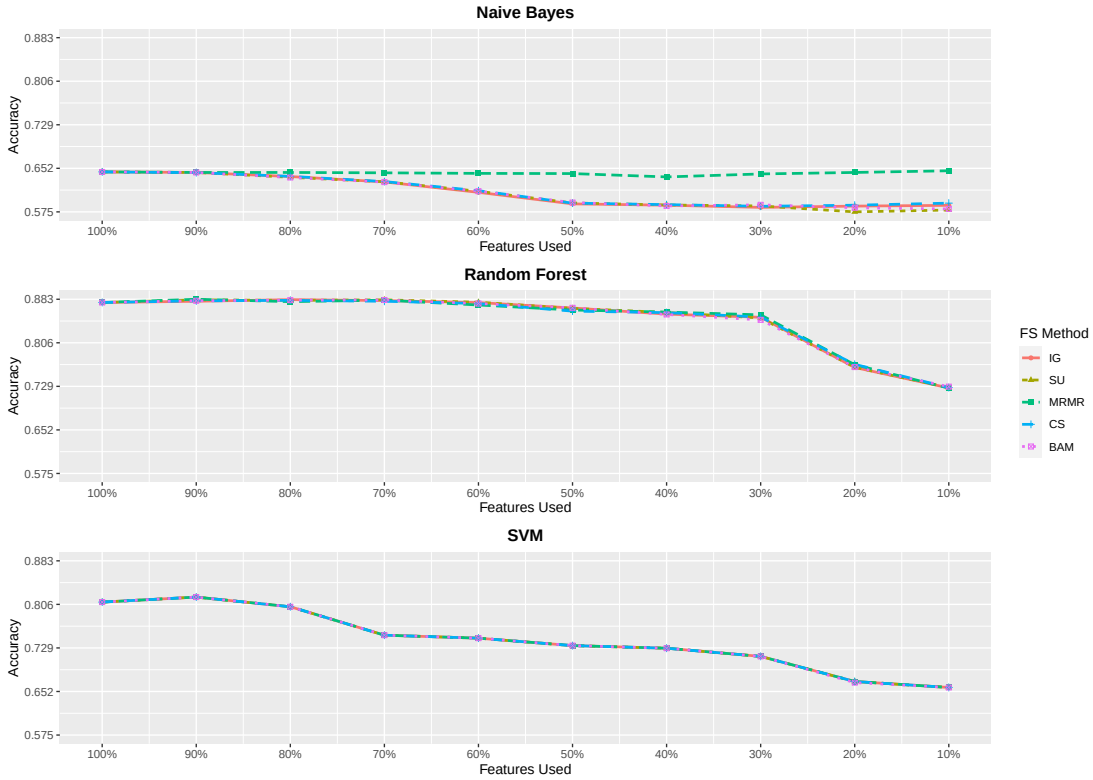

Figure 4.11. Indian Pines dataset classification results (by FS)

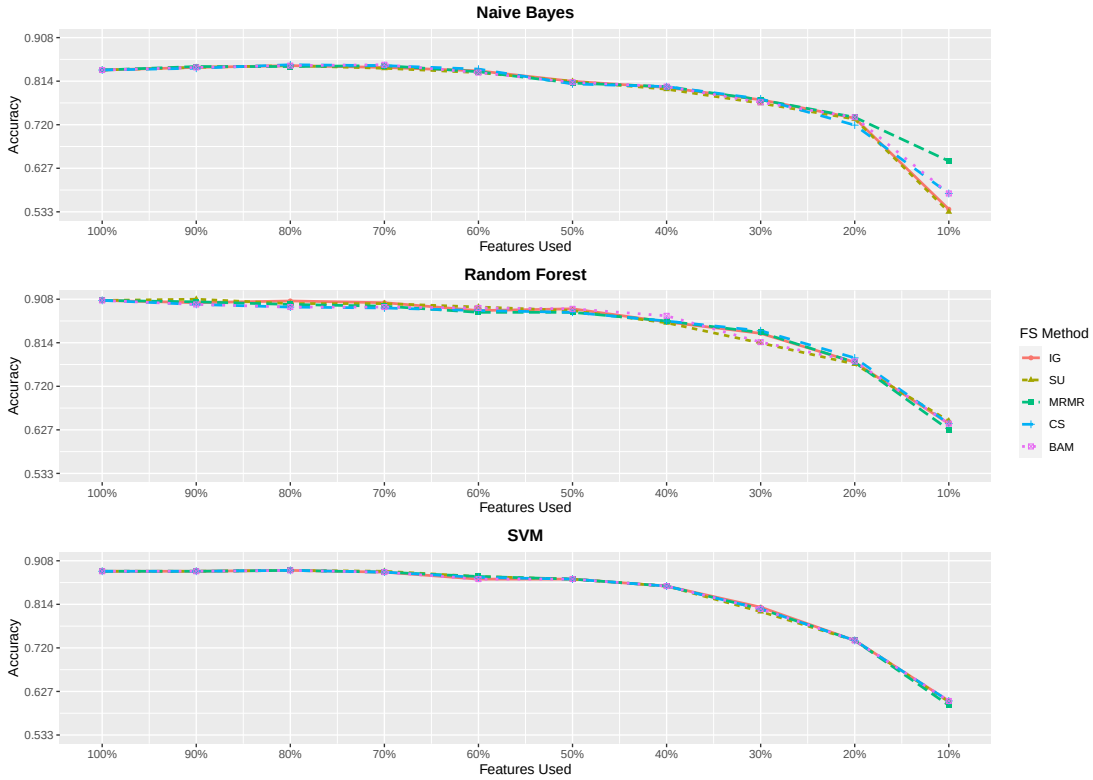

Figure 4.12. Semeion dataset classification results (by FS)

Table 4.3. The Weighted Average Accuracy (*WAcc*) across WAM and BAM frameworks (by FS)

Dataset	Classifier	Information Gain	Symmetric Uncertainty	MRMR	Chi-Squared	BAM
Jasmine	Logistic Regression	0.777737094	0.774542663	0.776988214	**0.779748205**	0.777648458
	Random Forest	0.811066766	0.811714121	0.80630254	0.810135811	**0.811940928**
	Naive Bayes	0.761989954	**0.762901166**	0.762668092	0.761784717	0.761161243
	SVM	0.721481573	0.736496728	0.746852638	**0.758842094**	0.732229371
Image	Logistic Regression	0.728227035	**0.735411663**	0.714032594	0.729149491	0.726829013
	Random Forest	0.773557276	0.773478472	**0.775014402**	0.769885659	0.771758501
	Naive Bayes	0.670619334	0.669195377	**0.688176411**	0.678267809	0.676746593
	SVM	0.859287772	0.855166274	**0.859563336**	0.856717166	0.853891445
Scene	Logistic Regression	0.818903899	0.818760524	**0.829500518**	0.822860766	0.822853002
	Random Forest	0.851768634	0.852691166	0.855387509	0.85806332	**0.858111801**
	Naive Bayes	0.704603002	0.69504158	0.698178571	**0.714646135**	0.709193237
	SVM	0.856276052	0.854928054	0.862983782	**0.864100242**	0.859195307
Musk	Logistic Regression	0.91559596	0.915343434	0.893161616	0.90689899	**0.916363636**
	Random Forest	**0.951343434**	0.949040404	0.949212121	0.94420202	0.950060606
	Naive Bayes	0.825888889	0.825151515	0.743707071	0.798545455	**0.829878788**
	SVM	**0.931093862**	0.926793146	0.920002016	0.920226368	0.92916029
Philippine	Logistic Regression	0.714459609	0.713131594	0.700473179	0.715055551	**0.715490537**
	Random Forest	0.747480838	0.74632694	0.730154919	**0.748725489**	0.746655889
	Naive Bayes	0.687216024	0.688416034	0.657540349	**0.691519919**	0.687125299
	SVM	0.722021573	0.722879148	0.705169215	**0.724411022**	0.72341917
Ionosphere	Logistic Regression	0.80821256	0.810958132	**0.820933977**	0.807689211	0.817504026
	Random Forest	**0.901602254**	0.891570048	0.856441224	0.859814815	0.862979066
	Naive Bayes	0.854951691	0.857705314	0.854830918	0.85242351	**0.859565217**
	SVM	**0.916215781**	0.897342995	0.900241546	0.899557166	0.897230274
Optdigits	Logistic Regression	0.975308568	0.975356102	**0.978297358**	0.975141945	0.975782547
	Random Forest	0.977264345	**0.978414957**	0.978034304	0.977905106	0.977229265
	Naive Bayes	0.94422791	0.937242828	**0.960119945**	0.948317085	0.944156514
	SVM	0.979991127	0.979183521	0.981020337	**0.981139743**	0.979290093
Satelite	Logistic Regression	0.994096015	0.992460867	**0.995365303**	0.994461912	0.994827772
	Random Forest	**0.995114628**	0.993141957	0.99490644	0.994748691	0.994892367
	Naive Bayes	0.946982387	0.969562201	0.961157292	**0.958576197**	0.962850611
	SVM	**0.99315772**	0.99112903	0.992673175	0.992869215	0.992920618
Ada	Logistic Regression	0.81341879	0.824654889	0.832365173	0.831398873	**0.834113954**
	Random Forest	0.82568438	0.832223385	**0.840397926**	0.838436085	0.840169751
	Naive Bayes	0.745233146	0.747732666	**0.766919781**	0.757098078	0.758832325
	SVM	0.827156502	0.828557643	**0.838015487**	0.83515949	0.837888871
Splice	Logistic Regression	0.923429926	0.923429926	**0.923851956**	0.922970278	0.922926977
	Random Forest	0.967699406	0.967699406	0.965505202	0.967408844	**0.968059719**
	Naive Bayes	0.925090421	0.925090421	**0.927738592**	0.925381963	0.924922902
	SVM	0.936007372	0.936007372	0.935605481	0.935882907	**0.936216804**
Indian Pines	Random Forest	0.818901759	0.819352962	**0.820038851**	0.81952797	0.818609529
	Naive Bayes	0.594505273	0.591830337	**0.643413263**	0.596480731	0.59360754
	SVM	**0.710693582**	0.710693582	0.710693582	0.710693582	0.710343855
Semeion	Random Forest	0.806870593	0.803915937	0.802848638	0.807401078	**0.805217426**
	Naive Bayes	0.742873362	0.739290115	**0.763815853**	0.747545666	0.749211096
	SVM	**0.7832915**	0.782098007	0.781322652	0.78303674	0.782702863

4.3.2 Comparison of the classification performance for aggregation techniques

Similar to the WAM and BAM findings of section 4.3.1 in which only the Arithmetic Mean aggregation was used in both ensembles; we note that under most aggregation procedures, the WAM still enhances the baseline ($k = 1.0$) classification performance. Moreover, the patterns depicted are consistent with those discussed in section 4.3.1. The Naive Bayes, for instance, highlights the effectiveness of the WAM framework under most aggregated feature selection methods. To illustrate, the MRMR accuracy rises from around 0.68 baseline accuracy to nearly 0.81 in the Image dataset in Figure 4.14 and from baseline 0.89 to nearly 0.99 in the Satellite dataset in Figure 4.20. A similar increase in the performance of the aggregated Chi-Squared method can be observed in the Image and Ionosphere datasets in Figure 4.14 and Figure 4.18. Moreover, none of the utilized feature selection techniques under WAM is consistently the best performing under most datasets, even when the same aggregation procedure is considered. For instance, in the Jasmine and Indian Pines datasets in Figure 4.13 and Figure 4.23, MRMR produces the best accuracy curves; yet it is also the worst performing feature selection method under most aggregations in the Scene, Musk, and Philipine datasets in Figures 4.15, Figure 4.16, and Figure 4.17. Generally speaking, the overall trend across different aggregation procedures depicts a stable or improved accuracy performance followed by a sharp decrease in the overall accuracy once the number of used features falls below a certain threshold. Furthermore, this pattern is still data-dependent, with the ideal feature reduction threshold varied between datasets, such as 30% features retained in the Ionosphere, Ada, and Splice datasets in Figure 4.18, Figure 4.21, and Figure 4.22, respectively; or 20% features retained in the Jasmine and Musk datasets in Figure 4.13 and Figure 4.16.

