# An Investigation of Fuzzy Methods and Meta Learning for Feature Selection

**Zixiao Shen**

School of Computer Science

University of Nottingham

This dissertation is submitted for the degree of

*Doctor of Philosophy*

June 2022

# Acknowledgements

I would like to express my sincere gratitude to my supervisor Prof. Jonathan M. Garibaldi, for his invaluable advice, support, and time throughout my Ph.D. journey. He is a tremendous mentor for me, always full of kindness, and gives me much help when I meet difficulties. I would like to thank my supervisor Dr. Xin Chen for his consistent help and detailed guidance with his visionary and insightful thinking. This thesis would not have been possible without any of them.

I also would like to thank my family for their unconditional support for my academic dreams and aspirations, the University of Nottingham for funding my Ph.D. research, and my lovely friends who always encourage me throughout the ups and downs of those years.

# Abstract

Recent developments in technology have led to accelerated growth of data, and the associated challenges of extracting information from them. Recently, the knowledge discovery process has become a central issue in extracting knowledge from data. Within this, feature selection (FS) acts as a preprocessing procedure, playing an essential role in aiming to discover a minimal feature subset or a reliable feature ranking sequence to represent the original data. However, practical datasets are inherently uncertain and imperfect due to the noise, incompleteness, and inconsistency which always exists. In this research, the fuzzy theory is introduced as a unified framework to model these uncertainties in the FS process.

Unlike semantics-preserving FS algorithms that output a final feature subset, this research aims to explore efficient feature ranking-based methods that rank the features based on feature importance. To begin with, this research investigates a fuzzy entropy-based FS and classification framework with several essential components. Different evaluation measurements and functions are implemented to find the combination which achieves the best performance. The proposed method has produced relatively high and stable classification accuracy when gradually removing features, indicating better performance than other comparable methods. Further, on account of the lack of suitable measurements to evaluate and compare FS algorithms effectively, two new evaluation methods are proposed on the aspects of accuracy and robustness. The proposed weighted accuracy and robustness measures have proven to be more sensitive on real-world and synthetic datasets. A multi-criteria evaluation method based on radar charts is also introduced, to comprehensively measure overall performance.

Next, this research investigates and proposes a novel ensemble learning framework to further improve FS algorithms' performance. The proposed method consists of three main steps: distribution generation of feature importance using bootstrap, distribution ensemble using aggregation methods, and defuzzification for feature ranking. Various methods represent the importance of features, such as score-based, rank-based, and fuzzy-based approaches. Both weighted combination and fuzzy aggregation methods are used to aggregate the different distributions. Following tuning using a reference data repository, the best combination approach is chosen for each of the score, rank, and fuzzy-based approaches, respectively. Compared with the base FS methods after the bootstrap process and the other state-of-the-art FS methods, the proposed methods have produced better performance on the testing data repository, especially for the technique using a fuzzy-based approach and drastic sum S-norm aggregation. It has shown that the proposed ensemble learning framework can improve the performance of FS algorithms in multiple aspects.

From the literature, there are a large number of FS methods, and it is impossible to state the best FS method for all kinds of data. Hence, this research finally develops a meta-learning framework to recommend a suitable FS method for a given dataset. Various synthetic datasets are generated as the training data repository, which is used to tune the parameters of the framework. Subsequently, a meta-learning framework is constructed by extracting six meta-features of the training data repository, applying the FS algorithms with the best multi-criteria performance as the meta labels, and utilizing the fuzzy similarity measure-based method as the classifier. In experiments, the proposed framework successfully recommends the best FS method from the candidate methods for six out of ten testing datasets with negligible additional time.

The limitations of the proposed methods, possible improvements, and future research directions are discussed in the last chapter of the thesis.

# Table of contents

# List of figures

# List of tables

# Chapter 1

# Introduction

The rapid development of cutting-edge technologies in information and computer science has primarily increased efficiency and productivity in various aspects. In the meantime, it effectively generates vast amounts of data [43, 183]. Accelerated growth of data and the challenges associated with extracting meaningful information from them have given rise to the field of knowledge discovery. This field describes the process of uncovering previously unknown trends, patterns, relationships, and knowledge from data [80]. As Rutherford D. Rogers mentions, *"we are drowning in information and starving for knowledge"* [176]. A growing gap exists between the ever increasing volumes of data being generated and our limited abilities to understand it.

## 1.1 Background

Typically, data is composed of a number of examples that are characterized by some features. The majority of the datasets used in the research come from a single data source which is always assumed to be independent and identically distributed (i.i.d). On the other hand, the data could also come from multiple and heterogeneous sources such as text, images, tags

and videos in social media. In addition, stream data has become more and more prevalent in real-world applications, such as user-generated post data on Twitter [128].

### 1.1.1 Knowledge Discovery Process

Knowledge is valuable when it can be used efficiently and effectively. Recently, the knowledge discovery process has become central to this issue [90]. The term knowledge discovery refers to the process of finding knowledge in data, which emphasizes the high-level application of particular data mining methods. Nowadays, discovering knowledge from data has become an essential and challenging target. Recent decades have also witnessed a revolution in the theory and application of artificial intelligence to achieve superior performance in many areas, such as decision making, clustering, regression, etc [113]. In general, the overall process of finding and interpreting knowledge from data involves the repeated application of the steps below [55].

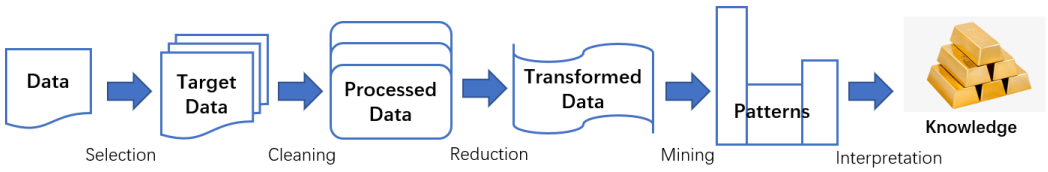



Fig. 1.1 An outline of steps of the knowledge discovery process

1. **Data Selection**: The target dataset is created by selecting a dataset or focusing on a subset of variables or data samples on which discovery is performed.

2. **Data Cleaning**: This step includes removing noise or outliers, handling any missing values, and collecting the necessary information to model noise.

3. **Data Reduction**: This step aims to find useful features to represent the data using dimensionality reduction or transformation methods.

4. **Data Mining**: This step searches for patterns of interest in a particular typical form, such as classification, regression and clustering.

5. **Interpretation/Evaluation**: This step evaluates the knowledge concerning validity, usefulness, novelty, and simplicity.

Within this generic framework, the data mining stage represents the study of algorithms that automatically improve their performance with experience using machine learning methods. Machine learning algorithms can be broadly categorized in the literature into supervised and unsupervised learning methods. Supervised learning is the machine learning task of learning a function that maps an input to an output using labelled datasets [177]. On the other hand, unsupervised learning learns the patterns from untagged data using machine learning algorithms. These algorithms discover the hidden patterns in data without labelling information [82].

### 1.1.2 Dimensionality Reduction

Practically, the extensive and complex data commonly face the curse of dimensionality problem. In some cases, the limited number of training samples are sparsely distributed within a high dimensional feature space [10]. The concept "sparsely distributed" indicates when the number of samples is far less than the number of features. Then the distribution of the limited samples becomes sparse in the data space. However, many features are either irrelevant or redundant, which do not provide any useful or additional information. When the learned models are not generic for unseen data samples, it may also lead to the overfitting problem [119]. Moreover, the high dimensional datasets could also lead to a significant increase in memory usage and high computational cost in the data analysis process [128].

The principle of Occam's Razor states that *"plurality ought never to be posited without necessity"* [182]. It is also called the principle of parsimony and paraphrased as "the simplest

explanation is usually the best one" [15]. This principle has been widely applied as a scientific and philosophical rule to mean that entities should not be multiplied unnecessarily. With the increasing amount of available big data, dimensionality reduction techniques gradually become active research topics and play significant roles in the issues above. Dimensionality reduction decreases the dimensions of the data space such that the low-dimensional representation maintains some meaningful characteristics of the original space, which is ideally close to its intrinsic properties [206]. There are two main approaches to dimensionality reduction, being feature extraction and feature selection. Feature extraction is a process that extracts a set of new features from the original ones through some functional mapping. Using linear or nonlinear combinations, it projects the original high dimensional feature space into a new low dimensional one. Some well known examples include Principle Component Analysis (PCA) [217], Linear Discriminant Analysis (LDA) [143], ISOMAP [201], etc.

In comparison, feature selection (FS) is a process that chooses a subset of features from the original feature set. Instead of creating new feature combinations, FS methods include and exclude features presented in data without changing their morphology [69]. It has been widely used as the pre-processing technique for supervised and unsupervised learning applications, such as classification, clustering and regression. Even though the label information is needed in some FS methods to guide the selection process, many unsupervised FS methods still exist independent of the labelling and are suitable for unsupervised scenarios. In the knowledge discovery area, FS methods have demonstrated many potential benefits over the past decades. Firstly, they are ordinarily computationally efficient and can provide highly readable and interpretable results [90]. Secondly, by selecting the discriminant features, they can help in the visualization and understanding of the characteristics of data. Thirdly, the use of FS methods also reduces the storage, training, and measuring requirements. Due to the reasons above, this thesis mainly focuses on FS methods.

### 1.1.3 Feature Selection Methods

In recent decades, FS methods have become one of the most frequently used and essential techniques in the data preprocessing stage [119]. By directly selecting a subset of relevant features for model construction, FS methods maintain the physical meaning of the original features and give models better readability and interpretability compared to feature extraction methods [128]. There are three types of FS methods, i.e. wrapper, embedded and filter methods. Wrapper methods select valuable features depending on specified learning algorithms, while embedded methods integrate the FS process into learning. In contrast, filter FS methods are independent of learning algorithms in that they identify a feature subset from the original space based on a set of given evaluation criteria.

From the literature, FS methods are widely applied in many practical problems. Firstly, FS methods are generally used to improve or optimize decision-making performance. The underlying principle is similar to the peaking phenomenon where the prediction performance would decrease after a peak with the increasing number of features [90]. There are irrelevant and redundant features within a feature space in practical datasets. The irrelevant features provide very little or even no predictive power. The redundant features give no additional information than the currently selected ones [119]. It becomes challenging to model the data mixed with such irrelevant and redundant features in the knowledge discovery process. A correct selection of relevant features could lead to an improvement of the inductive learner in terms of learning speed, generalization capacity or simplicity, etc [16]. Therefore, FS methods can simultaneously increase predictive performance and reduce the computational cost by removing irrelevant and redundant features.

Secondly, FS methods can drastically reduce the feature space of very high-dimensional datasets. Otherwise, the high dimensionality of data may lead to the underfitting problem, thereby becoming unsuitable for further processing. This issue is prevalent in some areas, such as bioinformatics. For example, a typical bioinformatics problem might be distinguish-

ing cancer patients from healthy people using gene expression profiles [220]. However, because of the enormous expense of collecting the gene expression microarrays, the number of samples in the existing gene expression datasets is tiny, where few of them exceed a hundred. On the other hand, the number of genes or features typically surpasses thousands or ten thousand. This is named the "Large $P$, Small $N$" problem in the machine learning area and can be addressed using FS methods.

Thirdly, FS methods may minimize the presence of redundant features by reducing the dimensions of potentially extensive datasets. In some areas, the acquisition and measurement of specific features are inconvenient and inaccessible to a particular degree. Therefore, by determining which features should be available to systems, FS methods could help simplify the design and implementation of actual machine learning methods and minimize the requirements for collecting various features. In the biological realm, FS methods can help recommend new metabolic pathways by identifying the essential features. Besides, they can also assist in the identification process for the hidden connection of specific cellular processes [49].

Fourthly, by minimizing the dimensionality of input feature spaces and preserving the data semantics simultaneously, FS methods can help reduce the data storage requirements. The corresponding processing speed could also be accelerated by employing a simpler machine learning model after using FS methods [127].

### 1.1.4 Motivations and Limitations

As mentioned before, a suitable FS method aims to provide advantages such as better predictive performance, reduced computational requirement, and the enhanced identification of relevant features. However, some central issues still exist to be addressed based on the literature.

**1. Noise and uncertainty** In practical applications, the presence of noise is arguably inevitable. Noise is introduced in many stages, such as data acquisition and transformation. In general, there are two types of noise: feature noise and class noise. Feature noise occurs when independent features contain either incorrect or missing values. Class noise occurs with the inclusion of contradictory examples and misclassifications. Contradictory examples refer to the same examples which appear more than once with different classifications, while misclassifications stand for the instances labelled with the wrong classes [241]. Data interpretation, modelling, and performance evaluation processes are undoubtedly affected by these sources of noise. The reduced-sized or poorly qualified datasets may also deteriorate FS results. Furthermore, the noise could misguide the downstream decision-making process and lead to suboptimal results.

However, there is no ideal noise handling mechanism to clean or remove all noise altogether. Noise tolerance or so-called noise resistance generally becomes desirable for tools and techniques. Many approaches are proposed to enhance the noise tolerance of machine learning methods. However, this also leads to increased attention on the ability of noise resistance and robustness for FS methods [5, 70]. On the other hand, practical datasets are inherently uncertain and imperfect due to the existence of noise, incompleteness, and inconsistency. There exist many incomplete features and text-based information within the data from sensors, social media, financial records, etc [8]. Appropriate analysis of such data requires advanced analytical techniques for efficiently predicting future courses with high precision and advanced decision-making strategies [77]. In this case, it becomes an essential demand to provide data reduction and handle the uncertainty for crisp and real-valued datasets simultaneously. FS methods with the property of producing consistent and robust FS results remain desirable.