Similarly, the classification performance can be considered data-dependent for each of the aggregation approaches used. While the GM aggregation procedure performs well in the Scene and Musk datasets in Figures 4.15-4.16, it is one of the worst aggregation frameworks in the Image, Ionosphere, and Philippine datasets in Figures 4.14, Figure 4.18, and Figure 4.17. Likewise, Stuart rank aggregation produces the best accuracy scores in the Jasmine, Indian Pines, and Splice datasets in Figure 4.13, Figure 4.23, and Figure 4.22, but is beaten by practically all other aggregations in the Musk

and Semeion datasets in Figure 4.16 and Figure 4.24. The nature of the feature selection techniques studied also contributes to these disparities. That is, under Information Gain, we see more pronounced differences in classification performance between the various aggregations; whereas Chi-Squared exhibits the most overlap. In fact, under both the Information Gain and MRMR methods, a repeated-measures ANOVA demonstrates significant differences in the classification accuracies between the different aggregations (Bonferroni-adjusted p-values 0.0068 and 0.0304 respectively). On further investigation, the significant pairwise differences were often linked to a difference between one of the score-based aggregations (AM, GM, L2) and the rank-based aggregations (Stuart, RRA). For instance, the rank-based RRA and all of the score-based aggregations (AM, GM, L2) were found to have significant accuracy differences under MRMR (adjusted p-values 0.001, 0.005 and 0.002, respectively). In addition, significant differences between the AM and GM aggregations were also discovered. However, there were no significant differences between the AM and L2 aggregations.

Overall, while the performance of the aggregation procedure remains data-depenedent, we find that the score-based Arithmetic Mean and L2 Norm aggregation procedures perform well across most experiments. In fact, in 11 of 12 datasets, both aggregation approaches consistently perform in the middle or outperform the other aggregations. Furthermore, both techniques have more consistent accuracy curves that are less volatile across datasets and feature selection methods. When compared to the L2 Norm, we find that the Arithmetic Mean is marginally more consistent, especially in the Ionosphere and Satellite datasets in Figure 4.18 and Figure 4.20. That is to say, while both aggregations procedures may be used as simple and efficient strategies for achieving good accuracy behavior under the WAM framework; the score-based Arithmetic Mean is a particularly good choice due to its simplicity and ease of implementation, as well as its demonstrated good performance in previous studies [51], [66].

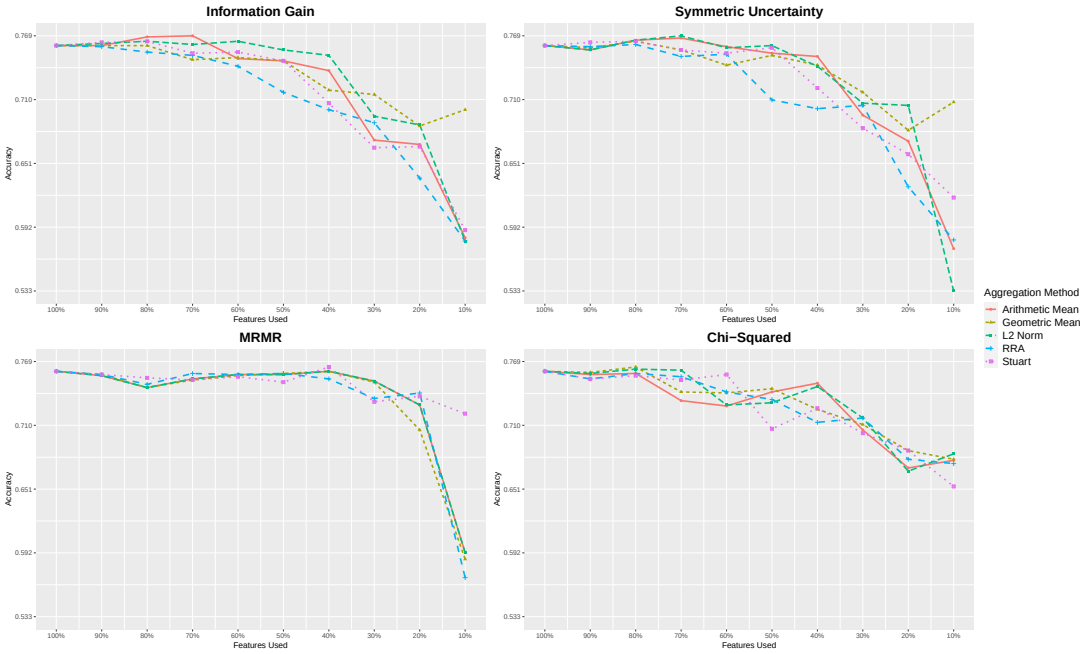

Figure 4.13. Jasmine dataset classification results (by AT)

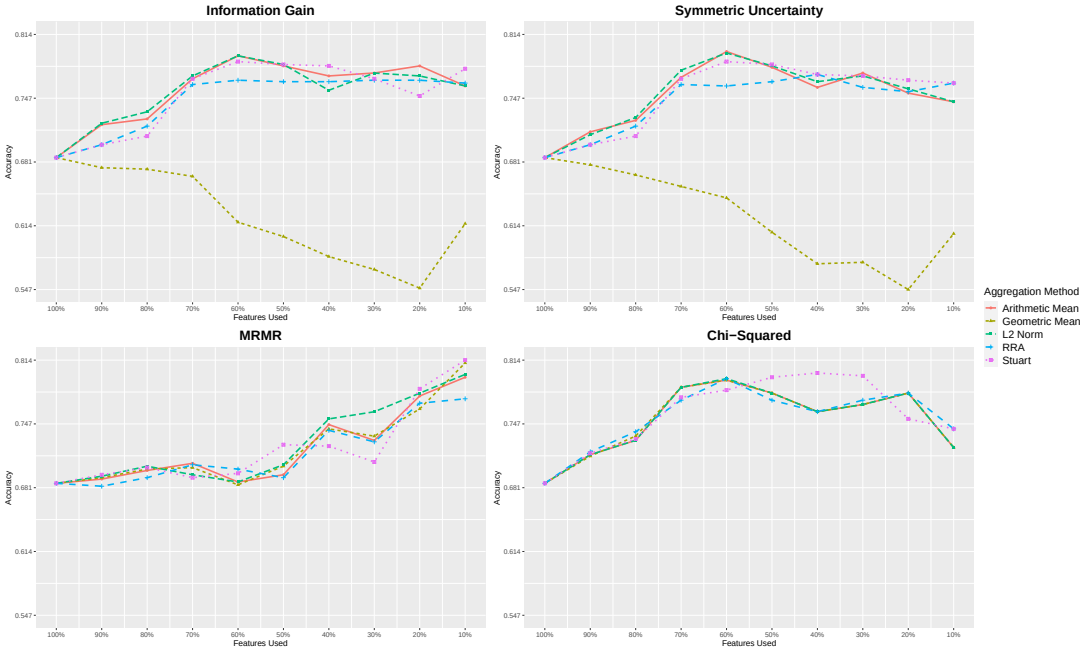

Figure 4.14. Image dataset classification results (by AT)

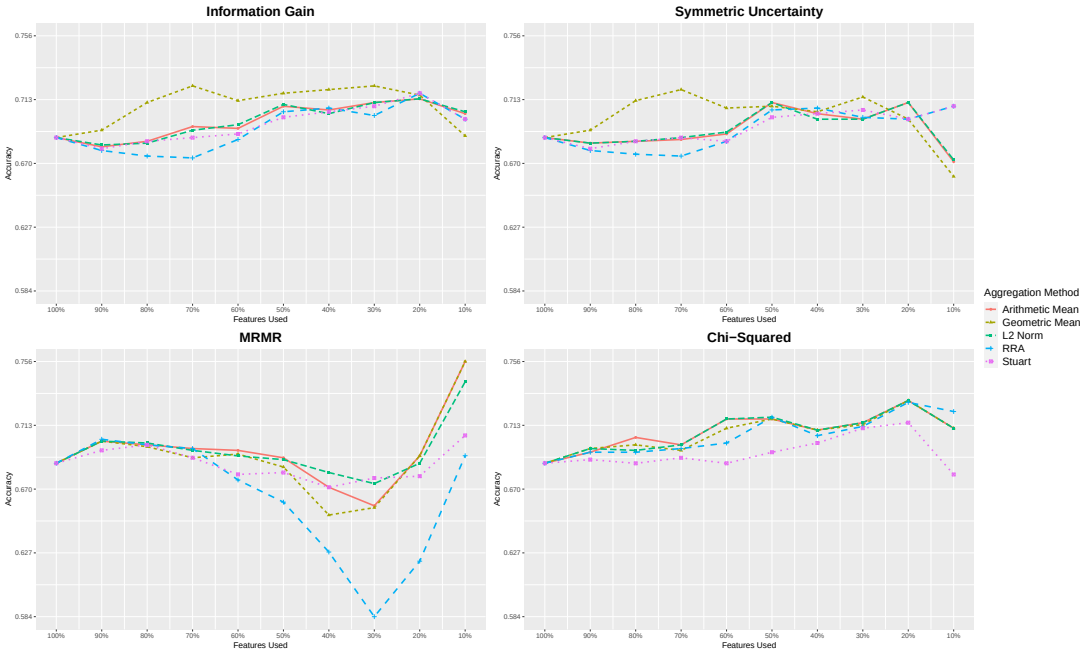

Figure 4.15. Scene dataset classification results (by AT)