From the literature, fuzzy sets and systems could provide a mechanism where real-valued features are effectively managed [228]. Fuzzy theory acts as a unified framework to model

the vagueness, imprecision, and uncertainty present in information. Through transforming values into belonging to one or more fuzzy sets using the membership functions, vagueness and uncertainty of data are modelled and further exploited to enable reasoning [90]. Under these circumstances, there has been increasing attention on employing fuzzy theory and fuzzy techniques in the FS area in recent decades.

**2. Data dependent performance** A vast body of different FS methods exists in the literature. However, the proliferation of FS algorithms does not necessarily lead to a general framework to help select the most suitable methods amongst the many existing algorithms [16]. Besides, the relationship between different datasets and their FS performance is still unclear for many FS algorithms. To make a correct decision on choosing the most suitable algorithms, a user needs to know the domain knowledge well and to fully understand the technical details of the available algorithms. Otherwise, different FS methods have to be tested and evaluated on various kinds of datasets using a trial and error approach, which can prove to be time-consuming and costly, especially on very large datasets [16, 190]. Therefore, it becomes an issue to choose and apply suitable FS algorithms where their performance varies in a data-dependent manner.

**3. Lack of attention on stability** FS methods' stability is defined as producing consistent feature preferences on different data subsets sampled from the same distribution. It quantifies how FS results are affected by different datasets from the same application scenario [102]. A stable FS algorithm aims to produce similar and consistent feature rankings or subsets under different training data variations [219]. The instability of input data may result in different outcomes, making the conclusion unreliable.

Nevertheless, in most cases, FS research mainly focuses on improving the predictive performance and extracting a smaller feature subset. In many practical applications such as gene selection, biological recognition, and cancer detection, a suitable FS method requires

not only high prediction accuracies but also a high level of stability [62, 45]. However, surprisingly, the stability of FS techniques receive relatively little attention in the previous research.

With increased focus on research into high dimensional datasets, the stability of various FS methods becomes an increasingly important issue. On the one hand, improving the algorithms' stability can help select the relevant features with higher confidence and lower processing time. On the other hand, it also provides domain experts with quantified evidence on the results' reliability, which is particularly crucial in biological areas such as genomics, DNA-microarrays, proteomics, and mass spectrometry. The motivation for improving stability is to provide confidence in the analysis of results, and to select the features which are relatively robust to any perturbations of input data [102]. Otherwise, the ignorance of stability issues and inconsistent FS results may reduce the experts' assuredness during the selected features' validation process.

## 1.2 Aims and Objectives

The FS process achieves dimensionality reduction by identifying and removing irrelevant and redundant features from data without altering the features themselves. At the same time, it provides many potential benefits, such as improving the computational efficiency of decision-making processes, reducing the measurement and storage requirements of data, and facilitating the readability and interpretability of the resulting knowledge. Some FS methods identify the feature subset without ranking the feature importance, such as the wrapper approach. Alternatively, FS methods which rank the features and obtain a feature sequence according to the feature importance can better represent the inherent difference and significance among the features, and so it is these which are mainly investigated in this research.

Although many FS methods have been proposed in the literature, some issues remain to be addressed. Firstly, the performance of FS methods is primarily evaluated using the accuracy of the subsequent decision-making tasks, whereas the repeatability and robustness of the methods are not often also considered. Secondly, the instability of data noise and methods may primarily affect the FS result. Hence, a ranking-based FS method that incorporates fuzzy theory to handle uncertainty needs to be better investigated. Thirdly, an FS method may only perform well on a particular dataset type. The relationship between the datasets and their FS performance is still unclear for many FS methods, resulting in a lack of clarity amongst some researchers as to which method to select for which dataset [189]. Besides, many learning-based FS methods suffer from data scarcity for training, which leads to poor generalizability when being applied to a new dataset.

This essential research that this thesis aims to explore is to create solutions for a better understanding of feature importance, such that the features represent the inherent characteristics of data well and allow comprehensively good performance to be achieved. Some objectives to achieve this overall aim and to try to address many of the issues mentioned above are as follows.

1. Explore a ranking-based FS method that incorporates fuzzy theory to handle the various uncertainties that may be present, thereby producing good predictive performance.
2. Review the existing performance evaluation metrics for FS method comparison and propose new metrics if supplementary to the existing methods.
3. Develop an ensemble learning framework that combines different FS methods to achieve better feature ranking.
4. As an alternative to this ensemble learning framework, develop an FS recommendation framework to suggest a suitable FS method for a given dataset. In order to overcome the data scarcity problem, the feasibility of using synthesized data for training a meta-

learning framework to achieve feature ranking is also investigated, and evaluated on various real datasets.

## 1.3 Thesis Structure

The remainder of this thesis is organized with the following chapters. Chapter 2 presents the literature review and background knowledge of FS methods, fuzzy theory, ensemble learning and meta-learning. For FS methods, the definition, general procedures, and different approaches are introduced at the beginning. Next, a further review of filter FS methods is discussed. Four important concepts are introduced concerning some aspects of fuzzy theory, including fuzzy sets, fuzzy entropy, fuzzy similarity, and fuzzy systems. The ensemble learning and meta-learning approaches are discussed with basic concepts, techniques, and main issues are highlighted.

Chapter 3 presents a fuzzy entropy-based FS framework that includes three ideal vector calculations, three similarity measures, and three fuzzy entropy functions. Different feature removal orders based on fuzzy entropy values are evaluated and compared. Based on three public datasets' experimental results, the recommendation is made as to the optimized combination using the ideal vector, similarity measure, and fuzzy entropy function for FS. Besides, the performance of the optimized framework is compared with the other classical filter-based FS methods.

Chapter 4 introduces the experimental datasets and proposes several evaluation metrics to measure the FS algorithms' performance. At first, twenty practical datasets and a synthetic dataset are introduced with detailed descriptions. After reviewing the existing performance evaluation metrics, several novel evaluation methods are proposed for accuracy and robustness. A comprehensive evaluation measurement based on a radar chart is also introduced to measure FS methods' performance from a multi-criteria perspective.

Chapter 5 presents an ensemble framework to aggregate FS results to produce a better performance on multiple aspects. Three main steps are introduced in the framework: distribution generation of feature importance, distribution ensemble using aggregation methods, and defuzzification for feature ranking. Three different approaches are proposed to generate feature importance distributions, including score, rank, and fuzzy-based approaches. The experiments are conducted using different repositories for methods tuning and performance testing. Furthermore, the proposed method is compared with the other state-of-the-art FS methods.

Chapter 6 presents a meta-learning framework to recommend the FS method with the best multi-criteria performance for a given dataset using a fuzzy similarity classifier. First, the framework is introduced comprising the three main steps: training data repository generation, metadata construction, and recommendation using a fuzzy similarity measure. Next, the experiments are conducted using different data repositories for parameter tuning and performance testing.

Chapter 7 concludes the thesis by summarizing the main points and outcomes. The contributions and limitations of the research in this thesis are discussed. In the end, the scope for future work in various directions is reviewed.

## 1.4 Contributions to Knowledge

This research has contributed four peer-reviewed conference papers and one journal paper under preparation. The publications are listed below.

1. **Zixiao Shen**, Xin Chen and Jonathan M. Garibaldi, "Performance Optimization of a Fuzzy Entropy based Feature Selection and Classification Framework", *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, IEEE, pp.1361-1367 (2018).

2. **Zixiao Shen**, Xin Chen and Jonathan M. Garibaldi, "A Novel Weighted Combination Method for Feature Selection using Fuzzy Sets", *2019 IEEE International Conference on Fuzzy Systems(FUZZ-IEEE)*, IEEE, pp. 1-6 (2019).

3. **Zixiao Shen**, Xin Chen and Jonathan M. Garibaldi, "A Novel Meta Learning Framework for Feature Selection using Data Synthesis and Fuzzy Similarity", *2020 IEEE World Congress on Computational Intelligence (WCCI)*, IEEE, pp. 1-8 (2020).

4. **Zixiao Shen**, Xin Chen and Jonathan M. Garibaldi, "A Fuzzy Aggregation based Ensemble Framework for Accurate and Stable Feature Selection", *2021 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, IEEE, pp.1-7 (2021).

5. **Zixiao Shen**, Xin Chen and Jonathan M. Garibaldi, "New Accuracy and Robustness Measures for Ranking-based Feature Selection Methods", *In preparation for IEEE Transactions on Knowledge and Data Engineering*, (2022).

# Chapter 2

# Literature Review

This chapter presents a literature review of the research in this thesis. Firstly, Section 2.1 provides a comprehensive review of FS methods. Secondly, Section 2.2 discusses some basic concepts and introduces the background knowledge of fuzzy theory, including fuzzy sets, fuzzy entropy, fuzzy similarity, and fuzzy systems. Last, Section 2.3 and Section 2.4 introduce the ensemble learning and meta-learning, respectively.

## 2.1 Feature Selection Methods

FS's main aim is to extract a minimal feature subset from a problem domain while retaining a high accuracy to represent the original data [38]. It acts as one of the most widely used and essential techniques in the data pre-processing stage and an indispensable component in the machine learning area [102]. It is also called variable selection, attribute selection, or feature subset selection in different areas such as machine learning, pattern recognition, signal processing, etc.

The central premise of applying an FS technique is that many features in the original data are either irrelevant or redundant, which can be removed without the loss of much information [172]. Irrelevant features provide no helpful information, and redundant fea-

tures provide no additional information than the currently selected ones. In comparison, relevant features are neither irrelevant nor redundant to the target concepts. Nevertheless, the information about relevant features is normally an unknown knowledge in many practical scenarios. Therefore, different techniques are developed to explore the candidate features to better represent the domain knowledge [97, 119].

In general, the application of FS methods can provide many benefits in practical applications. First, the correct selection of relevant features can help provide a good insight into the underlying concept of the problem and improve the overall predictive performance [38]. Second, eliminating irrelevant and redundant features can drastically reduce the running time and storage requirements for the subsequent learning algorithms. Third, FS methods can also help better understand the data, increase the comprehensibility of the learning results and gain knowledge from the visualization process.

### 2.1.1 Definitions

FS refers to the problem of selecting the most relevant features to the target event for a given dataset. Unlike other dimensionality reduction approaches, FS methods can preserve and maintain the original meaning of features after the selection process [128]. Many researchers define FS methods [38] from various aspects. By Kira and Rendell's definition, FS aims to find the minimal-sized feature subset, which is necessary and sufficient to infer the target event [107]. From Narendra and Fukunaga's viewpoint, FS aims to select a feature subset such that the value of a criterion function is optimized over all subsets [147]. From Koller and Sahami's perspective, by choosing a suitable feature subset, FS can be used to improve the prediction accuracy and decrease the data structure's complexity and size [112].

A *feature* is defined as 'relevant' when it is highly correlated with the target task. Otherwise, it may be defined as 'irrelevant' when uncorrelated with the target task. Redundant features refer to those with no more information than the currently selected features, and

irrelevant features provide no helpful information in any context. The existence of redundant and irrelevant features can misguide decision-making results to a certain degree [172]. In this case, FS methods are designed to identify and remove the irrelevant and redundant features that do not contribute to or reduce the models' performance. Feature relevance has acted as an independent and essential measurement, which needs to be defined appropriately. This section gives the definitions of FS methods and feature relevance below.

**1. Definition of feature selection** Given a dataset $\mathbb{D}$ that is composed of $S$ samples and $N$ features, the numbers of features in the original feature set $\mathbb{F}$ and the reduced feature subset $\mathbb{F}'$ are represented as $|\mathbb{F}| = n, |\mathbb{F}'| = m, (n \geq m)$. Let $L(.)$ be an evaluation criterion to be maximized and defined as $L : \mathbb{F}' \subseteq \mathbb{F} \rightarrow \Re$. Therefore, the candidate feature subset can be constituted with any of the following conditions [119].

- Condition 1: Find the optimal feature subsets $\mathbb{F}'$ to maximize $L(\mathbb{F}')$ where $m < n$ and $\mathbb{F}' \subset \mathbb{F}$.
- Condition 2: Set a threshold $\theta$ and then find a feature subset with the smallest feature number ($m$) which meets the condition $L(\mathbb{F}') > \theta$.

The choice of the different performance evaluation function $L(.)$ can result in the various selected feature subsets. Among them, one of the most widely used evaluation metrics is given below.

**2. Definition of relevance** Investigating the relationship between superior features and relevance becomes an important issue. There are different definitions for features to be 'relevant' in the machine learning area. Among them, one of the widely used definitions refers to the notion of being 'relevant' to target concepts [14].

**3. Relevant to the target task** A feature $F$ is relevant to the target task when there is a pair of examples $a$ and $b$ in the sample space such that $a$ and $b$ differ only in their assignment

to the feature $F$ and $C(a) \neq C(b)$. For the real-valued data, the feature $F$ is relevant when there are some examples in the sample space for which the changes in the $F$ value affect the decision making results [14].

The notions of relevance are helpful from the viewpoint of deciding which features to be kept or ignored. Relevant features are normally essential to retain. Otherwise, their elimination may give rise to an increase in the ambiguity of data. However, the definition of relevance is independent of any specific learning algorithm. The selection of the relevant features may not necessarily be beneficial for the downstream algorithm [14].

### 2.1.2 General Procedures and Approaches

In the ideal cases, all the candidate feature subsets can be searched and evaluated to locate the absolute best one for the given training dataset. However, the ideal solution to search all the candidate feature subsets proves to be ordinarily exhaustive, costly, and practically prohibitive, even in medium-sized datasets. For those issues, some sub-optimal search strategies are applied in the FS process, shown as follows.