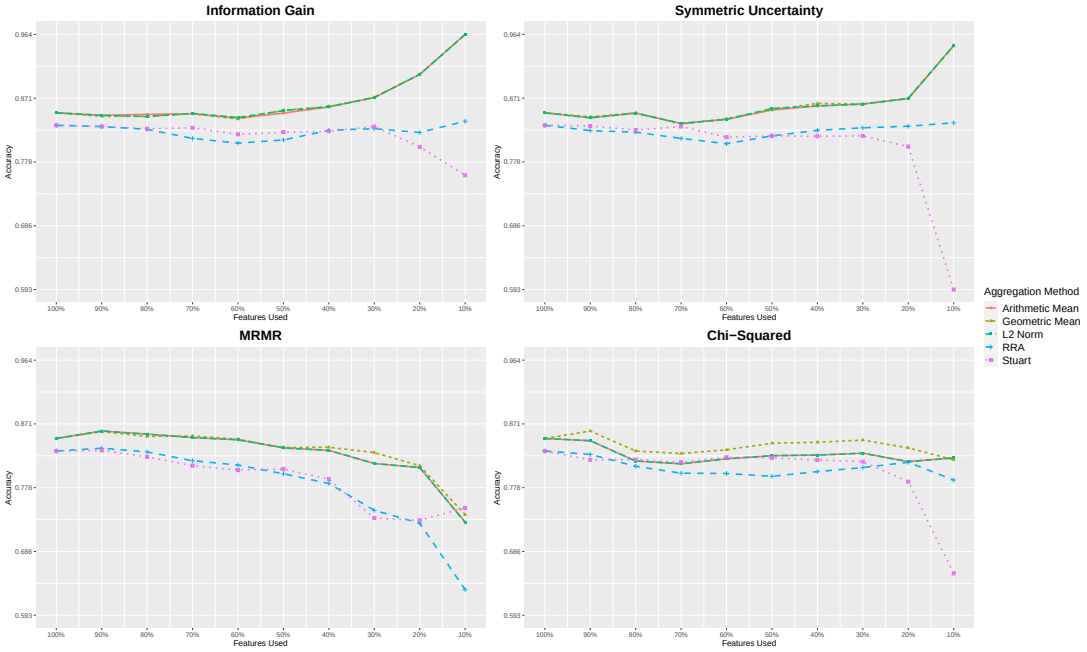

Figure 4.16. Musk dataset classification results (by AT)

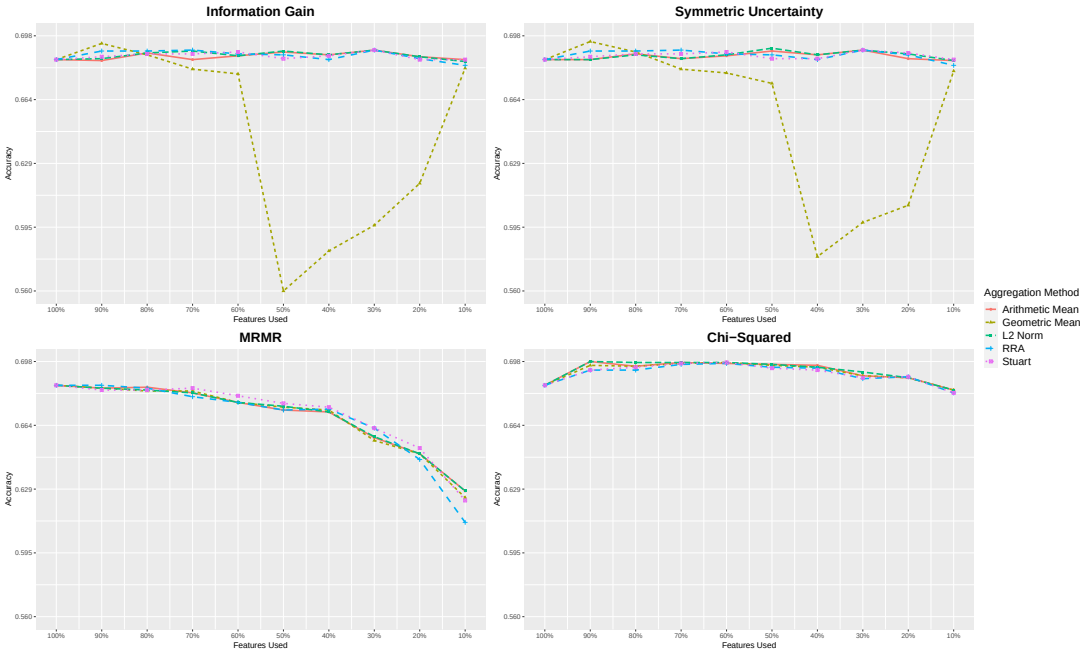

Figure 4.17. Philippine dataset classification results (by AT)

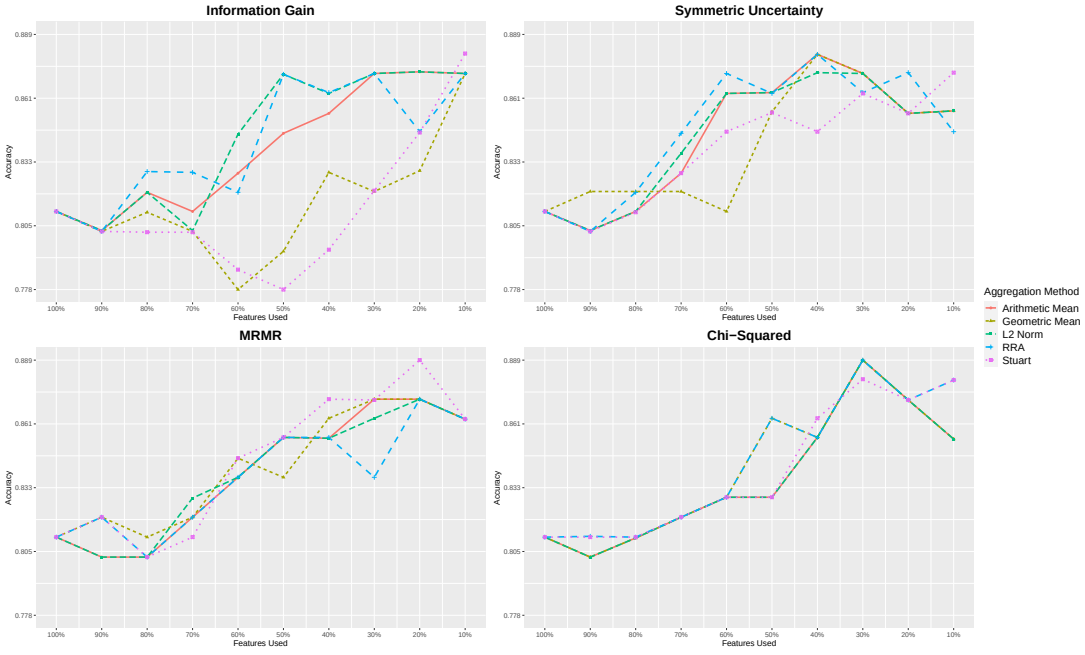

Figure 4.18. Ionosphere dataset classification results (by AT)

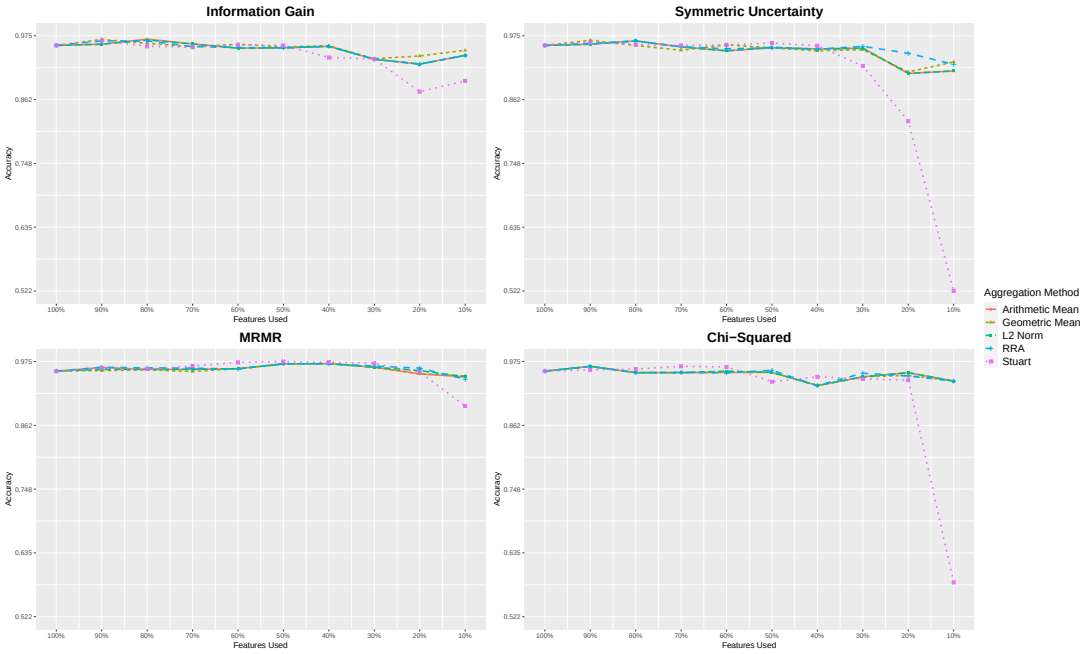

Figure 4.19. Optdigits dataset classification results (by AT)

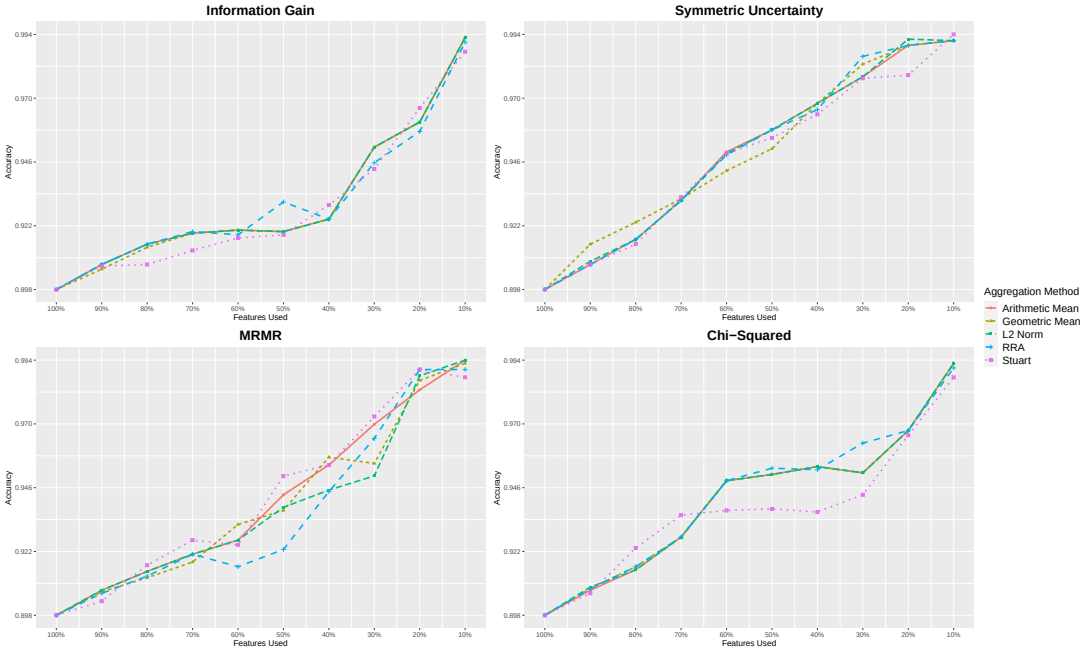

Figure 4.20. Satellite dataset classification results (by AT)

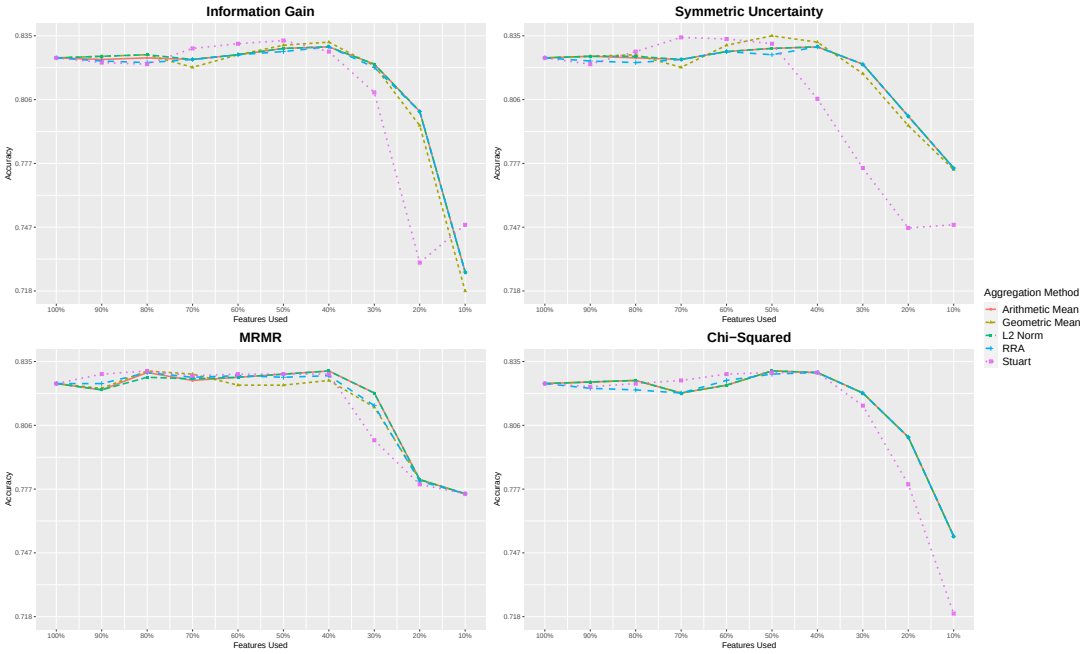

Figure 4.21. Ada dataset classification results (by AT)

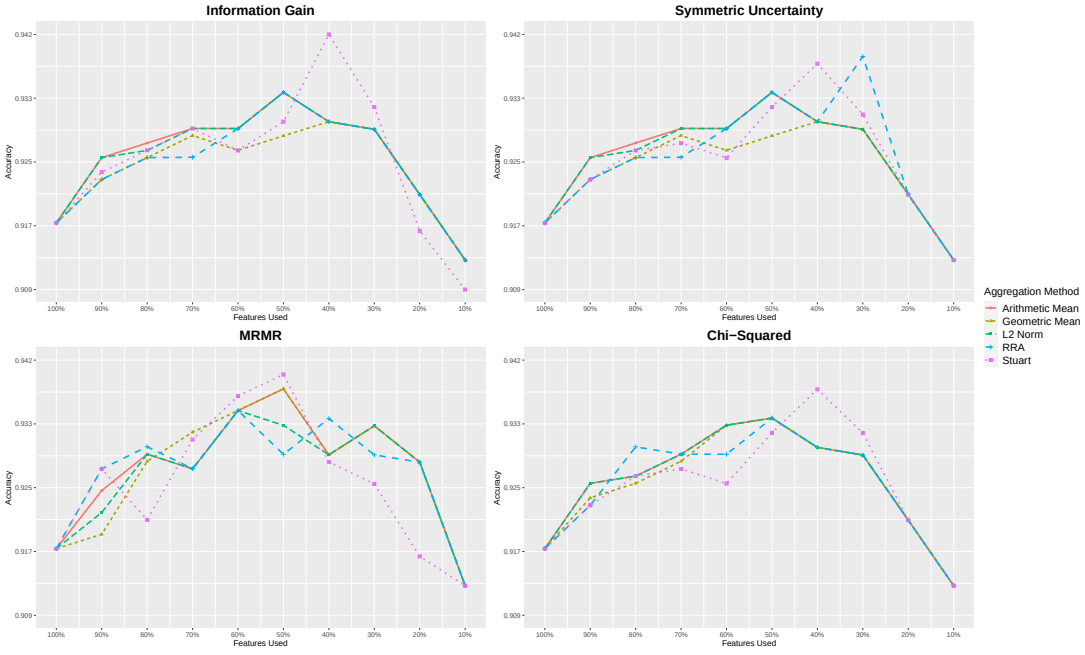

Figure 4.22. Splice dataset classification results (by AT)

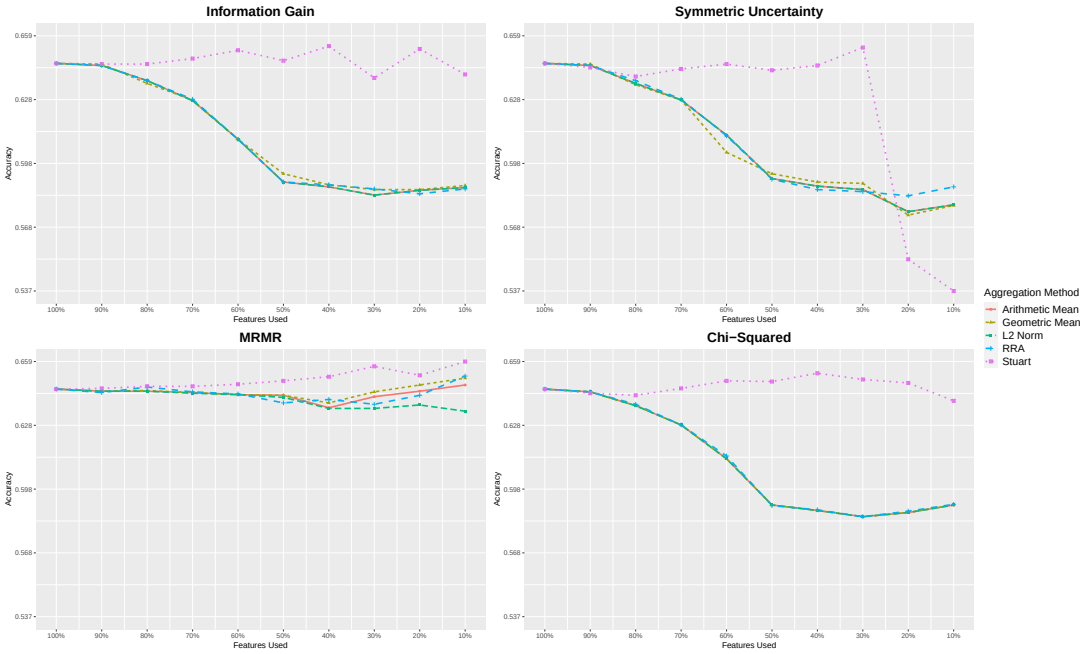

Figure 4.23. Indian Pines dataset classification results (by AT)

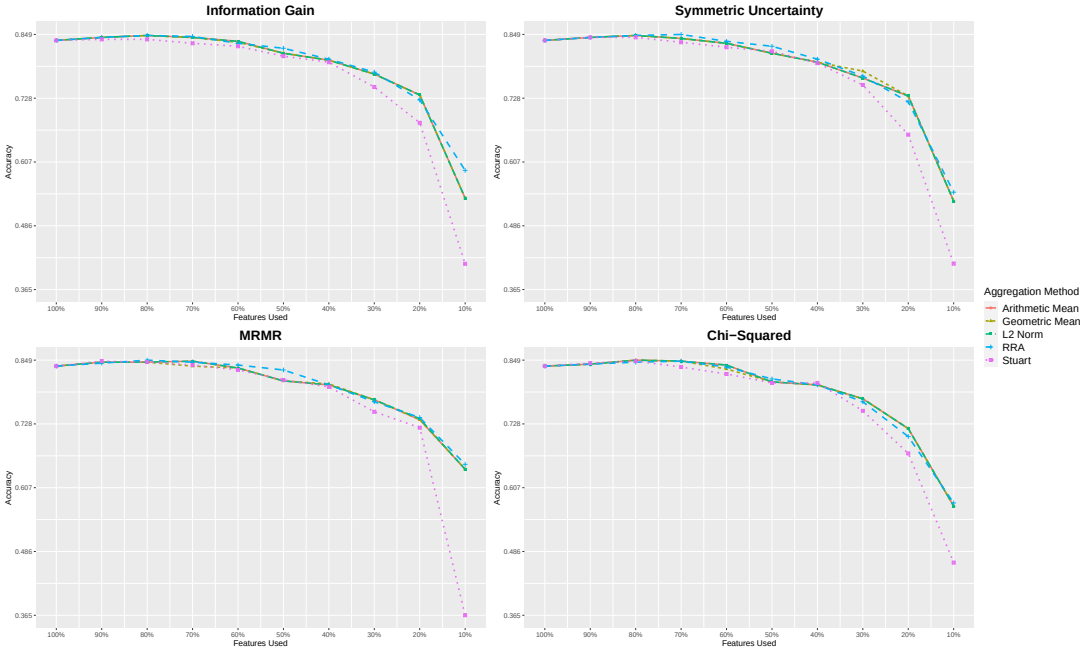

Figure 4.24. Semeion dataset classification results (by AT)

4.4 Stability Performance Results

The findings of the stability analysis of the four aggregated filters using WAM and BAM are presented in Table 4.4. The importance scores obtained from applying the feature selection method FS on every bootstrap sample are used to compute the stability scores of the average Pearson's Correlation Coefficient. In contrast, average Spearman's Rank Correlation Coefficient and Canberra's Distance are calculated using the ranks derived from the sorted importance scores. The Jaccard's Index is also calculated using feature subsets of the 25% topmost ranking features (represented by index vectors). Lastly, the Average Standard Deviation (ASD) is computed using the normalized importance scores averaged over the 1000 bootstraps. The bolded values in Table 4.4 reflect the best stability value for each dataset. This is the highest stability score in Jaccard's Index, Spearman's Rank Correlation Coefficient, and Pearson's Correlation Coefficient, but the lowest value in Canberra's Distance and ASD.