**1. Framework of the main steps** There are four main steps within the FS framework in the literature, which include subset generation, subset evaluation, stopping criterion, and result validation. The overall framework of the main steps is illustrated in Fig. 2.1.

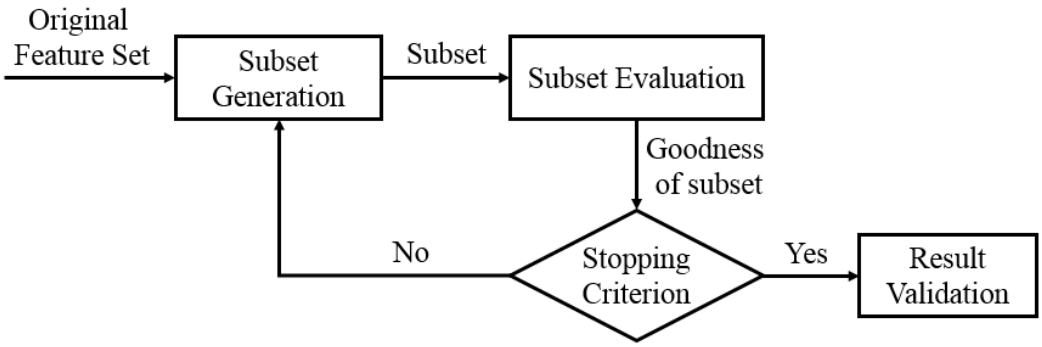



Fig. 2.1 Main steps and framework of feature selection

At first, feature subsets are selected using various search strategies in the subset generation process [135]. Then their performance is compared with the previous best one using specific evaluation metrics. Hence, the optimal feature subset is continuously renewed by the subset with better performance. The process will be repeated until a predefined stopping criterion is satisfied [38, 119]. To be specific, the four main steps are listed below.

1. **Subset Generation:** Generate the next candidate feature subset from the original one.
2. **Subset Evaluation:** Evaluate and score the performance of the feature subsets using the certain criterion.
3. **Stopping Criterion:** Set up the threshold to stop the loop based on the evaluation function. Some widely used stopping criteria include:
   (a) Some prerequisites are met, such as maximum iteration number, the minimum number of features, minimum classification error rate, etc.
   (b) The deletion or addition of the features in the feature subsets produces little difference in the performance.
   (c) The search process is completed.
4. **Result Validation:** Validate the effectiveness of selected feature subsets using the downstream decision-making methods.

Besides, the general procedures for FS can be illustrated in Algorithm 1. There exist plenty of FS methods in the literature. Those methods are grouped into the categories shown below based on different principles.

**2. Supervised, unsupervised and semi-supervised approaches** FS methods can be generally categorized as supervised, unsupervised, and semi-supervised approaches based on the availability of class labels. Supervised FS methods use labelled data for FS and evaluate feature relevance by measuring the correlation between feature values and the class labels. Therefore, the class label of each sample is given beforehand. Hence, it

**Algorithm 1** General algorithm for FS [119]

**Input:**
1: $\mathbb{F}$: Feature set of the dataset
2: $SG$: Successor Generation Operator
3: $E$: Evaluation measurement
4: $\theta$: Stopping Criteria
**Output:** $\mathbb{F}_{opt}$: Optimal feature set
5: **initialize**:
6: $\mathbb{F}'$:= Start_point($\mathbb{F}$)
7: $\mathbb{F}_{opt}$:= Best of $\mathbb{F}'$ using $E$
8: **repeat**:
9: $\mathbb{F}'$:= Search_Strategy($F', SG(E), X$)
10: $\mathbb{F}_{opt}$:= Best of $\mathbb{F}'$ according to $E$
11: **If** $E(\mathbb{F}') \geq E(\mathbb{F}_{opt})$ or $\{E(\mathbb{F}') == E(\mathbb{F}_{opt})$ and $|\mathbb{F}'| < |\mathbb{F}_{opt}|\}$
12: $\mathbb{F}_{opt} = \mathbb{F}'$
13: **until** Stop criteria are satisfied

becomes natural to include the features related to the classes and exclude those that are not. For example, correlation-based FS (CFS) relies on an underlying hypothesis that *"a good feature subset contains features highly correlated with the class, yet uncorrelated with each other"* [73]. Besides, the corresponding predictive accuracy can also guide the FS process as an independent evaluation index.

Unsupervised FS methods evaluate the feature relevance by the capability of keeping particular properties of the data, such as the variance or the preservation of locality [233]. The class labels are not given in advance. Therefore, without any guidance of class labels for relevant information, FS in unsupervised scenarios proved to be a much harder problem [172]. This approach searches for feature subsets with reduced size to uncover the interesting natural groups from the data based on a chosen criterion [50]. In real-world applications, there exist many unlabelled and small labelled datasets. To deal with the "small labelled sample problem", semi-supervised FS methods are developed to use both labelled and unlabelled data during the FS process.

**3. Ranking based and subset based approaches** According to the implemented internal procedures and results, FS methods can be grouped as ranking-based and subset-based approaches. The ranking based approach ranks the features using the degree of relevance. Many FS algorithms have employed it as a principal or auxiliary selection mechanism because of its simplicity, scalability, and empirical success. A typical ranking-based filter FS method is composed of two steps. Firstly, the features are ranked using some feature evaluation criteria. Secondly, the features with lower ranks are removed based on a predefined threshold. The importance evaluation procedures can be either univariate or multivariate. The univariate scheme ranks each feature individually regardless of other features. In contrast, the multivariate method ranks multiple features in a batch [128]. However, in some cases, a completely irrelevant feature may still significantly improve the performance in the combination process [68].

The subset evaluation FS approach constructs the candidate feature subsets directly using various search strategies, such as exhaustive search, quick reduct, etc [90]. The exhaustive search constructs a way of systematically listing all potential solutions to the problem, proving brute force to the feature subset selection process. Besides, some other algorithms attempt to calculate a reduced-sized feature set without exhaustively generating all possible subsets. For example, the QUICKREDUCT algorithm starts with an empty set and adds the features, resulting in the most significant increase in the rough set dependency metric [32]. However, those methods are not guaranteed to find a minimal and optimal subset. They usually result in a close-to-minimal subset, which is still helpful in significantly reducing the datasets' dimensionality [90].

**4. Wrapper, embedded, and filter approaches** Based on the dependency with learning algorithms, FS methods are roughly classified into three types, i.e., wrapper, embedded, and filter methods [30]. The brief introduction for those approaches is shown as follows.

• **Wrapper FS Approach** Wrapper FS approaches can be formalized as a combinatorial optimization problem. The FS methods maximize the learned hypothesis's quality by feeding back the decision-making outcome to seek the optimal feature subset. By incorporating the classification performance of the predefined learning algorithm, the wrapper FS methods explore the relationship and search for an optimal feature subset [110]. Different wrapper algorithms are generated through the various subset generation and evaluation metrics. Through adding or removing the features within the feature subsets, multiple methods are used to find the optimal combination and maximize the model's performance [118]. However, the algorithms' search space becomes enormous when the number of features increases significantly in high-dimensional datasets. It may also make the FS methods impractical in real applications [128]. Besides, the subsets produced by the wrapper approaches are explicitly tied to the learning algorithm, which may not be useful and applicable in a general sense [110].

• **Embedded FS Approach** The embedded approach integrates the FS process with learning phases at the same time [136]. In the embedded methods, the FS process is integrated as part of the learning algorithm. Firstly, the embedded methods train a machine learning model. Then they derive the feature importance from this model, which measures how much a feature is essential when making a prediction. Finally, the methods remove non-important features using the derived feature importance. Compared with wrapper FS methods, the embedded approach interacts with the learning algorithms at a lower computational cost. It provides a trade-off solution between filter and wrapper approaches. The feature dependencies are captured by considering the relationship between the input and output features and searching locally for features with better discrimination. Some famous examples include ridge regression [83], lasso [203] and random forest [132].

• **Filter FS Approach** Unlike the previous two FS approaches, filter FS methods are independent of any inductive learning algorithm. There are generally two classes of filter FS methods: ranking-based and search-based. A measure is incorporated in the ranking-based filter FS method to evaluate the feature subsets and filter out irrelevant features, such as heuristic scores. Features are therefore ranked based on their significance and quality. The feature ranking step is widely applied as the essence of the FS process, while the number of chosen features is specified by the users or analytically determined [196]. The process becomes quite computationally efficient, especially in the case of very high dimensional datasets [172].

On the other hand, the search-based filter methods perform a search through the space of feature subsets, such as Correlation-based FS. The exhaustive search is usually intractable, especially when the size of feature space becomes exponential with the increase of the number of features. Therefore, various methods have been used to find the suitable feature subsets, such as greedy hill-climbing search strategies [108], best-first search, beam search [170], etc.

However, the filter-based FS methods still face many disadvantages. Firstly, the ranking-based filter methods are not good at considering and investigating the dependencies between multiple features compared to the embedded and wrapper methods. Different features are evaluated and scored individually and separately by many filter-based FS methods. Secondly, the selected features may not be suitable for the target learning algorithms due to lacking a specific learning algorithm that guides the FS phase [128]. The following section briefly introduces the filter-based FS method to help form a comprehensive perspective.

### 2.1.3 Filter-based Feature Selection Methods

In general, typical filter-based methods for feature ranking consist of two steps: first, rank the features based on the specific evaluation criteria; second, select the features with higher scores than a predefined threshold [200]. In this case, the feature ranking procedure becomes

essential within the FS approach and can be used to guide the further machine learning process [163]. Therefore, the ranking-based FS methods gradually emerge as an active and growing research topic in the machine learning area. The rationale behind this phenomenon lies in that many machine learning issues are ranking problems by nature [172]. There are plenty of different kinds of filter FS in the literature. From a methodological point of view, filter FS methods are generally categorized into four groups, which include similarity-based, information-based, statistical-based, and graph-based approaches [129].

**1. Similarity-based approach** Similarity-based FS methods assess features' importance by measuring their ability to preserve data similarity. Data similarity derives from label information on supervised FS and distance metric measures on unsupervised FS. An affinity matrix is built at first; afterwards, the feature scores are obtained. The methods achieve excellent performance in both supervised and unsupervised scenarios. However, on account that each feature is evaluated individually, the methods cannot handle features' redundancy [129]. Some examples are shown below.

• **Laplacian Score** Laplacian Score FS method selects features based on their locality preserving power in an unsupervised scenario. The intrinsic principle depends on the observation that local geometric structure is crucial for discrimination, which means that two data points are probably related to the same topic when close to each other. The framework of this method is fundamentally based on Laplacian Eigenmaps and Locality Preserving Projection processes [81].

• **Fisher Score** Fisher score-based FS method is a supervised FS algorithm. It selects features based on the property that the feature values from the same class are similar, and those from different classes are dissimilar to each other [48]. For a given dataset, the fisher score of the $i$th feature $F^i$ is calculated in Equation 2.1.

$$Fisher\_Score(F^i) = \frac{\sum_{j=1}^{C} S_j(\mu_j^i - \mu^i)^2}{\sum_{j=1}^{C} S_j \sigma_{ij}^2} \tag{2.1}$$

where $C$ represents the number of classes; $S_j$ indicates the number of samples in the $j$th class $C_j$; $\mu^i$ represents the mean value of feature $F^i$; $\mu_j^i$ indicates the mean value of feature $F^i$ in class $C_j$; $\sigma_{ij}^2$ stands for the variances of feature $F^i$ for the samples in class $C_j$ [129].

• **ReliefF** ReliefF is an extension of the Relief method, which is not limited to two-class problems. It is pretty robust and can deal with incomplete and noisy data. The Relief and ReliefF FS methods are iterative, randomized, and supervised FS approaches that estimate the features' dependencies in a unified view. Their core concept is to evaluate the features' quality based on samples' distinction. For a randomly selected sample, the algorithm searches for its $K$ nearest neighbours from the same class (nearest hits) and the different classes (nearest misses), respectively. Therefore, the feature score is updated based on the corresponding feature values, nearest hits, and nearest misses. The conditional probabilities are also approximated using the relative frequencies [171]. The pseudo-code for the original Relief algorithm is shown below.

**Algorithm 2** The Relief algorithm

**Input:** A vector of feature values and the class value for each training sample;
1: $n$: number of training samples;
2: $\alpha$: number of features;
3: $m$: number of random training samples out of $n$ used to update $W$;
**Output:** The vector $W$ of feature scores that estimate the quality of features.
4: **Begin:**
5: Set all weights $W[A] = 0$;
6: **For** $i = 1$ **to** $m$ **do**
7: randomly select a 'target' sample $R_i$;
8: find the nearest hit $H$ and nearest miss $M$ samples;
9: **For** $A = 1$ **to** $\alpha$ **do**
10: $W[A] = W[A] - diff(A,R_i,H)/m + diff(A,R_i,M)/m$
11: **End**

An essential idea of the original Relief algorithm is to estimate the quality of features based on how well their values distinguish between the samples that are close to each other. For this purpose, given a randomly selected instance $R_i$, the Relief method searches for its two nearest neighbours: one from the same class, which is called *nearest hit* $H$, and the other from a different class, called *nearest miss* $M$. The quality estimation $W[A]$ is updated for all features $A$ depending on their values of $R_i$, $M$, and $H$. This process is repeated for $m$ times, where $m$ is a user-defined parameter.

**2. Statistical-based approach** The statistical-based FS approach assesses the features' relevance using different statistical measures, which is simple and straightforward. It analyzes the features individually and filters out the unwanted ones. It is very computationally efficient and can be widely applied. However, the separate and independent feature evaluation process may ignore the features' redundancy inevitably [129]. Some representative FS algorithms are introduced below.