Likewise, the findings of the stability analysis of the five aggregation procedures under WAM are presented in Tables 4.5-4.6. The feature ranks obtained from embedding the WAM within a 100-fold cross-validation procedure and aggregating using each of the discussed aggregation procedures are used to compute the stability scores of the average Spearman's Rank Correlation Coefficient and Canberra's Distance. The Jaccard's Index is also calculated using feature subsets of the 25% topmost ranking features (represented by index vectors) across each of the aggregation strategies. The bolded values in Tables 4.4-4.6 reflect the best stability value for each dataset. This is the highest stability score in Jaccard's Index and Spearman's Rank Correlation Coefficient, but the lowest value in Canberra's Distance. Note that Table 4.5 represents the stability results using Information Gain, whereas Table 4.6 depicts the stability results using MRMR.

In the following two subsections, we analyze and compare the performance of the WAM and BAM algorithms, in terms of the stability behavior of their feature selection processes. We also analyze and compare the stability influence of the different aggregation techniques within the WAM framework.

4.4.1 Comparison of the stability for feature selection methods

Similar to the classification performance results, there is no single feature selection method that consistently produces optimal stability behavior for every experimental dataset. In other words, the stability of the feature selection approaches is data dependant, though there are some observable patterns. For example, in the Philippine and Ada datasets, IG is the most stable approach across all stability metrics. However, in the Ionosphere dataset, it is MRMR that is the most stable across all similarity measures. Likewise, Chi-Squared achieves the best stability scores in the Scene and Optidgits datasets, whereas Symmetric Uncertainty is more robust in the Splice and Semeion datasets. Therefore, none of the tested feature selection approaches can be deemed the most stable overall. However, across most experiments, IG and MRMR exhibit relatively good stability behavior under several metrics.

While none of the feature selection methods is consistently the most stable in every measure, we observe that the Pearson, Spearman, and Canberra-based stability scores are high for each of the filters. Given the low average standard deviation scores, there also seems to be a lack of variation across the features importance scores obtained from every bootstrap sample, which translates to similar rankings and high feature selection stability. On the other hand, the comparatively smaller values of the average Jaccard's index suggest the features subsets constructed using the topmost ranked features are less stable. This disparity between the rank and index-based stability measures can be explained by larger inconsistencies across the higher feature ranks than the lower ones. In the previous discussion, we observed that for most datasets, the accuracy scores drop steadily once the threshold for selecting a subset of features falls below a certain threshold. In other words, while the feature selection succeeds in removing the most irrelevent features with relative confidence, it is difficult to single out the most significant features accurately. Alternatively, it is possible that below the indicated threshold, there are no irrelevant features to discard in the first place, which indicates the goal of feature selection has been effectually achieved.

Conversely, when comparing the stability behavior of the feature selection processes under the WAM and BAM, Table 4.4 suggests that when aggregated using WAM, at least one of the singular feature selection techniques achieves better stability than the BAM.

Note that, with the exception of the Image dataset, each of the similarity measures in Table 4.4 is generally dominated by one of the individual feature selection methods. In contrast, the stability under BAM is consistently placed in the middle of the other feature selection approaches. This is similar to the observations noted in section 4.3.1 regarding the classification performance of the BAM. While BAM reveals a comparable stability behavior to that of each single feature selection method under WAM; it appears that in most cases, the use of an individual feature selection method is better for maintaining stability of the results. Interestingly, there also seems to be some indication of a positive association between the stability behavior of the individual feature selection methods and their classification performance under the WAM. For instance, the Chi-Squared feature selection method dominant in terms of stability in the Scene dataset is also the best performing method in Scene (Figure 4.3). A similar pattern can be seen with respect to the higher stability of Information Gain in the Ada dataset (Figure 4.9) or the Weighted Average Accuracy of Chi-Squared in Satellite. While it is possible that the feature selection method that outperforms in terms of classification accuracy may also outperform in terms of stability behavior, this relationship is likely dependent on both the dataset composition and the similarity measures utilized in the analysis of the stability.

4.4.2 Comparison of the stability for aggregation techniques

Finally, in terms of the stability behavior of the WAM ensemble under different aggregation procedures, we observe more representative patterns. That is, in comparison to the rank-based aggregation methods (Stuart, RRA), the score-based aggregation methods (AM, GM, L2) produce better stability scores across both Information Gain and MRMR. Furthermore, across all experimental datasets and for all implemented stability criteria, the rank-based Stuart aggregation provides the lowest stability results. These patterns are observable under both binary and multiclass datasets. The findings of this analysis emphasize the importance of recognizing the differences in stability influence between score-based and rank-based aggregations in the construction of the ensemble feature selection framework. Based on the results of these experiments, we remark that using a score-based aggregation procedure in the construction of the ensemble appears to be a suitable alternative in many cases, especially since the scores have a stronger

scale and provide a higher level of detail about the importance of the features. Moreover, under MRMR, we particularly note that the Arithmetic Mean and L2 Norm generally produce the highest stability scores, peaking at multiple datasets (e.g. Jasmine, Optdigits, Indian Pines) in comparison to the other tested aggregation procedures.

Statistically, one-way ANOVA reveals that the observed stability differences between the five aggregation techniques are significant (Bonferroni-adjusted p-values $<$ 0.0001). Post-hoc analysis demonstrates that these significant differences can nearly always be attributable to Stuart aggregation when compared to any other aggregation technique (Bonferroni-adjusted p-values $<$ 0.0001). Other significant differences are also discovered under Information Gain using Canberra's Distance between the Arithmetic Mean and L2 Norm (Bonferroni-adjusted p-value $=$ 0.037), Arithmetic Mean and RRA (Bonferroni-adjusted p-value $=$ 0.041), and using Jaccard's Index between the Arithmetic Mean and RRA (Bonferroni-adjusted p-value $=$ 0.031). Overall, the Arithmetic Mean and L2 Norm appear to outperform alternative aggregation rules in terms of stability, whereas RRA outperforms Stuart in terms of rank-based aggregation. In fact, the Arithmetic Mean and L2 Norm seem to exhibit similar influences on the ensemble's stability as well as its classification performance. According to the experimental analysis of this thesis, these two aggregation functions present a preferable choice in terms of performance and efficiency when compared to more sophisticated options in most cases.

Note that also, when comparing the IG and MRMR columns in Table 4.4 with the results obtained in Tables 4.5 and 4.6, respectively; the findings demonstrate noticeable improvement in the shared rank-based and index-based similarity measures. In other words, the experimental stability analysis demonstrates the suitability of the proposed ensemble, since it matched or improved on the stability results achieved by the individual feature selection methods. Moreover, based on the findings in Tables 4.5-4.6, it becomes possible to recommend the score-based aggregation techniques for similar experimental designs.

Table 4.4. Stability analysis results across all datasets (by FS)

Dataset	Stability Measure	Information Gain	Symmetric Uncertainty	MRMR	Chi-Squared	BAM
Jasmine	Average Pearson Correlation	0.299705	0.258270	**0.902748**	0.360289	0.335806
	Average Spearman Rank Correlation	0.319009	0.378017	**0.730414**	0.231101	0.334655
	Average Jaccard's Index	0.254683	**0.319411**	0.308712	0.286586	0.252964
	Average Canberra Distance	0.278113	0.269500	**0.124180**	0.337122	0.294955
	Average Standard Deviation	0.744817	0.753504	**0.149429**	0.747054	0.765530
Image	Average Pearson Correlation	0.768404	0.760257	0.461867	0.784970	**0.793634**
	Average Spearman Rank Correlation	0.702970	0.690212	0.541442	**0.716971**	0.671209
	Average Jaccard's Index	0.534739	0.514374	0.275411	0.557456	**0.560470**
	Average Canberra Distance	**0.140367**	0.142500	0.236440	0.242640	0.254949
	Average Standard Deviation	0.437907	0.462153	0.643365	0.459268	**0.434591**
Scene	Average Pearson Correlation	0.898425	0.887953	0.652895	**0.933014**	0.908673
	Average Spearman Rank Correlation	0.871633	0.863620	0.705409	**0.904599**	0.881263
	Average Jaccard's Index	0.725580	0.718157	0.429004	**0.834032**	0.761022
	Average Canberra Distance	**0.150622**	0.156575	0.206240	0.169175	0.182344
	Average Standard Deviation	0.309285	0.328335	0.501868	**0.253048**	0.296812
Musk	Average Pearson Correlation	0.953028	0.939819	**0.983622**	0.972910	0.971086
	Average Spearman Rank Correlation	0.897172	0.920754	**0.978164**	0.958189	0.932817
	Average Jaccard's Index	0.254683	**0.319411**	0.308712	0.286586	0.252964
	Average Canberra Distance	0.278113	0.269500	**0.124180**	0.337122	0.294955
	Average Standard Deviation	0.198588	0.230549	**0.106096**	0.153881	0.164122
Philippine	Average Pearson Correlation	**0.992381**	0.987185	0.949337	0.974312	0.990140
	Average Spearman Rank Correlation	**0.948322**	0.945942	0.876291	0.794429	0.826292
	Average Jaccard's Index	**0.907578**	0.895855	0.599476	0.882073	0.898057
	Average Canberra Distance	**0.036655**	0.037883	0.123565	0.216403	0.199559
	Average Standard Deviation	**0.065557**	0.093865	0.194033	0.133756	0.090117
Ionosphere	Average Pearson Correlation	0.398351	0.583203	**0.803445**	0.689003	0.678480
	Average Spearman Rank Correlation	0.391580	0.583566	**0.779300**	0.634363	0.621247
	Average Jaccard's Index	0.322984	0.418490	**0.588096**	0.511871	0.514258
	Average Canberra Distance	0.284348	0.254220	**0.185660**	0.245600	0.249635
	Average Standard Deviation	0.731482	0.606503	**0.397275**	0.546923	0.549206
Optdigits	Average Pearson Correlation	0.974733	0.956047	0.946192	**0.978112**	0.976264
	Average Spearman Rank Correlation	0.965357	0.959320	0.913443	**0.968890**	0.967125
	Average Jaccard's Index	**0.776935**	0.687190	0.621800	0.699440	0.740367
	Average Canberra Distance	0.087271	0.094572	0.112731	**0.077847**	0.090535
	Average Standard Deviation	0.150498	0.196308	0.188786	**0.141531**	0.146916
Satellite	Average Pearson Correlation	0.962102	0.735536	0.735555	**0.962324**	0.932846
	Average Spearman Rank Correlation	0.913703	0.737037	0.886279	**0.941141**	0.912465
	Average Jaccard's Index	**0.889171**	0.523733	0.579217	0.711644	0.540189
	Average Canberra Distance	0.128159	0.206656	**0.093391**	0.117366	0.126052
	Average Standard Deviation	0.189240	0.429693	0.289669	**0.186215**	0.235742
Ada	Average Pearson Correlation	**0.998732**	0.998655	0.997906	0.992348	0.998004
	Average Spearman Rank Correlation	**0.956392**	0.952797	0.823995	0.955461	0.952162
	Average Jaccard's Index	**0.919947**	0.866222	0.607214	0.830739	0.863409
	Average Canberra Distance	**0.106299**	0.108989	0.155215	0.122942	0.125797
	Average Standard Deviation	**0.028835**	0.031522	0.029137	0.083938	0.042076
Splice	Average Pearson Correlation	0.992299	**0.993156**	0.990926	0.974606	0.989386
	Average Spearman Rank Correlation	0.841882	0.842385	0.738889	**0.843115**	0.836391
	Average Jaccard's Index	0.761442	**0.762814**	0.597770	0.760529	0.742453
	Average Canberra Distance	0.187747	0.187556	0.224846	**0.187334**	0.190992
	Average Standard Deviation	0.070557	**0.065447**	0.081051	0.157357	0.096031
Indian Pines	Average Pearson Correlation	**0.999238**	0.988947	0.742512	0.996915	0.992818
	Average Spearman Rank Correlation	**0.997207**	0.983358	0.900745	0.993381	0.990461
	Average Jaccard's Index	0.90582	0.849971	0.613475	0.868253	**0.937861**
	Average Canberra's Distance	**0.024178**	0.039431	0.116739	0.036038	0.038302
	Average Standard Deviation	**0.026581**	0.085295	0.305766	0.050679	0.06472
Semeion	Average Pearson Correlation	0.958882	**0.959117**	0.941782	0.956113	0.956536
	Average Spearman Rank Correlation	0.944449	0.94398	0.937229	**0.944859**	0.943758
	Average Jaccard's Index	0.728269	0.732672	0.696779	**0.764083**	0.731358
	Average Canberra's Distance	0.116476	**0.116419**	0.121602	0.119251	0.118144
	Average Standard Deviation	0.198025	**0.197594**	0.228502	0.207605	0.206073

Table 4.5. Stability analysis results using Information Gain (by AT)

Dataset	Stability Measure	Arithmetic Mean	Geometric Mean	L2 Norm	Stuart	RRA
Jasmine	Average Spearman Rank Correlation	**0.984421**	0.978226	0.98267	0.088026	0.92984
	Average Jaccard's Index	0.996506	**1.00000**	0.996506	0.306569	0.961509
	Average Canberra Distance	**0.020466**	0.026401	0.022534	0.266369	0.04919
Image	Average Spearman Rank Correlation	**0.992861**	**0.966235**	0.992041	0.106854	0.980959
	Average Jaccard's Index	0.903514	**0.929316**	0.899481	0.357117	0.890638
	Average Canberra Distance	**0.027475**	0.070757	0.028673	0.253253	0.044732
Scene	Average Spearman Rank Correlation	**0.996959**	0.974216	0.996281	0.04031	0.984125
	Average Jaccard's Index	0.960134	0.879713	**0.961829**	0.350817	0.886225
	Average Canberra Distance	**0.023078**	0.055415	0.024018	0.260263	0.045151
Musk	Average Spearman Rank Correlation	0.998464	0.998376	**0.998497**	0.125679	0.992484
	Average Jaccard's Index	0.938857	**0.944736**	0.934836	0.326162	0.927091
	Average Canberra Distance	0.014975	0.014985	**0.014913**	0.255385	0.03036
Philippine	Average Spearman Rank Correlation	0.949456	0.947388	0.944734	0.257199	**0.999066**
	Average Jaccard's Index	**0.984514**	0.573192	0.984349	0.374306	0.969606
	Average Canberra Distance	0.037079	0.126254	0.039078	0.217622	**0.011732**
Ionosphere	Average Spearman Rank Correlation	0.983104	0.969398	**0.984172**	0.054752	0.913307
	Average Jaccard's Index	**0.774483**	0.724724	0.770639	0.312562	0.770212
	Average Canberra Distance	**0.045669**	0.059043	0.046972	0.274561	0.079306
Optdigits	Average Spearman Rank Correlation	0.998708	0.991615	**0.998774**	0.26705	0.994696
	Average Jaccard's Index	**0.996923**	0.909464	0.996923	0.511447	0.937676
	Average Canberra Distance	0.009819	0.022968	**0.009697**	0.213391	0.020513
Ada	Average Spearman Rank Correlation	0.99697	**0.997174**	0.996392	0.212377	0.986638
	Average Jaccard's Index	**1.00000**	**1.00000**	**1.00000**	0.403033	**1.00000**
	Average Canberra Distance	0.012065	**0.010789**	0.012796	0.238021	0.026267
Splice	Average Spearman Rank Correlation	0.994203	**0.994218**	0.993797	0.053527	0.965871
	Average Jaccard's Index	0.910949	**0.912467**	0.910025	0.289253	0.888437
	Average Canberra Distance	0.01733	**0.017227**	0.017946	0.26515	0.0394
Indian Pines	Average Spearman Rank Correlation	0.999915	0.999028	**0.999915**	0.34175	0.999584
	Average Jaccard's Index	**0.999286**	0.980462	0.999286	0.442751	0.955356
	Average Canberra Distance	**0.005959**	0.01017	0.005969	0.195324	0.013514
Semeion	Average Spearman Rank Correlation	0.999066	**0.99907**	0.999061	0.149696	0.994699
	Average Jaccard's Index	0.965517	**0.966032**	0.9652	0.234598	0.933886
	Average Canberra Distance	0.012241	**0.012223**	0.01223	0.259725	0.027009

Table 4.6. Stability analysis results using MRMR (by AT)

Dataset	Stability Measure	Arithmetic Mean	Geometric Mean	L2 Norm	Stuart	RRA
jasmine	Average Spearman Rank Correlation	**0.985348**	0.975607	0.969901	0.127768	0.945334
	Average Jaccard's Index	0.920859	0.712669	0.860327	0.28029	**0.942433**
	Average Canberra Distance	**0.037728**	0.071905	0.052573	0.265316	0.062051
Image	Average Spearman Rank Correlation	0.990206	**0.999271**	0.991754	0.077584	0.986775
	Average Jaccard's Index	0.824668	**1.00000**	0.911903	0.34549	0.877772
	Average Canberra Distance	0.044145	**0.000756**	0.034265	0.260363	0.052491
Scene	Average Spearman Rank Correlation	**0.991528**	0.867312	0.990774	0.037092	0.985775
	Average Jaccard's Index	0.883476	**0.919565**	0.912394	0.352322	0.85963
	Average Canberra Distance	0.048357	0.043893	**0.035833**	0.260327	0.058293
Musk	Average Spearman Rank Correlation	0.999319	0.957317	**0.9995**	0.192018	0.997759
	Average Jaccard's Index	0.971851	0.975257	**0.9864**	0.331341	0.920576
	Average Canberra Distance	0.011183	0.014444	**0.008829**	0.24121	0.019513
Philippine	Average Spearman Rank Correlation	**0.997471**	0.886464	0.99692	0.124873	0.991156
	Average Jaccard's Index	0.944297	0.896914	**0.973887**	0.303979	0.855237
	Average Canberra Distance	0.022058	0.04292	**0.017112**	0.252163	0.041142
Ionosphere	Average Spearman Rank Correlation	0.986624	0.823187	**0.987527**	0.127751	0.964272
	Average Jaccard's Index	**0.96563**	0.755886	0.860929	0.327636	0.833535
	Average Canberra Distance	**0.033533**	0.092643	0.038098	0.264702	0.047948
Optdigits	Average Spearman Rank Correlation	0.997515	0.96065	**0.998079**	0.220924	0.990739
	Average Jaccard's Index	0.948225	0.881989	**0.982315**	0.453761	0.827538
	Average Canberra Distance	0.017777	0.025685	**0.015212**	0.220903	0.034954
Ada	Average Spearman Rank Correlation	**0.997238**	0.97531	0.992067	0.181563	0.989869
	Average Jaccard's Index	**1.00000**	0.693282	0.925326	0.35123	0.939871
	Average Canberra Distance	**0.013484**	0.056628	0.0219	0.249259	0.027471
Splice	Average Spearman Rank Correlation	**0.993365**	0.95587	0.977623	0.048368	0.954847
	Average Jaccard's Index	**0.907029**	0.658117	0.850792	0.282154	0.898472
	Average Canberra Distance	**0.024897**	0.096655	0.042943	0.274512	0.048559
Indian Pines	Average Spearman Rank Correlation	0.995184	0.982723	0.985878	0.077339	**0.995655**
	Average Jaccard's Index	0.923521	0.899203	**0.930099**	0.332715	0.909569
	Average Canberra Distance	**0.026688**	0.0313	0.02894	0.252948	0.027732
Semeion	Average Spearman Rank Correlation	0.998864	**0.998867**	0.998861	0.133935	0.994133
	Average Jaccard's Index	0.950387	0.948793	**0.951807**	0.264046	0.905127
	Average Canberra Distance	0.015151	**0.015151**	0.015215	0.256725	0.030999

Chapter 5. Conclusion and Future Work

Over the years, datasets have grown increasingly larger in size and dimensionality. To mitigate the curse of dimensionality in high-dimensional datasets, feature selection has become an essential preprocessing technique in machine learning applications, as well as the focus of a wide spectrum of literature spanning practice in several disciplines. However, no feature selection method is able to consistently deliver optimal performance across different application fields. For this reason and in order to improve the stability of the feature selection process, ensemble feature selection frameworks have become increasingly popular. Because ensemble feature selection frameworks combine the findings of several feature selection iterations, their adoption is thought to lower the likelihood of picking an unstable feature subset. The degree of variation in the selected features, given minor changes in the training data used to select them, characterizes the feature selection stability or robustness. This element of feature selection assessment has received a lot of attention in recent years, as practitioners now recognize the importance of having feature selection results that are resilient to fluctuations in the training data.

In contribution to this field, this thesis develops a general framework for ensemble feature selection via bootstrap induced diversity. Using this ensemble framework, importance scores are aggregated within and between different feature selection techniques in order to reduce the input dimensionality and improve the stability of the selected features. The tested ensemble framework is thus validated on real-life datasets and analyzed in terms of both the classification performance and stability behavior. While many have examined the construction of ensemble techniques under various conditions, very little work has shed light on the impact of the aggregation techniques themselves on the stability of the feature selection strategy. Therefore, this work also examines how the robustness and accuracy of the aggregation process influences the ensemble feature selection. To this end, five different aggregation approaches are evaluated and compared using twelve real datasets from a variety of application fields, in terms of both the classification performance and the stability influence. Moreover, the experimental evaluation includes four filter feature selection methods and a variety of stability criteria. A merit of this work is that it singles out the stability resulted from the aggregation procedure alone. This has seldom been thoroughly investigated in the literature before, and not

with the focus on the underused score-based aggregation procedures.