• **T-Score** T-Score based FS approach is used for binary classification problems, which computes the ratio between the mean and variance difference of two classes. The importance of a given feature is defined by how much it differs from the means of the two classes statistically [39]. The high values of the $t$-score indicate the corresponding features are more important. The $t$ score value of feature $F$ is calculated as:

$$T\_Score(F) = |\mu_1 - \mu_2| / \sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}} \tag{2.2}$$

where $\mu_i$, $\sigma_i$ and $n_i$ $(i = 1, 2)$ represent the mean values, standard deviation and numbers of samples of two different classes respectively.

• **Gini Index Feature Selection (GIFS)** Gini index, or called Gini impurity, measures the inequality of a frequency distribution, which is a statistical dispersion [61]. Gini index-

based FS approach quantifies the features' ability to separate the samples from different classes. Given a feature $F$, which separates the dataset into $\mathbb{W}$ and $\widehat{\mathbb{W}}$, the Gini index score of that feature is calculated below [129].

$$GI\_Score(F) = \min_{\mathbb{W}} \left( p(\mathbb{W})(1 - \sum_{s=1}^{C} p(C_s|\mathbb{W})^2) + p(\widehat{\mathbb{W}})(1 - \sum_{s=1}^{C} p(C_s|\widehat{\mathbb{W}})^2) \right) \tag{2.3}$$

where $p(.)$ and $p(C_s|\mathbb{W})$ represent the probability and conditional probability on class $s$ of $\mathbb{W}$ respectively. Lower Gini index values indicate more important and relevant features.

• **Correlation based Feature Selection (CFS)** CFS method couples the evaluation formula with an appropriate correlation measurement and a heuristic search strategy. The central hypothesis is that good feature sets contain highly correlated features with the class, yet uncorrelated with each other [73]. A correlation-based heuristic method to evaluate the worth of a feature subset $\mathbb{S}$ is calculated in Equation 2.4.

$$CFS\_Score(\mathbb{S}) = \frac{k\overline{R_{CF}}}{\sqrt{k + k(k-1)\overline{R_{FF}}}} \tag{2.4}$$

where $\overline{R_{CF}}$ and $\overline{R_{FF}}$ stand for the mean feature class correlation and average feature-feature correlation respectively. CFS score represents the heuristic merit of the feature subset $\mathbb{S}$ with $k$ features [129].

**3. Information-based approach** The information-based filter FS approach maximize the feature relevance and minimize feature redundancy by exploiting different hand-designed information criteria [48]. The relevance and redundancy of the features are considered in a probabilistic framework. Considering that the features' relevance is typically measured based on the class labels, most existing information-based FS methods only work in a

supervised scenario [129]. Some basic concepts and representative algorithms are introduced as follows [26].

1. The entropy value of a random feature $X$ is defined in Equation 2.5.

$$H(X) = -\sum_{x \in X} P(x) log(P(x)) \tag{2.5}$$

   where $x$ and $P(x)$ represent the specific value and its probability among all possible values of $X$.

2. The conditional entropy of $X$ is calculated given the label information $Y$.

$$H(X|Y) = -\sum_{y \in Y} P(y) \sum_{x \in X} P(x|y) log(P(x|y)) \tag{2.6}$$

   where $P(y)$ and $P(x|y)$ represent the priori probability of $y$ and the conditional probability of $x$ given $y$ respectively.

3. The mutual information between $X$ and $Y$ is calculated in Equation 2.7.

$$I(X;Y) = H(X) - H(X|Y) = \sum_{x \in X} \sum_{y \in Y} P(x,y) log \frac{P(x,y)}{P(x)P(y)} \tag{2.7}$$

   where $P(x,y)$ represents the joint probability of $x$ and $y$, the mutual information is symmetric and will shrink to zero when the features are independent.

• **Mutual Information Feature Selection (MIFS)** MIFS method considers both feature relevance and redundancy in the FS phase [9]. The feature scores of the new unselected feature $X_k$ are formulated in Equation 2.8.

$$J_{MIFS}(X_k) = I(X_k;Y) - \beta \sum_{X \in S} I(X_k;X) \tag{2.8}$$

It is based on the rationale that good features are strongly correlated with class labels and highly uncorrelated with each other. In Equation 2.8, the first term $I(X_k;Y)$ evaluates the feature relevance with class labels; the second term measures the mutual information among the currently selected features. The feature redundancy is minimized by penalizing the features with a high mutual information [129]. The overlapped information between the candidate and existing features is regulated using a proportional term $\beta$.

• **Minimum Redundancy Maximum Relevance (MRMR)** From Equation 2.8, the MRMR method is proposed by setting the $\beta$ value as the reverse value of the number of selected features [161].

$$J_{MRMR}(X_k) = I(X_k;Y) - \frac{1}{|S|}\sum_{X\in S} I(X_k;X) \tag{2.9}$$

In Equation 2.9, the feature redundancy is gradually reduced when the number of selected features increases. The increasing number of the selected non-redundant features makes it much difficult to add a new non-redundant one. Besides, pairwise independence among features becomes robust when more features are added inside [26].

• **Fast Correlation Based Filter (FCBF)** Unlike the previous approaches by the unified conditional likelihood maximization framework, FCBF exploits the feature-class and feature-feature correlation simultaneously. With a predefined threshold $\delta$, feature subset $\mathbb{S}$ with high correlation of class labels is firstly selected with $SU \geq \delta$, while $SU$ is short for the symmetric uncertainty. The symmetric uncertainty is used to compensate for the information gain's bias toward features with more values. Then it is normalized between 0 and 1, while 1 indicates the complete prediction and 0 stands for independence. The symmetric uncertainty $SU$ between the features $\mathbb{S}$ ($X_S \in \mathbb{S}$) and the class label $Y$ is calculated below [226].

$$SU(X_S, Y) = 2\frac{I(X_S;Y)}{H(X_S)+H(Y)} \tag{2.10}$$

Feature $X_j$ is redundant with feature $X_k$ when $SU(X_j, X_k) \geq SU(X_k, Y)$. The redundant feature set is denoted as $\mathbb{S}_R$ and further split into $\mathbb{S}_R^+$ and $\mathbb{S}_R^-$, which represent the redundant features are contained to feature $X_k$ with $SU(X_j, X_k) > SU(X_k, Y)$ and $SU(X_j, X_k) < SU(X_k, Y)$ respectively. Various heuristic algorithms are implemented on $S_R$ to remove the redundant features and retain the most relevant ones with the class labels [129].

**4. Graph-based approach** The graph-based FS approach builds an undirected fully-connected graph to represent the feature distributions and relationships. The structure within the features is taken explicitly as prior knowledge and fed into the FS process. It holds the advantage of handling both feature relevance and feature redundancy. However, the mechanism within the methods leads to high computational costs. The model's complexity also makes it challenging to infer the data structures for FS purpose [237]. Some typical examples of this approach are shown below.

• **Infinite Feature Selection (IFS)** IFS is an unsupervised graph-based FS method, which exploits the matrices' power series' convergence properties. The method measures the features' importance by considering the selection of features as a path with an infinite number. Firstly, an undirected fully-connected graph $G = (V, E)$ with the given feature set distributions are built. Nodes $V$ represent the feature distributions. Edges $E$ codify their independence. Feature subsets are described using the paths in the graphs. Secondly, the energy of the path is calculated using the edges' product. The paths with higher energy mean a higher score and more independent nodes. Thirdly, the feature score of every single feature is computed to represent the features' importance or independence [175].

• **Eigenvector Centrality Feature Selection (ECFS)** ECFS method identifies the features' importance into an arbitrary set of cues and ranks the features afterwards. By mapping the FS problem into an affinity graph using nodes (on behalf of features), features' importance is measured by indicators of centrality such as eigenvector centrality. Firstly, the weighted edges are defined using an undirected graph $G = (V, E)$ associated with adjacency matrix $A$. Secondly, the graph is weighted based on some reasonable criteria which are related to class separation. The design of the $\varphi$ function is an essential operation in this step. It is handcrafted or automatically learned from data and used to weight the graph using different heuristics. Thirdly, the nodes that correspond to the individual eigenvector centrality within graphs are identified and scored. The rationale behind the eigenvector centrality lies in that a node or feature proves to be important when it is linked to other important ones with high scores [173].

• **Infinite Latent Feature Selection (ILFS)** ILFS is a robust probabilistic latent graph-based FS method. It considers all the possible feature subsets as paths on a graph and bypasses the combinatorial problem analytically. FS is then mapped as an affinity graph by considering the feature subset to connect the set of nodes. One unique characteristic of this approach is that a given feature's importance is modelled to be a conditional probability of a latent feature. Therefore, the method can model the important hidden feature behind data, which is relevant as an abstract latent feature [174].

### 2.1.4 Other Feature Selection Methods

There are many other well-known and widely used FS methods in the literature, such as evolutionary computation, fuzzy-based methods, etc. A general introduction is provided for those FS methods as below.

**1. Evolutionary based FS methods** When considering feature interactions, it could make an individual relevant feature redundant or a weakly relevant feature highly correlated to the decision-making outcome, which is undesirable. The evolutionary-based algorithm is proposed to address these issues, which is a generic optimization technique that mimics the ideas of natural evolution. Compared with the traditional search methods, their population-based mechanism can produce multiple solutions in a single run and is particularly suitable for multi-objective problems. In general, there are three basic steps. Firstly, parents create offspring (crossover). Secondly, the individuals undergo some minor changes (mutation). Thirdly, the likelihood of survival is higher for fitter individuals (selection). One of the famous examples is the genetic algorithm, which is a metaheuristic inspired by the process of natural selection [214]. On the other hand, the evolutionary computation holds a significant limitation on requiring a relatively high computational cost due to the involvement of a large number of evaluations [222].

**2. Fuzzy based FS methods** Practical datasets are typically imperfect with some text-based information, noise and incomplete features. Fuzzy methods are designed to model the vagueness, imprecision, and uncertainty [8]. It becomes a natural and straightforward process to overcome the real datasets' practical problems and integrate fuzzy methods with the FS process. Therefore, various fuzzy methods are applied to solve the FS problems. In the literature, many widely used fuzzy-based FS methods are proposed from different technical perspectives. In 1999, the concept of fuzzy feature selection was proposed by Rezaee *et al.* to describe the method which selects the optimal fuzzy sets automatically using conventional search techniques and a representative labelled data set. The optimal subset of fuzzy features is determined after projecting the original dataset onto a fuzzy space [169]. In 2002, a fuzzy neural network-based method was proposed by Li *et al.* for pattern classification and FS. The proposed neural network attempts to select the essential features from the original ones and maintain the maximum recognition rate [130]. In 2004, Jensen and Shen introduced

a semantics-preserving dimensionality reduction method based on rough and fuzzy rough sets [93, 94]. This method has already been widely used in many applications, such as web categorization [92].

Besides, many FS methods mentioned above need user-supplied information in the algorithm, such as the number of features to be selected, the threshold to terminate the algorithm, etc. Jensen and Shen [90] have developed some techniques to reduce the dimensionality and preserve the features' meaning using the data alone without any additional information, which are introduced below.

• **Rough set attribute reduction (RSAR)** Rough set theory acts as an extension of the conventional set theory, which is generally defined using topological operations, i.e., approximations [159]. It is used to discover the data dependencies and reduce the number of features only using the information within the datasets [91]. Given a dataset with discretized feature values, RSAR aims to find the original features' most informative subset. By using the granularity structure of data, no additional parameters are needed. However, the RSAR method still faces the restrictive requirement that all data is discrete and the limitation of handling noisy data [90].

• **Fuzzy-rough set based feature selection (FRFS)** In order to overcome the main limitation of RSAR, which can only operate effectively on the discrete-valued datasets, the FRFS method has been developed with the use of fuzzy-rough sets and the new measure of feature significance, which is the fuzzy-rough degree of dependency [94]. By utilizing fuzzy-rough sets to handle the real-valued features via fuzzy tolerance relations instead of crisp equivalence, the FRFS method achieves better noise and uncertainty handling performance. Therefore, it can be used to reduce the dimensionality of the discrete, real-valued noisy data, or a mixture of both without any user-supplied information [93].

## 2.2 Fuzzy Theory

Zadeh proposed fuzzy theory in 1965 based on the mathematical theories of fuzzy logic and fuzzy sets [228]. By introducing the membership functions in the fuzzy IF-THEN rules, the fuzzy theory extends the true or false boolean logic, provides valuable flexibility for reasoning, and quantifies the inaccuracies and uncertainties [41]. With the capability to interpret the vagueness of information, the fuzzy theory is then widely applied to represent human knowledge and model data's ambiguity [126]. Therefore, some background knowledge has been introduced on fuzzy theory as follows.

### 2.2.1 Fuzzy Sets

Fuzzy mathematics provides the fundamental languages and theoretical basis in fuzzy theory. The principle of fuzzy mathematics is to generalize crisp sets in classical mathematical theory using fuzzy sets [211]. In specific applications, all the possible elements are depicted as the universe of discourse, or universe set $\mathbb{U}$. Therefore, the definition of fuzzy sets is given below.

**1. Definition of fuzzy sets** A *fuzzy set* within a universe set $\mathbb{U}$ is characterized using a membership function $\mu(x)$ with the values between 0 and 1. For instance, a fuzzy set $A$ in $\mathbb{U}$ is represented as a set of ordered pairs of a generic element $x$, and the membership value $\mu_A(x)$ [211].

$$A = \{(x, \mu_A(x)) | x \in \mathbb{U}\} \tag{2.11}$$

When $\mathbb{U}$ is continuous, $A$ is named *continuous fuzzy set*, as shown in Equation 2.12.

$$A = \int_{\mathbb{U}} \mu_A(x)/x \tag{2.12}$$

When $\mathbb{U}$ is discrete, $A$ is named *discrete fuzzy set*, as shown in Equation 2.13.

$$A = \sum_{\mathbb{U}} \mu_A(x)/x \tag{2.13}$$

In Equation 2.12 and 2.13, the integral and summation signs neither indicate integration nor arithmetic addition operators. Those operators represent the collection of all points $x \in \mathbb{U}$ in the membership function $\mu_A(x)$.