The extensive experimental analysis of twelve real datasets across different application fields demonstrates the effectiveness of the WAM and BAM frameworks in improving the performance of the learning algorithm, guiding the selection of the optimal feature subsets, and facilitating the identification of the most appropriate feature selection method for a given dataset. Both methods are comparable in terms of their accuracy scores, computational costs, and ability to determine the optimal feature reduction percentages. However, the BAM is demonstrably slower and less consistent across classification experiments. In particular, the accuracy differences between the WAM and BAM are found to be insignificant in the Random Forest and Naive Bayes classifiers, and less pronounced when higher feature reduction percentages are given more weight. Conversely, the WAM demonstrates better stability behavior than the BAM across most datasets, whereas the BAM stability scores fall in the middle range of the computed values on most of the similarity metrics. In addition, a comparison of the rank and index-based stability results for the WAM and BAM indicates that lower feature ranks are associated with higher confidence in the selected feature subsets. Based on the results of the experiments and the extent to which the feature selection can be influenced by the data composition and learning algorithm, we recommend that both BAM and WAM methods be implemented in order to achieve better insight into the underlying application domain and to guide the selection of the most important features for the given dataset. If the computational cost is a concern, then the WAM can be recommended over the BAM. Nonetheless, it is also important to note that optimizing the computational cost depends largely on the dataset properties and the learning algorithm utilized.

On the other hand, the analysis of the stability and accuracy behavior of the WAM ensemble demonstrates significant results for the tested score-based (Arithmetic Mean, Geometric Mean, L2 Norm) and rank-based (Robust Rank Aggregation, Stuart) aggregations. That is, the results of the experimental evaluation highlight the strengths of the implemented score-based aggregations in comparison to the rank-based methods. In terms of classification accuracy, the performance of the aggregation approaches is found to be generally data-dependent, with significant disparities in the classification accuracy results between score-based and rank-based aggregation procedures. In terms

of stability behavior, the performance of the aggregation methods shows similar disparities, though the highest stability scores are almost always attributed to one of the three score-based aggregation functions. Moreover, the stability analysis of the ensemble framework exhibits improved feature selection stability across most aggregation functions in comparison to prior implementations. In particular, the experimental findings demonstrate that the Arithmetic Mean and L2 Norm outperform the other aggregation procedures in terms of stability, and that these two score-based aggregations consistently deliver good performance, allowing them to be recommended in many problem settings.

Overall, these findings validate the ensemble frameworks introduced in this thesis and highlight both the accuracy and stability differences between score-based and rank-based aggregations. Moreover, the experimental results obtained demonstrate the efficiency of using simpler aggregations such as the Arithmetic Mean over more complex alternatives. Given that the scores have stronger interval scale than the ranks and can possibly better differentiate between the features, they can be considered better suited for aggregation. This research, however, is confined to the aforementioned aggregation methodologies as well as the binary and multiclass classification datasets that were investigated. In turn, future research might include more score and rank-based aggregations in the comparison, as well as expand the underlying feature selection approaches to include embedded and wrapper techniques. Such findings can have significant practical implications as they guide the ensemble feature selection methods to the most efficient aggregation rule given the data structure and corresponding application domain.

References

[1] B. Liu, "Supervised learning," *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*, pp. 55–116, 2007.

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

[3] P. Brazdil, C. G. Carrier, C. Soares, and R. Vilalta, *Metalearning: Applications to data mining*. Springer Science & Business Media, 2008.

[4] A. Kalousis, J. Prados, and M. Hilario, "Stability of feature selection algorithms: A study on high-dimensional spaces," *Knowledge and information systems*, vol. 12, no. 1, pp. 95–116, 2007.

[5] S. Nogueira and G. Brown, "Measuring the stability of feature selection," in *Joint European conference on machine learning and knowledge discovery in databases*, Springer, 2016, pp. 442–457.

[6] B. Pes, "Evaluating feature selection robustness on high-dimensional data," in *International Conference on Hybrid Artificial Intelligence Systems*, Springer, 2018, pp. 235–247.

[7] S. Alelyani, "Stable bagging feature selection on medical data," *Journal of Big Data*, vol. 8, no. 1, pp. 1–18, 2021.

[8] C. Lai, M. J. Reinders, L. J. van't Veer, and L. F. Wessels, "A comparison of univariate and multivariate gene selection techniques for classification of cancer datasets," *BMC bioinformatics*, vol. 7, no. 1, pp. 1–10, 2006.

[9] S. H. Huang, "Dimensionality reduction in automatic knowledge acquisition: A simple greedy search approach," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 6, pp. 1364–1373, 2003.

[10] G. Forman *et al.*, "An extensive empirical study of feature selection metrics for text classification.," *J. Mach. Learn. Res.*, vol. 3, no. Mar, pp. 1289–1305, 2003.

[11] D. Mo and S. H. Huang, "Feature selection based on inference correlation," *Intelligent Data Analysis*, vol. 15, no. 3, pp. 375–398, 2011.

[12] H. Sulieman and A. Alzaatreh, "A supervised feature selection approach based on global sensitivity," *Archives of Data Science, Series A (Online First)*, vol. 5, no. 1, p. 03, 2018.

[13] S. A. Shahee and U. Ananthakumar, "An effective distance based feature selection approach for imbalanced data," *Applied Intelligence*, vol. 50, no. 3, pp. 717–745, 2020.

[14] J. T. Kent, "Information gain and a general measure of correlation," *Biometrika*, vol. 70, no. 1, pp. 163–173, 1983.

[15] I. H. Witten and E. Frank, "Data mining: Practical machine learning tools and techniques with java implementations," *Acm Sigmod Record*, vol. 31, no. 1, pp. 76–77, 2002.

[16] P. E. Greenwood and M. S. Nikulin, *A guide to chi-squared testing*. John Wiley & Sons, 1996, vol. 280.

[17] I. Kononenko, "Estimating attributes: Analysis and extensions of relief," in *European conference on machine learning*, Springer, 1994, pp. 171–182.

[18] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.

[19] M. A. Hall, "Correlation-based feature selection for machine learning," 1999.

[20] M. Dash and H. Liu, "Consistency-based search in feature selection," *Artificial intelligence*, vol. 151, no. 1-2, pp. 155–176, 2003.

[21] J. Tang, S. Alelyani, and H. Liu, "Feature selection for classification: A review," *Data classification: Algorithms and applications*, p. 37, 2014.

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

[23] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," 1988.

[24] J. Xia, L. Sun, S. Xu, Q. Xiang, J. Zhao, W. Xiong, Y. Xu, and S. Chu, "A model using support vector machines recursive feature elimination (svm-rfe) algorithm to classify whether copd patients have been continuously managed according to gold guidelines," *International Journal of Chronic Obstructive Pulmonary Disease*, vol. 15, p. 2779, 2020.

[25] K. J. Archer and R. V. Kimes, "Empirical characterization of random forest variable importance measures," *Computational statistics & data analysis*, vol. 52, no. 4, pp. 2249–2260, 2008.

[26] H.-H. Hsu, C.-W. Hsieh, and M.-D. Lu, "Hybrid feature selection by combining filters and wrappers," *Expert Systems with Applications*, vol. 38, no. 7, pp. 8144–8150, 2011.

[27] N. Almugren and H. Alshamlan, "A survey on hybrid feature selection methods in microarray gene expression data for cancer classification," *IEEE Access*, vol. 7, pp. 78 533–78 548, 2019.

[28] Y. Saeys, T. Abeel, and Y. Van de Peer, "Robust feature selection using ensemble feature selection techniques," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, 2008, pp. 313–325.

[29] H. Wang, T. M. Khoshgoftaar, and A. Napolitano, "A comparative study of ensemble feature selection techniques for software defect prediction," in *2010 Ninth International Conference on Machine Learning and Applications*, IEEE, 2010, pp. 135–140.

[30] N. Hoque, M. Singh, and D. K. Bhattacharyya, "Efs-mi: An ensemble feature selection method for classification," *Complex & Intelligent Systems*, vol. 4, no. 2, pp. 105–118, 2018.

[31] P. Drotár, M. Gazda, and L. Vokorokos, "Ensemble feature selection using election methods and ranker clustering," *Information Sciences*, vol. 480, pp. 365–380, 2019.

[32] C.-W. Chen, Y.-H. Tsai, F.-R. Chang, and W.-C. Lin, "Ensemble feature selection in medical datasets: Combining filter, wrapper, and embedded feature selection results," *Expert Systems*, vol. 37, no. 5, e12553, 2020.

[33] G. Brown, “Ensemble learning.,” *Encyclopedia of machine learning*, vol. 312, pp. 15–19, 2010.

[34] W. W. Ng, Y. Tuo, J. Zhang, and S. Kwong, “Training error and sensitivity-based ensemble feature selection,” *International Journal of Machine Learning and Cybernetics*, vol. 11, no. 10, pp. 2313–2326, 2020.

[35] M. Bramer, *Principles of data mining*. Springer, 2007, vol. 180.

[36] B. Seijo-Pardo, I. Porto-Díaz, V. Bolón-Canedo, and A. Alonso-Betanzos, “Ensemble feature selection: Homogeneous and heterogeneous approaches,” *Knowledge-Based Systems*, vol. 118, pp. 124–139, 2017.

[37] B. Seijo-Pardo, V. Bolón-Canedo, I. Porto-Díaz, and A. Alonso-Betanzos, “Ensemble feature selection for rankings of features,” in *International Work-Conference on Artificial Neural Networks*, Springer, 2015, pp. 29–42.

[38] J. Wang, J. Xu, C. Zhao, Y. Peng, and H. Wang, “An ensemble feature selection method for high-dimensional data based on sort aggregation,” *Systems Science & Control Engineering*, vol. 7, no. 2, pp. 32–39, 2019.

[39] B. Pes, N. Dessì, and M. Angioni, “Exploiting the ensemble paradigm for stable feature selection: A case study on high-dimensional genomic data,” *Information Fusion*, vol. 35, pp. 132–147, 2017.

[40] R. S. Subramanian and D Prabha, “Customer behavior analysis using naive bayes with bagging homogeneous feature selection approach,” *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–12, 2020.

[41] C.-F. Tsai and Y.-T. Sung, “Ensemble feature selection in high dimension, low sample size datasets: Parallel and serial combination approaches,” *Knowledge-Based Systems*, vol. 203, p. 106 097, 2020.

[42] V. Bolón-Canedo and A. Alonso-Betanzos, “Ensembles for feature selection,” in *Recent Advances in Ensembles for Feature Selection*, Springer, 2018, pp. 53–81.

[43] N. Dessì, E. Pascariello, and B. Pes, “A comparative analysis of biomarker selection techniques,” *BioMed research international*, vol. 2013, 2013.

[44] N. Gopalakrishnan, V. Krishnan, and V. Gopalakrishnan, “Ensemble feature selection to improve classification accuracy in human activity recognition,” in *Inventive Communication and Computational Technologies*, Springer, 2020, pp. 541–548.

[45] V. Bolón-Canedo and A. Alonso-Betanzos, “Ensembles for feature selection: A review and future trends,” *Information Fusion*, vol. 52, pp. 1–12, 2019.

[46] A. Tsymbal, S. Puuronen, and D. W. Patterson, “Ensemble feature selection with the simple bayesian classification,” *Information fusion*, vol. 4, no. 2, pp. 87–100, 2003.

[47] K. L. Chiew, C. L. Tan, K. Wong, K. S. Yong, and W. K. Tiong, “A new hybrid ensemble feature selection framework for machine learning-based phishing detection system,” *Information Sciences*, vol. 484, pp. 153–166, 2019.

[48] B. Seijo-Pardo, V. Bolón-Canedo, and A. Alonso-Betanzos, “On developing an automatic threshold applied to feature selection ensembles,” *Information Fusion*, vol. 45, pp. 227–245, 2019.

[49] B. Sahu, S. Dehuri, and A. K. Jagadev, “An ensemble model using genetic algorithm for feature selection and rule mining using apriori and fp-growth from cancer microarray data,” *International Journal of Applied Engineering Research*, vol. 12, no. 10, pp. 2391–2408, 2017.

[50] T. Gangavarapu and N. Patil, “A novel filter–wrapper hybrid greedy ensemble approach optimized using the genetic algorithm to reduce the dimensionality of high-dimensional biomedical datasets,” *Applied Soft Computing*, vol. 81, p. 105 538, 2019, ISSN: 1568-4946. DOI: https://doi.org/10.1016/j.asoc.2019.105538.

[51] N. Dessì, B. Pes, and M. Angioni, “On stability of ensemble gene selection,” in *International Conference on Intelligent Data Engineering and Automated Learning*, Springer, 2015, pp. 416–423.

[52] B. Pes, “Ensemble feature selection for high-dimensional data: A stability analysis across multiple domains,” *Neural Computing and Applications*, pp. 1–23, 2019.

[53] D. Álvarez-Estévez, N. Sánchez-Maroño, A. Alonso-Betanzos, and V. Moret-Bonillo, “Reducing dimensionality in a database of sleep eeg arousals,” *Expert Systems with Applications*, vol. 38, no. 6, pp. 7746–7754, 2011.

[54] V. Bolón-Canedo, N. Sánchez-Marono, and A. Alonso-Betanzos, “Data classification using an ensemble of filters,” *Neurocomputing*, vol. 135, pp. 13–20, 2014.

[55] A. B. Brahim and M. Limam, “Ensemble feature selection for high dimensional data: A new method and a comparative study,” *Advances in Data Analysis and Classification*, vol. 12, no. 4, pp. 937–952, 2018.

[56] A. Onan and S. Korukoğlu, “A feature selection model based on genetic rank aggregation for text sentiment classification,” *Journal of Information Science*, vol. 43, no. 1, pp. 25–38, 2017.

[57] S. Najdi, A. A. Gharbali, and J. M. Fonseca, “Feature ranking and rank aggregation for automatic sleep stage classification: A comparative study,” *Biomedical engineering online*, vol. 16, no. 1, pp. 1–19, 2017.

[58] J. D. López-Cabrera and J. V. Lorenzo-Ginori, “Feature selection for the classification of traced neurons,” *Journal of neuroscience methods*, vol. 303, pp. 41–54, 2018.

[59] P. Willett, “Combination of similarity rankings using data fusion,” *Journal of chemical information and modeling*, vol. 53, no. 1, pp. 1–10, 2013.

[60] S. Aerts, D. Lambrechts, S. Maity, P. Van Loo, B. Coessens, F. De Smet, L.-C. Tranchevent, B. De Moor, P. Marynen, B. Hassan, *et al.*, “Gene prioritization through genomic data fusion,” *Nature biotechnology*, vol. 24, no. 5, pp. 537–544, 2006.

[61] R. Kolde, S. Laur, P. Adler, and J. Vilo, “Robust rank aggregation for gene list integration and meta-analysis,” *Bioinformatics*, vol. 28, no. 4, pp. 573–580, 2012.

[62] T. Joachims, “Optimizing search engines using clickthrough data,” in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2002, pp. 133–142.

[63] A. Woznica, P. Nguyen, and A. Kalousis, “Model mining for robust feature selection,” in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2012, pp. 913–921.

[64] D. J. Dittman, T. M. Khoshgoftaar, R. Wald, and A. Napolitano, “Classification performance of rank aggregation techniques for ensemble gene selection,” in *The Twenty-Sixth International FLAIRS Conference*, 2013.

[65] R. Wald, T. M. Khoshgoftaar, D. Dittman, W. Awada, and A. Napolitano, “An extensive comparison of feature ranking aggregation techniques in bioinformatics,” in *2012 IEEE 13th International Conference on Information Reuse & Integration (IRI)*, IEEE, 2012, pp. 377–384.

[66] R. Wald, T. M. Khoshgoftaar, and D. Dittman, “Mean aggregation versus robust rank aggregation for ensemble gene selection,” in *2012 11th International Conference on Machine Learning and Applications*, IEEE, vol. 1, 2012, pp. 63–69.

[67] D. J. Dittman, T. M. Khoshgoftaar, R. Wald, and A. Napolitano, “Comparison of rank-based vs. score-based aggregation for ensemble gene selection,” in *2013 IEEE 14th International Conference on Information Reuse & Integration (IRI)*, IEEE, 2013, pp. 225–231.

[68] Y. Li, D. F. Hsu, and S. M. Chung, “Combining multiple feature selection methods for text categorization by using rank-score characteristics,” in *2009 21st IEEE International Conference on Tools with Artificial Intelligence*, IEEE, 2009, pp. 508–517.

[69] D. Dernoncourt, B. Hanczar, and J.-D. Zucker, “Stability of ensemble feature selection on high-dimension and low-sample size data,” in *Proceedings of the 3rd International Conference on Pattern Recognition Applications and Methods*, 2014, pp. 325–330.

[70] S. Alelyani, Z. Zhao, and H. Liu, “A dilemma in assessing stability of feature selection algorithms,” in *2011 IEEE International Conference on High Performance Computing and Communications*, IEEE, 2011, pp. 701–707.

[71] D. Dittman, T. Khoshgoftaar, R. Wald, and A. Napolitano, “Similarity analysis of feature ranking techniques on imbalanced dna microarray datasets,” in *2012*

IEEE International Conference on Bioinformatics and Biomedicine, IEEE, 2012, pp. 1–5.

[72] R. Wald, T. M. Khoshgoftaar, and A. Napolitano, "Stability of filter-and wrapper-based feature subset selection," in *2013 IEEE 25th International Conference on Tools with Artificial Intelligence*, IEEE, 2013, pp. 374–380.

[73] K. Dunne, P. Cunningham, and F. Azuaje, "Solutions to instability problems with sequential wrapper-based approaches to feature selection," *Journal of Machine Learning Research*, pp. 1–22, 2002.

[74] L. I. Kuncheva, "A stability index for feature selection.," in *Artificial intelligence and applications*, 2007, pp. 421–427.

[75] F. Yang and K. Mao, "Robust feature selection for microarray data based on multi-criterion fusion," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 8, no. 4, pp. 1080–1092, 2010.

[76] E. Yu and S. Cho, "Ensemble based on ga wrapper feature selection," *Computers & industrial engineering*, vol. 51, no. 1, pp. 111–116, 2006.

[77] T. G. Dietterich, "Machine-learning research," *AI magazine*, vol. 18, no. 4, pp. 97–97, 1997.

[78] U. M. Khaire and R Dhanalakshmi, "Stability of feature selection algorithm: A review," *Journal of King Saud University-Computer and Information Sciences*, 2019.

[79] G. Jurman, S. Riccadonna, R. Visintainer, and C. Furlanello, "Canberra distance on ranked lists," in *Proceedings of advances in ranking NIPS 09 workshop*, Citeseer, 2009, pp. 22–27.

[80] Z. Shen, X. Chen, and J. M. Garibaldi, "A novel weighted combination method for feature selection using fuzzy sets," in *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, IEEE, 2019, pp. 1–6.

[81] J. R. Vergara and P. A. Estévez, "A review of feature selection methods based on mutual information," *Neural computing and applications*, vol. 24, no. 1, pp. 175–186, 2014.

[82] R. Arias-Michel, M. García-Torres, C. Schaerer, and F. Divina, "Feature selection using approximate multivariate markov blankets," in *International Conference on Hybrid Artificial Intelligence Systems*, Springer, 2016, pp. 114–125.

[83] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature selection: A data perspective," *ACM Computing Surveys (CSUR)*, vol. 50, no. 6, pp. 1–45, 2017.

[84] C. Ding and H. Peng, "Minimum redundancy feature selection from microarray gene expression data," *Journal of bioinformatics and computational biology*, vol. 3, no. 02, pp. 185–205, 2005.

[85] M. Grabisch, J.-L. Marichal, R. Mesiar, and E. Pap, *Aggregation functions*. Cambridge University Press, 2009, vol. 127.

[86] G. H. John and P. Langley, "Estimating continuous distributions in bayesian classifiers," *arXiv preprint arXiv:1302.4964*, 2013.

[87] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[88] E. García-Gonzalo, Z. Fernández-Muñiz, P. J. García Nieto, A. Bernardo Sánchez, and M. Menéndez Fernández, "Hard-rock stability analysis for span design in entry-type excavations with learning classifiers," *Materials*, vol. 9, no. 7, p. 531, 2016.

[89] I. Guyon, L. Sun-Hosoya, M. Boullé, H. J. Escalante, S. Escalera, Z. Liu, D. Jajetic, B. Ray, M. Saeed, M. Sebag, A. Statnikov, W.-W. Tu, and E. Viegas, "Analysis of the AutoML Challenge Series 2015–2018," in *Automated Machine Learning*, F. Hutter, L. Kotthoff, and J. Vanschoren, Eds., Series Title: The Springer Series on Challenges in Machine Learning, Cham: Springer International Publishing, 2019, pp. 177–219, ISBN: 978-3-030-05317-8 978-3-030-05318-5. DOI: 10.1007/978-3-030-05318-5_10.

[90] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern Recognition*, vol. 37, no. 9, pp. 1757–1771, Sep. 2004, ISSN: 0031-3203. DOI: 10.1016/j.patcog.2004.03.009.

[91] "Multi-Instance Multi-Label Learning with Application to Scene Classification," in *Advances in Neural Information Processing Systems 19*, B. Schölkopf, J. Platt, and T. Hofmann, Eds., The MIT Press, 2007, ISBN: 978-0-262-25691-9. DOI: 10.7551/mitpress/7503.003.0206.

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

[93] M. F. Baumgardner, L. L. Biehl, and D. A. Landgrebe, *220 band aviris hyperspectral image data set: June 12, 1992 indian pine test site 3*, 2015. DOI: doi:/10.4231/R7RX991C. [Online]. Available: https://purr.purdue.edu/publications/1947/1.

Vita

Reem Elfatih was born in 1998, in Dubai, UAE. She received her Bachelor's degree in Mathematics from the American University of Sharjah (AUS) in June 2020. As an undergraduate, Reem worked as a grader, tutor, and research assistant at the department of Mathematics and Statistics. She was an active volunteer at the university's community services center and has been awarded the Al Ghurair STEM Scholarship throughout the duration of her study.

In September 2020, she joined the AUS Master's program as a graduate teaching and research assistant at the department of Mathematics and Statistics. In Spring 2022, Reem was awarded the Graduate Student Research, Scholarly, and Creative Work Excellence Award. Her research interests include the analysis of big data, feature selection, and the assessment of stability in machine learning applications.