**2. Basic concepts and operations** Some basic concepts and terminologies in fuzzy sets are introduced as below [211].

- *Support:* The support of a fuzzy set $A$ in the universe set $\mathbb{U}$ is a set which contains all the elements of $\mathbb{U}$ with nonzero membership values in $A$.

$$supp(A) = \{x \in U | \mu_A(x) > 0\} \tag{2.14}$$

- *Fuzzy Singleton:* A fuzzy set whose support is a single value in $\mathbb{U}$.
- *Height:* Largest membership value in the fuzzy set.
- *Normal Fuzzy Set:* A fuzzy set with the height equals one.

Assuming that $A$ and $B$ are fuzzy sets in the same universe set $\mathbb{U}$, some basic fuzzy operations are given below.

- *Equal:* $A$ and $B$ are equal if and only if $\mu_A(x) = \mu_B(x)$ for all $x \in \mathbb{U}$.
- *Contain:* $B$ contains $A$, denoted by $A \subseteq B$, if and only if $\mu_A(x) \leq \mu_B(x)$ for all $x \in \mathbb{U}$.
- *Complement:* Complement of $A$ is a fuzzy set $\bar{A}$ in $\mathbb{U}$ with the membership function defined in Equation 2.15.

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \tag{2.15}$$

- *Union:* Union of $A$ and $B$ is a fuzzy set in $\mathbb{U}$, denoted by $A \cup B$ with the membership function in Equation 2.16.

$$\mu_{A\cup B}(x) = max[\mu_A(x), \mu_B(x)] \tag{2.16}$$

- *Intersection:* Intersection of $A$ and $B$ is a fuzzy set $A \cap B$ in $\mathbb{U}$ with the membership function in Equation 2.17.

$$\mu_{A\cap B}(x) = min[\mu_A(x), \mu_B(x)] \tag{2.17}$$

In Equation 2.16, the fuzzy set $A \cup B$ is defined as the smallest fuzzy set containing both $A$ and $B$. The fuzzy set $A \cap B$ defined by Equation 2.17 shows the largest fuzzy set contained by both $A$ and $B$. However, the definition above is only one type of operation on fuzzy sets. Many other possible solutions still exist in the literature for practical applications. Therefore, the following subsections introduce other types of operations on fuzzy sets.

**3. Further operations on fuzzy sets**

• ***S*-norms** *S*-norm is a kind of generalized form of fuzzy union, which maps the membership functions of fuzzy sets $A$ and $B$ into the union of $A$ and $B$ [211].

$$s[\mu_A(x), \mu_B(x)] = \mu_{A\cup B}(x) \tag{2.18}$$

Given that the function $s$ is a union operator, $a$, $b$, and $c$ represent the membership function's values, the following requirements are satisfied.

1. *Boundary Condition:* $s(1,1) = 1, s(0,a) = s(a,0) = a$
2. *Commutative Condition:* $s(a,b) = s(b,a)$
3. *Nondecreasing Condition:* $b \leq c \Rightarrow s(a,b) \leq s(a,c)$
4. *Associative Condition:* $s(s(a,b),c) = s(a,s(b,c))$

The boundary condition represents the extreme cases of a union function. The commutative condition ensures the order insensitivity character of fuzzy sets during the combination process. The non-decreasing condition leads to the monotone increasing characteristics, where an increase in membership values of the two fuzzy sets should lead to an increase in the membership value of their union. The associative condition ensures the extension ability within different fuzzy sets. Some examples of $s$-norms are illustrated as below [211].

- *Drastic sum*

$$s_{ds}(a,b) = \begin{cases} a & if\ b = 0 \\ b & if\ a = 0 \\ 1 & otherwise \end{cases} \tag{2.19}$$

- *Algebraic sum*

$$s_{as}(a,b) = a + b - ab \tag{2.20}$$

- *Einstein sum*

$$s_{es}(a,b) = \frac{a+b}{1+ab} \tag{2.21}$$

- *Dombi class* [44]

$$s_{\lambda}(a,b) = \frac{1}{1+[(\frac{1}{a}-1)^{-\lambda}+(\frac{1}{b}-1)^{-\lambda}]^{-1/\lambda}}, \quad \lambda \in (0,\infty) \tag{2.22}$$

- *Yager class* [223]

$$s_{\omega}(a,b) = min[1,(a^{\omega}+b^{\omega})^{1/\omega}], \quad \omega \in (0,\infty) \tag{2.23}$$

Different choices of the parameters $\lambda$ and $\omega$ in the Dombi and Yager class define a specific $s$-norm.

• ***T*-norms** Unlike *s*-norms, *t*-norms represent the binary operation to generalize the intersection by transforming the membership functions of fuzzy sets $A$ and $B$ into the intersection of $A$ and $B$.

$$t[\mu_A(x), \mu_B(x)] = \mu_{A\cap B}(x) \tag{2.24}$$

Given that the function $t$ is qualified as an intersection, the following requirements need to be met.

1. *Boundary Condition:* $t(0,0) = 0;\ t(a,1) = t(1,a) = a$
2. *Commutative Condition:* $t(a,b) = t(b,a)$
3. *Nondecreasing Condition:* $b \leq c \Rightarrow t(a,b) \leq t(a,c)$
4. *Associative Condition:* $t(t(a,b),c) = t(a,t(b,c))$

For the *s*-norms introduced beforehand, there is a corresponding *t*-norm. Based on the requirements above, the *t*-norms are shown below.

- *Drastic product*

$$t_{dp}(a,b) = \begin{cases} a & if\ b = 1 \\ b & if\ a = 1 \\ 0 & otherwise \end{cases} \tag{2.25}$$

- *Algebraic product*

$$t_{ap}(a,b) = ab \tag{2.26}$$

- *Einstein product*

$$t_{ep}(a,b) = \frac{ab}{2-(a+b-ab)} \tag{2.27}$$

- *Dombi class* [44]

$$t_\lambda(a,b) = \frac{1}{1+[(\frac{1}{a}-1)^\lambda + (\frac{1}{b}-1)^\lambda]^{1/\lambda}}, \quad \lambda \in (0,\infty) \tag{2.28}$$

- *Yager class* [223]

$$t_\omega(a,b) = 1 - min[1,((1-a)^\omega + (1-b)^\omega)^{1/\omega}], \quad \omega \in (0,\infty) \tag{2.29}$$

• **Averaging Operators** Given that $a$ and $b$ represent the values within the membership function, the $s$-norms and $t$-norms hold the following inequality functions [211].

$$\begin{aligned} max(a,b) \leq s(a,b) \leq s_{ds}(a,b) \\ t_{dp}(a,b) \leq t(a,b) \leq min(a,b) \end{aligned} \tag{2.30}$$

It is seen that the value range of $s$-norms and $t$-norms could not cover all the interval between $min(a,b)$ and $max(a,b)$. Therefore, the averaging operators are introduced to fill the gap between $min(a,b)$ and $max(a,b)$. The averaging operators map the value space of $[0,1]\times[0,1]$ into $[0,1]$, which is denoted by $v$. Some examples are listed as below.

- *Max-min averages*

$$v_\lambda(a,b) = \lambda max(a,b) + (1-\lambda)min(a,b), \quad \lambda \in [0,1] \tag{2.31}$$

- *Generalized means*

$$v_\alpha(a,b) = (\frac{a^\alpha + b^\alpha}{2})^{1/\alpha}, \quad \alpha \neq 0 \tag{2.32}$$

- *Fuzzy and*

$$v_p(a,b) = pmin(a,b) + \frac{(1-p)(a+b)}{2}, \quad p \in [0,1] \tag{2.33}$$

- *Fuzzy or*

$$v_\gamma(a,b) = \gamma max(a,b) + \frac{(1-\gamma)(a+b)}{2}, \quad \gamma \in [0,1] \tag{2.34}$$

The max-min averages and generalized means cover the whole interval between $min(a,b)$ and $max(a,b)$ by using different parameters. In addition, 'fuzzy and' and 'fuzzy or' cover

the value ranges $[min(a,b), \frac{a+b}{2}]$ and $[\frac{a+b}{2}, max(a,b)]$ respectively. In summary, the overall comparison among the fuzzy aggregation operators are shown in Fig. 2.2.

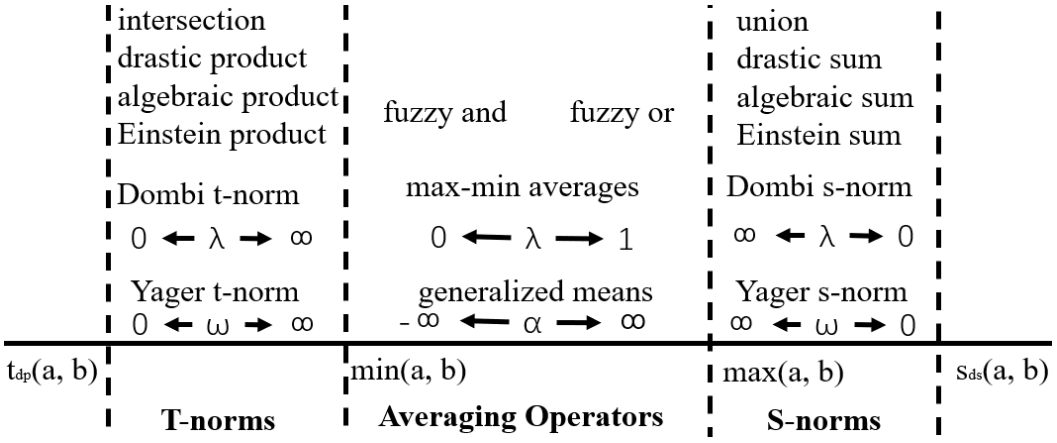



Fig. 2.2 Comparison of the fuzzy aggregation operators

### 2.2.2 Fuzzy Entropy

Based on Shannon's theory [186], information is normally evaluated through the measurement of uncertainty. On the other hand, fuzzy information is denoted to measure the fuzziness of fuzzy sets [115]. Fuzzy entropy represents the measure of fuzzy information from fuzzy sets or fuzzy systems. Different from the classical Shannon entropy, which utilizes the probability information, fuzzy entropy includes vagueness uncertainties and excludes any probabilistic concept [34].

Fuzzy entropy is defined using the membership function and customarily used to evaluate fuzzy sets' fuzziness. Through processing within a fuzzy structure, fuzzy entropy is widely applied in many areas [114]. In 1972, fuzzy entropy was mathematically defined by Luca and Termini [40] using Shannon's function. Four properties are introduced to design the new fuzzy entropy functions and gradually become a widely accepted criterion.

**1. Luca-Termini axioms** Given a fuzzy set $A$ defined in the universe set $\mathbb{U}$, the membership function is denoted as $\mu_A$. The axioms of a fuzzy entropy measure $H(A)$ of a fuzzy set $A$ are depicted below.

- *Axiom 1:* $H(A) = 0$ *iff* $A$ is a crisp set.
- *Axiom 2:* $H(A)$ is maximum *iff* $\mu_A(x) = 0.5, \forall x \in A$.
- *Axiom 3:* If $A$ is less fuzzy than $B$, then $H(A) \leq H(B)$.
- *Axiom 4:* $H(A) = H(\bar{A})$, where $\bar{A}$ denotes the complement of $A$.

**2. Different fuzzy entropy functions** Based on Luca-Termini Axioms, many fuzzy entropy functions are proposed in the literature. Given a fuzzy set $A$ with $N$ values, $\mu_i$ represents the $i$th membership values ($1 \leq i \leq N$); $P = \{p_1, p_2, ..., p_N\}$ indicates the probability distribution functions. Hence, different proposed fuzzy entropy functions are shown as below.

1. *Probabilistic Entropy (Zadeh's Entropy):* Probabilistic entropy depicts the uncertainty associated with a fuzzy event [228]. It is defined as a weighted Shannon entropy using the membership values below [229].

$$H_1(A) = -\sum_{i=1}^{N} \mu_i \times p_i \times \log p_i \tag{2.35}$$

2. *Non-Probabilistic Entropy (Luca's Entropy):* Non-probabilistic entropy measures the fuzziness and interprets the information as an extension of Shannon entropy. It is introduced by Luca *et al.* [40] and shown in Equation 2.36.

$$H_2(A) = -\frac{1}{N}\sum_{i=1}^{N}[(\mu_i \log \mu_i) + (1-\mu_i)\log(1-\mu_i)] \tag{2.36}$$

3. *Weighted Measures of Fuzzy Entropy (Parkash's Entropy):* Two weighted measures of fuzzy entropy, which are essentially the same, are proposed by Parkash *et al.* within the maximum entropy principle [155]. The entropy functions are shown below.

$$H_3(A;W) = \sum_{i=1}^{N} w_i[sin\frac{\pi\mu_i}{2} + sin\frac{\pi(1-\mu_i)}{2} - 1] \tag{2.37}$$

$$H_4(A;W) = \sum_{i=1}^{N} w_i[cos\frac{\pi\mu_i}{2} + cos\frac{\pi(1-\mu_i)}{2} - 1] \tag{2.38}$$

4. *Geometry of Fuzzy Entropy (Kosko's Entropy):* In 1986, Kosko defined the fuzzy entropy based on the hypercube geometry with the concepts of overlap and underlap [114], as shown in Equation 2.39.

$$H_5(A) = \frac{\sum_{i=1}^{N}(\mu_i \wedge \bar{\mu}_i)}{\sum_{i=1}^{N}(\mu_i \vee \bar{\mu}_i)} \tag{2.39}$$

where $\bar{\mu}_i$ represents the $i$th membership value of the complement of fuzzy set $A$.

### 2.2.3 Fuzzy Similarity

The similarity is normally defined as a generalized form of equivalence. The fuzzy similarity is a kind of fuzzy relation with various conditions, such as reflexive and symmetric [33, 230]. It is a valuable tool and widely applied in multiple applications, such as classification. Fuzzy similarity-based classifiers are supervised and non-parametric processes, which classify the multi-valued objects by finding the inherent similarities [47].

**1. Łukasiewicz Structure** A generalized form named Łukasiewicz-structure is utilized as a similarity measure. The Łukasiewicz algebra represents a continuous t-norm and defines the membership function as a fuzzy structure [139]. Łukasiewicz-structure defines the objects' memberships based on the principle that the mean of different Łukasiewicz-structure functions is still a Łukasiewicz-structure function. It has a multi-valued structure with a

solid connection to the first-order fuzzy logic [152]. The mathematical background is shown below.

• **Łukasiewicz Algebra** Given that $\mathbb{R}$ represents the set of real numbers, the infinite-valued Łukasiewicz algebra is defined as $\langle I, \rightarrow, \wedge, \vee, \otimes, \oplus, \neg \rangle$ on the unit interval $I := [0,1] := \{x \in \mathbb{R} : 0 \leq x \leq 1\}$ in the following [36].

- *Implication*: $(\rightarrow)$Ł $\quad a \rightarrow b := min(1, 1-a+b)$
- *Conjunction*: $(\wedge)$Ł $\quad a \wedge b := min(a,b)$
- *Disjunction*: $(\vee)$Ł $\quad a \wedge b := max(a,b)$
- *Strong conjunction*: $(\otimes)$Ł $\quad a \otimes b := max(0, a+b-1)$
- *Weak disjunction*: $(\oplus)$Ł $\quad a \oplus b := min(1, a+b)$
- *Negation*: $(\neg)$Ł $\quad \neg a := 1-a$

A lattice, denoted as $\langle L, \leq, \wedge, \vee \rangle$, is referred to a partially ordered set when $x \wedge y$ and $x \vee y$ both exist in $L$ ($\forall x, y \in L$). It is called residuated when the number 1 is regarded as the greatest element; binary operation multiplication, $\odot$; residuum, $\rightarrow$, as shown in the following conditions [139].

1. $\odot$ is associative, commutative and isotone.
2. $a \odot 1 = a, \ \forall a \in L$.
3. $\forall a, b, c \in L, \ a \odot b \leq b$ iff $a \leq b \rightarrow c$.

• **Generalized Lukasiewicz-Structure** Given that $L$ is the real unit interval [0, 1] endowed with the usual order relation, then the generalized Łukasiewicz-structure is constructed using the residuated lattice in Equation 2.40.

$$\begin{aligned} a \odot b &= \sqrt[p]{max\{a^p + b^p - 1, 0\}} \\ a \rightarrow b &= min\{1, \sqrt[p]{1 - a^p + b^p}\} \end{aligned} \tag{2.40}$$

where $p$ is a fixed number. It becomes the normal Łukasiewicz structure when $p = 1$.

• **Fuzzy Similarity** Given that $L$ and $\mathbb{X}$ are the residuated lattice and a non-empty set, respectively, the fuzzy similarity is defined as the $L$ valued binary relation $S$ in $\mathbb{X}$ with the following conditions [205]. It is also called a Łukasiewicz-valued fuzzy similarity.

1. $\forall x \in \mathbb{X}: \ S\langle x,x\rangle = 1$
2. $\forall x,y \in \mathbb{X}: \ S\langle x,y\rangle = S\langle y,x\rangle$
3. $\forall x,y,z \in \mathbb{X}: \ S\langle x,y\rangle \odot S\langle y,z\rangle \leq S\langle x,z\rangle$

**2. Maximal fuzzy similarity** Given that a dataset $\mathbb{D}$ has $M$ samples and $N$ features $F_1,...,F_N$, the fuzzy similarity for comparing any two samples $x_j, x_k$ ($x_j, x_k \in \mathbb{D}$) within the feature $F_i$ is depicted as [205]:

$$S_{F_i}\langle x_j,x_k\rangle = x_j(F_i) \leftrightarrow x_k(F_i), \quad 1 \leq i \leq N \tag{2.41}$$

where $1 \leq j,k \leq M$ and $j \neq k$. Through averaging the fuzzy similarity among all the features, the maximal fuzzy similarity is depicted below.

$$S\langle x_j,x_k\rangle = \frac{1}{N}\sum_{i=1}^{N}(x_j(F_i) \leftrightarrow x_k(F_i)) \tag{2.42}$$

To emphasize different feature importance, various non-zero weights $(w_1,...,w_N)$ are applied on features using the formula below [205].

$$S\langle x_j,x_k\rangle = \frac{\sum_{i=1}^{N} w_i(x_j(F_i) \leftrightarrow x_k(F_i))}{\sum_{i=1}^{N} w_i} \tag{2.43}$$

In Łukasiewicz structure, the equivalence relation $a \leftrightarrow b$ is defined as $1 - max\{a,b\} + min\{a,b\}$ or equally $1 - |a-b|$. Therefore, the formula of maximal fuzzy similarity is represented in Equation 2.44.

$$S\langle x_j, x_k\rangle = 1 - \frac{1}{N}\sum_{i=1}^{N} |x_j(F_i) - x_k(F_i)| \tag{2.44}$$

Hence, the formula with different weights is shown below.

$$S\langle x_j, x_k\rangle = 1 - \frac{\sum_{i=1}^{N} w_i |x_j(F_i) - x_k(F_i)|}{\sum_{i=1}^{N} w_i} \tag{2.45}$$

From Equation 2.40, the generalized Łukasiewicz structure implicates that $a \to b = min\{1, \sqrt[p]{1 - a^p + b^p}\}$. Therefore, the following form is given below.

$$\begin{aligned} a \leftrightarrow b =& (a \to b) \wedge (b \to a) \\ =& min\{1, \sqrt[p]{1 - a^p + b^p}\} \wedge min\{1, \sqrt[p]{1 - b^p + a^p}\} \\ =& min\{\sqrt[p]{1 - a^p + b^p}\}, \sqrt[p]{1 - b^p + a^p}\} \\ =& \sqrt[p]{1 - max\{a^p, b^p\} + min\{a^p, b^p\}} \\ =& \sqrt[p]{1 - |a^p - b^p|} \end{aligned} \tag{2.46}$$

Hence, the formulas of maximal fuzzy similarity are depicted in Equation 2.47 and 2.48.

$$S\langle x_j, x_k\rangle = \frac{1}{N}\sum_{i=1}^{N} \sqrt[p]{1 - |x_j^p(F_i) - x_k^p(F_i)|} \tag{2.47}$$

$$S\langle x_j, x_k\rangle = \frac{\sum_{i=1}^{N} w_i \sqrt[p]{1 - |x_j^p(F_i) - x_k^p(F_i)|}}{\sum_{i=1}^{N} w_i} \tag{2.48}$$

The correct choice of the power value $p$ in the generalized Łukasiewicz structure can significantly improve the decision-making performance.

### 2.2.4 Fuzzy Systems

Fuzzy systems are knowledge-based or rule-based systems, which consist of the so-called fuzzy IF-THEN rules. A fuzzy IF-THEN rule is an IF-THEN statement using the membership functions to characterize the natural languages. The fuzzy system, also named fuzzy logic system (FLS), utilizes fuzzy logic and fuzzy theory as the basis to represent knowledge [140]. It simultaneously handles numerical data and linguistic knowledge by nonlinear mapping an input data vector to a scalar output. The enormous number of possibilities and mappings lead to the proliferation of various fuzzy systems [141]. There are three types of fuzzy systems: pure fuzzy systems, Takagi-Sugeno-Kang (TSK) fuzzy systems, and Mamdani fuzzy systems. Compared with the other alternatives, Mamdani fuzzy systems provide a natural framework to represent the knowledge using different fuzzy principles [211]. Therefore, it has become a widely used fuzzy system and is introduced below.

**1. Framework of Mamdani fuzzy system** The Mamdani fuzzy system transforms the real-valued variables and fuzzy sets using a fuzzifier and defuzzifier. The overall framework and basic configurations of the Mamdani fuzzy system are shown in Fig. 2.3.

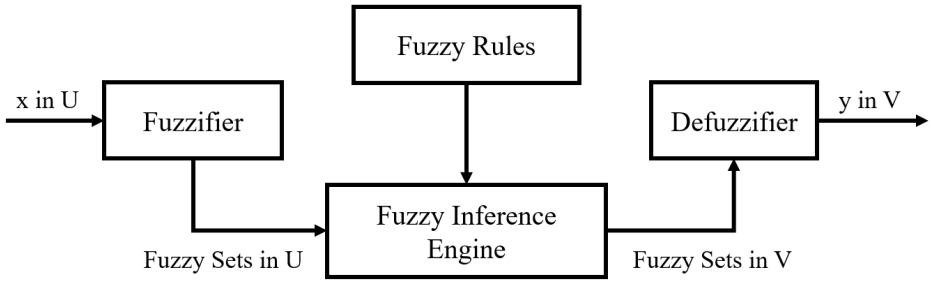



Fig. 2.3 The framework of Mamdani fuzzy systems

Mamdani fuzzy systems map the crisp inputs into the crisp outputs with four components: fuzzifier, fuzzy inference engine, fuzzy rules, and defuzzifier. Using the established rules, the fuzzy system, which maps from inputs to outputs, can be expressed quantitatively as $y = f(x)$. In the framework of Fig. 2.3, crisp numbers are mapped into fuzzy sets using fuzzifiers at the

beginning. Fuzzy rules, which are a collection of IF-THEN statements, are then extracted from human knowledge or numerical data [212]. Next, the fuzzy inference engine maps the fuzzy sets into fuzzy sets based on the generated fuzzy rules [141]. Finally, defuzzifiers transform the output fuzzy sets back into crisp numbers. The different components of a fuzzy system are shown as follows.

**2. Components of the Fuzzy System**

• **Fuzzification** Fuzzification maps a real-valued point $\mathbf{x}^* \in \mathbb{U} \subset \mathbb{R}^N, \mathbf{x}^* = (x_1^*, ..., x_N^*)$ into a fuzzy set $A'$ in $\mathbb{U}$. The fuzzification process could provide the fuzzy system manifold benefits, such as suppressing the input noise, simplifying the subsequent computations, etc. Given that $a_i$, $b_i$ $(1 \leq i \leq N)$ are the positive parameters and t-norm $\star$ is the algebraic product, different kinds of fuzzifiers are introduced below [211].

1. *Singleton Fuzzifier:* A singleton fuzzifier maps a real-valued point into a fuzzy singleton, as shown in Equation 2.49.

$$\mu_{A'}(x) = \begin{cases} 1 & if\ x = \mathbf{x}^* \\ 0 & otherwise \end{cases} \tag{2.49}$$

2. *Gaussian Fuzzifier:* A gaussian fuzzifier maps $\mathbf{x}^* \in \mathbb{U}$ into the fuzzy sets with the gaussian membership function.

$$\mu_{A'}(x) = e^{-(\frac{x_1 - x_1^*}{a_1})^2} \star ... \star e^{-(\frac{x_N - x_N^*}{a_N})^2} \tag{2.50}$$

3. *Triangular Fuzzifier:* A triangular fuzzifier maps $\mathbf{x}^* \in \mathbb{U}$ into the fuzzy sets with the triangular membership function.

$$\mu_{A'}(x) = \begin{cases} (1 - \frac{|x_1 - x_1^*|}{b_1}) \star ... \star (1 - \frac{|x_N - x_N^*|}{b_N}), \; if \; |x_i - x_i^*| \leq b_i \\ 0, \quad otherwise \end{cases} \tag{2.51}$$

• **Fuzzy Rules Base** The fuzzy rule base is the heart of a fuzzy system, consisting of a set of fuzzy IF-THEN rules. The other components are used to implement these rules reasonably and efficiently. Specifically, the following fuzzy IF-THEN rules are comprised in the fuzzy rule base.

$$Rule^l : IF \; x_1 \; is \; A_1^l \; and \; ... \; and \; x_n \; is \; A_n^l, \; THEN \; y \; is \; B^l \tag{2.52}$$

where $A_i^l$ and $B^l$ present fuzzy sets in $\mathbb{U}_i \subset \mathbb{R}$ and $\mathbb{V} \subset \mathbb{R}$, respectively. $x = (x_1, x_2, ..., x_n)^T \in \mathbb{U}$ and $y \in \mathbb{V}$ are the input and output variables of the fuzzy system. Besides, $Rule^l$ stands for the $l$th rule ($l = 1, 2, ..., M$), where $M$ stands for the number of total rules in the fuzzy rules base [211].

• **Fuzzy Inference Engine** By using fuzzy logic principles, the fuzzy inference engine combines the fuzzy IF-THEN rules of the fuzzy rule base into a mapping from a fuzzy set $A$ in $\mathbb{U}$ to a fuzzy set $B'$ in $\mathbb{V}$. In the literature, there are generally two inference approaches using a set of rules: composition-based inference and individual-rule-based inference. In composition-based inference, all rules in the fuzzy rule base are combined into a single fuzzy relation in $\mathbb{U} \times \mathbb{V}$, which is viewed as a single fuzzy IF-THEN rule. For the individual-rule-based inference, each rule in the fuzzy rules base determines an output fuzzy set. Besides, the output of the whole fuzzy inference engine is the combination of the $M$ individual fuzzy sets. Those combinations can be taken either by a union or an intersection [211].

• **Defuzzification** Defuzzification process maps a fuzzy set $B' \in \mathbb{V} \subset \mathbb{R}^N$ into the crisp point $y^* \in \mathbb{V}$. Conceptually, it selects the most representative point in $\mathbb{V}$ to represent the fuzzy set $B'$. From the literature, some widely used defuzzifiers are introduced below [211].

1. *Center of Gravity Defuzzifier:* A center of gravity defuzzifier specifies $y^*$ as the center of area covered by membership function of $B'$, as shown in Equation 2.53.

$$y^* = \frac{\int_{\mathbb{V}} y \mu_{B'}(y) dy}{\int_{\mathbb{V}} \mu_{B'}(y) dy} \tag{2.53}$$

   where $\int_{\mathbb{V}}$ represents the conventional integral.

2. *Center Average Defuzzifier:* The fuzzy set $B'$ represents the union or intersection of different fuzzy sets. The weighted average of the centers of the fuzzy sets is estimated using a good approximation of Equation 2.53. Given that the number of the fuzzy sets is $M$ and the center of gravity of the fuzzy set $B_j$ ($1 \leq j \leq M$) is $\bar{y}_j$, the defuzzifier is depicted as below [141].

$$y^* = \frac{\sum_{j=1}^{M} \bar{y}_j \mu_{B_j}(y_j)}{\sum_{j=1}^{M} \mu_{B_j}(y_j)} \tag{2.54}$$

3. *Maximum Defuzzifier:* A maximum defuzzifier selects the fuzzy sets' maximum values as the output, which is a computationally efficient approach. When only a single point is contained in the fuzzy set's height, the output $y^*$ is uniquely defined. If the maximum values contain more than one data point, various defuzzifiers are proposed, such as the smallest of maximum, largest of maximum, or mean of maximum defuzzifiers [211].

## 2.3 Ensemble Learning

### 2.3.1 Background

Ensemble learning is defined as a machine learning paradigm that trains multiple learners to solve the same problem. Unlike the common machine learning approaches, which only learn one hypothesis, the ensemble methods construct a set of hypotheses and combine them [239]. It is known to obtain better performance than a single one and has become a prolific field in the machine learning area. As the old proverb said, "two heads are better than one." The rationale behind ensemble learning is to build a set of hypotheses using different methods and combine them in order to obtain better results [25]. An ensemble contains various learners named base learners or base methods. The generalization ability of an ensemble is usually stronger than that of base methods. Ensemble learning is quite effective in practice because it can boost weak learners, which are slightly better than a random guess, to strong learners with high predictive performance [239]. Unlike the typical approach that builds a single learning model to solve the problem, diverse kinds of base methods and their variance control make this approach effective and successful in many practical applications.

The idea of using ensemble learning is not only applicable to classification but also beneficial to improving some other machine learning disciplines such as FS. By combining the output of different FS algorithms, ensemble FS methods provide a solution to understand the feature importance better and bring about many potential benefits at the same time. Firstly, by representing the inherent characteristics of data well, ensemble FS methods can produce a comprehensively good performance, especially on stability and robustness. Secondly, ensemble FS methods act as the unified approach to stabilising the feature scores and assigning the weights and ranks among all the features. Rather than utilize a single FS technique, ensemble methods combine multiple models to solve particular problems. Therefore, the users can be released from the pressure of choosing the optimal one. Thirdly, the optimal feature ranking result can hardly be attained due to a particular FS method having constrained search space. Ensemble FS methods help alleviate this problem by aggregating

the outputs from various FS methods and producing a better approximation to the optimal feature ranking sequence [179]. Fourthly, the ensemble FS framework provides solutions to handle the high discrepancy in the results caused by the increasing number of various FS methods.

### 2.3.2 Existing Approaches

Numerous effective techniques within ensemble learning are developed from the literature to improve FS methods' performance. The famous examples using ensemble learning include bagging and boosting [240]. The bagging methods increase the performance and reduce the variance of the final result through the repeated sampling process, such as Random Forest, etc [122]. Besides, the boosting methods improve the performance mainly by fitting the errors produced by the weaker learners, such as Gradient Boosted Decision Tree (GBDT), etc [221]. In general, the data perturbation strategy increases FS methods' stability through perturbing the training set, adding new data, and integrating multiple FS methods. Specifically, random sampling reorganizes the original training datasets and integrates the feature subsets to improve the possibility of choosing a similar feature sequence, leading to higher stability and robustness.

In 2008, by investigating the different ensemble FS methods, Syeys *et al.* have proved that the ensemble approach can achieve reasonably good performance on high-dimensional datasets with small sample sizes. The results indicate that the ensemble techniques provide more stable feature subsets than a single FS method. Besides, they also investigated the impact of the ensemble FS method on classification performance. Therefore, it is concluded that incorporating both classification performance and robustness in the evaluation strategy will improve the model's overall performance [179]. In 2010, targeted for the gene selection problem in the bioinformatics area, Zhao *et al.* proposed a novel approach to integrate different types of knowledge to identify biologically relevant genes [236]. In 2012, to make

further improvements on the performance, Li *et al.* proposed a diversity regularized ensemble feature weighting framework. The base selectors are constructed by local learning with the logistic loss for robustness to huge irrelevant features and small samples [131]. In 2014, Mitchell *et al.* implemented a task-parallel version of the random forest algorithm using the bootstrap aggregation sampling process. By ranking the results from each base learner, the FS method's final result is obtained [145]. In 2017, Moran-Fernandez *et al.* proposed a distributed approach to data partitioned by features. From the experiments, it can be concluded that the proposed method significantly reduced the runtime while maintaining the classification performance [146]. In 2017, Borja *et al.* presented several FS ensemble configurations based on combining rankings of features from individual rankers according to the combination method and threshold value used. Different synthetic, real classical, and even bioinformatic datasets have been used to evaluate the performance of each proposed ensemble configuration [184].

From the literature, the individual FS methods from an ensemble approach are known as base selectors. Based on the chosen base selectors, ensemble FS methods are generally categorized into homogeneous approach (data variation approach), and heterogeneous approach (function variation approach) [16]. When the base selectors are all of the same kind of FS method, the ensemble is known as homogeneous. In this case, the same FS method is implemented on different nodes distributed by the datasets. Otherwise, the ensemble approach is named heterogeneous, where different FS methods are applied to the same training dataset [185]. The detailed introduction and literature review of the applied techniques on homogeneous and heterogeneous approaches are shown below.

• **Homogeneous approach** By definition, a homogeneous method implements the same FS algorithm on different subsets of the data. The computational time can also be reduced by processing the data in parallel nodes [185]. In the literature, various methods have been proposed by researchers. In 2012, Nikulin *et al.* developed a homogeneous ensemble FS

method to deal with the imbalanced datasets and classes using Wilcoxon based FS method. The use of the Wilcoxon criterion can help the computation for sparse data [150]. In 2017, Pes *et al.* evaluated and discussed the rationale, effects and implications of the homogeneous ensemble approach on various aspects, such as predictive accuracy and stability. The genomic benchmarks' results provide helpful insight into both the benefits and the limitations of such an ensemble approach [162]. In 2020, Soheili *et al.* analyzed nine different rank fusion methods within a homogeneous ensemble feature ranking algorithm. The statistical analysis revealed that no significant difference in the performance of those rank fusion methods was found [194]. In 2021, Hosni *et al.* investigated different filter FS techniques to check the predictive capability of various machine learning techniques. Then, they concluded that the homogeneous ensembles are statistically more accurate than each of the single techniques [85].

• **Heterogeneous approach** By definition, a heterogeneous method applies different FS methods to the same dataset for training. The use of the heterogeneous approach can help ensure a stable and robust FS result in various applications [185]. In 2012, Bolon-Canedo *et al.* described a new heterogeneous ensemble framework for FS, which consists of five filter FS methods with different metrics, and a variety of strengths and weaknesses. The diversity among the base selectors was guaranteed to take advantage of different methods and therefore boost the overall performance [17]. In 2016, Haque *et al.* proposed a genetic-based search method to find the optimum combination of the heterogeneous ensemble FS method. From the empirical study and experimental results, the genetic algorithm is proven to be a superior and reliable approach for heterogeneous ensemble construction [75]. In 2017, Hosni *et al.* investigated the impact between the estimated accuracy of heterogeneous ensembles and two widely used filter FS methods, which include correlation-based FS and ReliefF method [84]. In 2018, Brahim *et al.* proposed a heterogeneous ensemble FS approach based on the reliability assessment of FS methods, which aims to provide a unique and

stable FS without ignoring the aspect of predictive accuracy. The results indicated that the proposed approach improved the classification performance and stability on high dimensional datasets [21].

### 2.3.3 Main Issues

As mentioned above, various ensemble FS methods have been proposed on the aspect of both homogeneous and heterogeneous approaches. The homogeneous approach utilizes the data's information, while the heterogeneous approach takes advantage of multiple methods. However, the FS methods that incorporate homogeneous and heterogeneous approaches into the same framework have rarely been reported. Besides, some central issues still need to be clarified and addressed when designing the ensemble FS framework in general.

**1. Selection of base selectors** A vast body of different FS methods exist, such as filter, embedded, and wrapper approaches. Among them, many FS methods have been utilized as the base selectors from the literature. In Minaei-Bidgoli's work, the improved fuzzy entropy-based FS methods have been used to solve the drawback of parameter sensitivity with the ensemble-based approach [144]. From Bolon-Canedo's paper, five filter FS methods based on different metrics were employed as the base selectors, such as correlation-based FS, consistency-based filter FS, information gain FS, ReliefF, etc [17]. From Zhou's work, different reliefF algorithms (combined as multi-reliefF methods) acted as the base selectors within an improved ensemble FS framework [238].

From the literature, it can be concluded that filter methods are preferred over the other FS methods because they are independent of the learning algorithms with high computational efficiency. In practice, the choice of the base selectors remains an essential topic in the FS area. A good and reasonable selection of the FS methods can benefit the ensemble method's final performance.

**2. Bootstrapping aggregation process** Bootstrap aggregation, or named bagging, is one of the essential steps within some of the ensemble FS methods' frameworks. It can improve machine learning methods' generalizability and predictive performance by using the data perturbation strategy. Recently, it has gradually become a widely-used meta-algorithm in the FS area. In 2014, Zhou *et al.* proposed an improved ensemble FS framework with the use of random sampling and random FS to increase the performance on stability and reduce the computational cost at the same time. The bootstrap aggregation process was utilized to aggregate the results of multi-ReliefF methods [238]. In 2017, Pes *et al.* designed an ensemble framework with bootstrap sampling for FS. Two different strategies were applied to integrate the feature ranking results, including median and exponential aggregation [162]. In the bioinformatics area, the bootstrap aggregation process can generate various bags and therefore enhance the FS methods' stability, which is essential in biological research [1, 179].

The literature has different ways of representing the FS methods' results using the bootstrap aggregation process. However, a comprehensive discussion and comparison among those methods are rarely reported to the author's best knowledge. Hence, it is still desirable to find some novel methods to aggregate different methods' results more effectively with an all-around better performance.

**3. Selection of aggregation methods** Various aggregation approaches emerged using FS evaluation criteria within the ensemble learning framework, such as simple and complex methods. Examples of the simple combination include minimum, mean, median, etc [215]. The complex methods include weighted mean aggregation (WMA), complete linear aggregation (CLA) [1] and so forth. In the literature, researchers have developed various methods for the aggregation process. In 2002, Joachims *et al.* utilized an SVM-rank algorithm to learn the ranking functions and combine different methods [96]. In 2010, Yang *et al.* developed a Multi-Criteria Fusion based Recursive Feature Elimination (MCF-RFE) to improve both the classification performance and stability of FS results [224]. In 2012, Kolde *et al.* uti-

lized RobustRankAggreg (RRA) package to improve the efficiency and accuracy during the combination process [111].

There are various ways of aggregating different FS methods' results in the literature. However, according to the author's best knowledge, a comprehensive discussion and comparison within those methods are still limited. Besides, developing novel approaches to better aggregate various methods' results for specific applications is still needed.

## 2.4 Meta Learning

### 2.4.1 Background

**1. Introduction** With the increasing number of high-dimensional datasets, FS gradually becomes one of the critical steps in the machine learning area. Many kinds of FS methods were proposed in the literature [128]. However, no single dominating FS method always obtains the best performance in all aspects [218]. One of the main issues is that those FS methods' performance varies data-dependently. It seems impossible to state categorically the optimal FS method which provides the best performance for all kinds of data [86]. Hence, for a given dataset without a priori knowledge, various methods need to be experimented with using a trial and error approach. Therefore, the process of selecting a suitable FS method becomes time-consuming, costly, and even unachievable in the unsupervised scenarios [190].

Under the circumstances that no single method can always perform the best in all cases, many different approaches have been proposed in the literature. In the last section, ensemble learning methods can construct an ultimate FS method that consistently achieves optimal performance by assuming that the output combination of multiple models is better than a single one. In this section, meta-learning methods, also described as the formal method integration [153], can help solve this problem by finding and recommending a suitable FS method for a given dataset. The related and even broad concept of "automated machine

learning" (AutoML) has become increasingly essential and recently gained more attention in the literature [209, 124]. To better understand those concepts, this section provides a general description of AutoML as follows.

**2. Automated Machine Learning (AutoML)** As illustrated by the name "automated machine learning", AutoML combines automation and machine learning. It is defined as a computer program with good generalization performance on the input data and given tasks. AutoML aims to take humans' place in identifying configurations that are proper to machine learning computer programs with little computational cost. Besides, one of its goals is to construct high-level controlling methods over learning tools to find the proper configurations automatically and efficiently [225]. Inspired by the human-involved process in the automation area, a framework for AutoML can be summarized as shown in Figure 2.4.

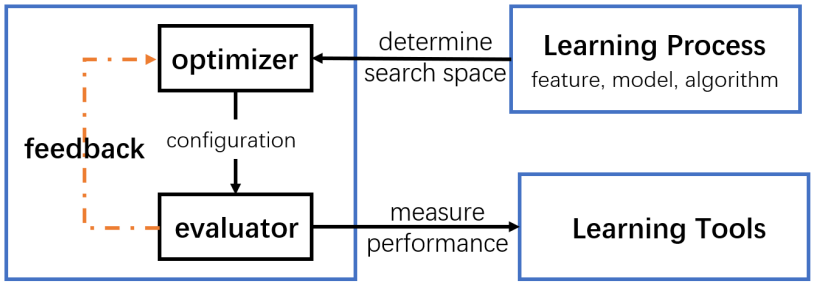



Fig. 2.4 Basic framework of AutoML [225]

In the framework, the AutoML controller takes the place of human beings to search for the learning tools' proper configurations. There are two critical components within the controller: evaluator and optimizer. The evaluator aims to measure the learning tools' performance with configurations from the optimizer. Optimizer aims to update or generate the learning tools' most suitable configurations [225]. Based on how to solve an AutoML problem, the existing techniques can be divided into basic and experienced ones. The basic techniques are categorized using the optimizer and evaluator. On the other hand, the experienced techniques learn and accumulate knowledge from the past searches or external data, including meta-

learning and transfer learning [154, 209]. The following parts introduce and discuss the definition and applications of meta-learning to form a better understanding.

**3. Definition of Meta Learning** Meta-learning, also named learning to learn, is designed to learn the meta knowledge in order to improve the model learning performance [22]. It learns the new tasks faster by learning from the experience or the "meta-data". By observing how different machine learning approaches perform on the learning tasks, meta-learning methods learn to select the most appropriate FS method for a given dataset. In practice, meta-learning methods can improve machine learning architectures' design and help select the most suitable algorithms using a data-driven approach [207]. It is normally trained to learn the relationship between the characteristics of the training datasets and the suitable algorithms [156].

The challenges behind meta-learning methods lie in learning from prior experience in a systematic and data-driven approach. Firstly, meta-data need to be collected, which describe prior learning tasks and previously learned models. The exact algorithm configurations are used to train the models, including hyperparameter settings, pipeline compositions, and network architectures. Secondly, the knowledge is extracted and transferred from the prior meta-data for searching the most suitable models in later tasks [207].

### 2.4.2 Applications

Meta-learning methods could not only dramatically improve the design of the machine learning structure but also solve the model selection issue with a data-driven approach [207]. With those benefits, the methods are widely applied in different machine learning areas, such as classification, regression, optimization, time series prediction, etc [117, 124, 125]. Scholars investigated the applications of meta-learning for FS [156, 56] in two main aspects. One aspect is algorithm selection, which chooses the best algorithm by learning the relationship

between the characteristics of data and the performance of different algorithms [100]. The other one is parameter selection, which determines the optimal parameters within a sophisticated method [167]. Specifically, the AutoML and meta-learning methods are applied in several popular areas, such as hyperparameter optimization, feature engineering, etc. Besides, neural architecture search (NAS) has become an essential and particular topic for neural networks, targeting deep learning. The NAS method configures the features, models and algorithms simultaneously [78, 225]. Therefore, the concept of NAS is firstly introduced in this section, followed by the other application areas.

**1. Neural Architecture Search (NAS)** NAS methods are currently a widely studied research topic, which targets searching for suitable deep network architectures that suit the learning problem. They can automatically search the neural network architectures for a given problem. Therefore, it can help remedy the difficulties of manually designing neural networks [242]. Recent research in the meta-learning area focuses on the neural architecture search based on its robust, generic, and versatile properties. However, this may lead to a high computational cost when exploring suitable configurations.

**2. Hyperparameter optimization** Hyperparameter optimization methods choose a set of hyperparameters for a specific learning algorithm. The inherent hyper-parameters stand for the parameters used to control the learning process by itself [78].

**3. Feature Engineering** Feature engineering methods aim to construct features from the data so that the subsequent learning tools can achieve good performance. The process has two main steps: creating features from the data and enhancing features' discriminative ability. However, the first step heavily depends on the application scenarios and humans' expertise, with no standard or principled methods. The second step, feature enhancing methods, applies

when the original features from the data may not be good enough. Some post-processing steps need to be performed to improve the learning performance [12].

There are three kinds of methods to enhance features from the literature. (1) dimension reduction: reduces the number of random features by obtaining a set of principal ones. It becomes valid and valuable when the meta-features have gained significant redundancy with high feature dimensionality. (2) feature generation: constructs new features based on pre-defined operations. In some cases, the new interactions among the original feature may significantly improve the learning performance [105]. (3) feature encoding: re-interprets the original features using dictionaries learned from the data. The indiscriminate samples in the original space may become separable in the new space [51].

The proper use of feature engineering could lead to many potential benefits, such as high computational efficiency. Domain knowledge, data science expertise, and even the trial process are still required in the manual stage to design and select the most appropriate meta features [78]. Because of its high computational efficiency, this research investigates and applies the feature engineering approach.

### 2.4.3 Main Issues

In the literature, some central issues of applying feature engineering are discussed below.

**1. Construction of training data repository** The central concept of meta-learning is to learn the knowledge from a data repository. Hence, the selection and construction of different data repositories become especially important. Many real-world datasets are used to construct the meta database in the literature. Parmezan et al. [156] utilized 150 practical datasets during the meta-learning process. However, it is time-consuming and sometimes even restricted by ethical issues to collect various datasets from real-world applications, especially in the biomedical area. Different data sources may need various types of consent,

such as informed consent, broad consent and implied consent [54, 202]. More importantly, these real-world datasets may not cover a wide range of characteristics similar to the given unseen dataset under consideration.

**2. Selection of meta features** Meta features are used to depict certain relationships with the algorithm performance. The selection of meta-features is dependent on the related problem. In 2014, Shafait *et al.* generally classified the meta-features into five different groups, including simple, statistic, information theoretic-based, model-based, and land-marking based [168]. In 2017, Cruz *et al.* proposed and extracted 15 sets of meta-features in the research [35]. In 2019, Vanschoren *et al.* provided a concise overview of the most commonly used meta-features for the reason they are indicative of model performance [208].

**3. Construction of meta model** Meta model refers to the decision-making method to recommend the most suitable configurations given the new task's meta-features. It learns the complex relationship between the extracted meta-features and the meta label using specific configurations. In the literature, there are a vast number of research on building meta models in the meta-learning process, including decision trees, support vector machine, KNN, etc [22, 124, 156]. However, the fuzzy-based approach is rarely reported as the meta-model before, to the author's best knowledge.

### 2.4.4 Existing Approaches

In the literature, the research on meta-learning can be generally categorized in five different directions: based on metric learning, based on parameter training, based on gradient optimization, based on memory augmentation and based on data augmentation. The detailed description of those directions is shown as follows.

**1. Based on metric learning** In 2015, Koch *et al.* explored a method by learning siamese neural networks to solve the one-shot image recognition problem. A unique structure has been employed to rank the similarity between the inputs [109]. In 2016, Vinyals *et al.* introduced Matching Networks, a new neural architecture by employing the ideas from metric learning based on deep neural features. The framework can learn a network that maps a small labelled support set and an unlabelled example to its label [210]. In 2017, Snell *et al.* proposed Prototypical Networks, which learn a metric space where the classification is performed by computing the distances to prototype representations of each class [193]. In 2018, Sung *et al.* presented a conceptually simple, flexible, and generic framework called Relation Network by training end-to-end from scratch. The framework learns a deep distance metric to compare a small number of images within episodes [199].

**2. Based on parameters training** In 2017, Finn *et al.* proposed a well-known algorithm named Model-Agnostic Meta-Learning (MAML), which is compatible with any model trained with gradient descent and can be applied to different learning problems. The model's parameters are explicitly trained so that a small number of gradient steps can generate good generalization performance on that task [57]. In 2019, Sun *et al.* proposed a novel few-shot learning method called Meta-Transfer Learning (MTL) which learns to adapt a deep neural network for few-shot learning tasks. The "transfer" process is achieved by learning the scaling and shifting functions of deep neural network weights for each task [198].

**3. Based on gradient optimization** In 2017, Ravi *et al.* proposed a Long Short Term Memory (LSTM) based meta-learner model to learn the optimization algorithm. The LSTM meta learner can utilize its state to represent the learning updates of a classifier's parameter. It is trained to discover both a good initialization of the learner's parameters and a successful mechanism to update the learner's parameters for some new classification task [166].

**4. Based on memory augmentation** In 2016, based on the architectures with augmented memory capacities, such as Neural Turning Machines (NTMs), Santoro *et al.* demonstrated the ability of a memory-augmented neural network to rapidly assimilate new data and leverage this data to make accurate predictions using only a few samples [180]. In 2018, Cai *et al.* proposed Memory Matching Networks (MM-Net), a novel deep architecture with the idea of augmenting Convolutional Neural Networks (CNNs) using memory and learning to learn the network parameters for the unlabelled images in one-shot learning [27].

**5. Based on data augmentation** In 2017, Hariharan *et al.* introduced the techniques to hallucinate additional training examples for data-starved classes in the few-shot recognition problem [76]. In 2018, Wang *et al.* also presented an approach to low-shot learning which uses a trained hallucinator to generate additional examples [213]. In 2018, based on the idea that fake samples produced by the generator can help classifiers learn a sharper decision boundary between different classes from a few samples, Zhang *et al.* proposed a conceptually simple and generic framework named MetaGAN to boost the performance of few-shot learning models [234]. In the same year, Gao *et al.* proposed a novel approach, Covariance-Preserving Adversarial Augmentation Networks (CPAAN), to low-shot learning, which can augment data for novel classes by training a cyclic GAN model [60].

## 2.5 Summary

This chapter presents an overview of four fundamental concepts: FS methods, fuzzy theory, ensemble learning and meta-learning in this research. The literature review and background knowledge of those concepts are also discussed. To begin with, this chapter introduces the definitions and working principles of many FS methods in the literature, such as the ranking-based, subset-based, wrapper, embedded, and filter approaches. Unlike the other techniques, the ranking-based filter FS methods are independent of any learning algorithms,

which are computationally efficient, especially in the case of high dimensional data. For these rank-based filter methods, a measure is incorporated to evaluate the features' importance and filter out irrelevant features, where features are therefore ranked based on their significance and quality. Hence, with the investigation of the different ranking-based filter FS methods, this research aims to create solutions for a better understanding of feature importance, such that the features can represent the inherent characteristics of data well and achieve comprehensively good performance.

With the knowledge that noise and uncertainty exist in practical datasets, the fuzzy theory is introduced as a unified framework to model the uncertainties. Four related concepts are introduced: fuzzy sets, fuzzy entropy, fuzzy similarity, and fuzzy systems. Introducing the fundamental knowledge and different categories of FS methods and fuzzy theory helps build a good foundation to explore ranking-based FS methods using fuzzy theory. Therefore, it naturally leads to the research in Chapter 3, which explores a ranking-based FS method that incorporates fuzzy theory to handle uncertainties, thereby producing a good predictive performance.

Additionally, the introduction of various FS methods raises a question: how to reasonably and sensitively measure the performance of different FS techniques. Many FS methods are normally evaluated using the downstream predictive performance alone. Other aspects of FS performance receive relatively little attention. It may lead to a biased result during the evaluation process. Therefore, Chapter 4 aims to discuss and tackle this issue in more detail by reviewing the existing performance evaluation metrics for FS method comparison and proposing new metrics that are supplementary to the existing methods.

Given a range of various FS algorithms, Section 2.3 introduces and describes ensemble learning, which provides a solution to combine different FS methods to obtain better performance. However, the majority of the existing ensemble learning methods are either homogeneous or heterogeneous. Therefore, the research in Chapter 5 aims to develop an

ensemble learning framework which combines homogeneous and heterogeneous approaches using the fuzzy-based framework to achieve better feature ranking.

On the other hand, as an alternative to the ensemble learning framework, Section 2.4 discusses meta-learning, which can help find and recommend a suitable FS method for a given dataset. Various existing approaches are introduced, such as metric learning-based, data augmentation-based, etc. The research in Chapter 6 aims to develop an FS recommendation framework to suggest a suitable FS method for a given dataset. In order to overcome the data scarcity problem, the data augmentation techniques are employed by using synthesized data for training a meta-learning framework to achieve feature ranking.

# Chapter 3

# Performance Optimization of a Fuzzy Entropy based FS Method

## 3.1 Introduction

Unlike the semantics-preserving FS algorithms that output the final feature subset [93], this chapter aims to explore an efficient feature ranking-based method that ranks the features based on feature importance rather than producing a feature subset. As discussed in Section 2.2.3, similarity measure becomes one of the fundamental mathematical processes, while Łukasiewicz-Structure is similarity based and has been applied in many pattern recognition problems. In 2001, Pasi Luukka followed the Łukasiewicz-Structure and proposed a similarity based classifier using generalized mean [139]. In his later published work [138], combined with this similarity-based classifier, fuzzy entropy measures were used to achieve FS. The fuzzy entropy measure is used as a fuzziness measure, which evaluates the global deviations from the type of ordinary fuzzy sets [8]. The method successfully managed to discard the non-important features from the candidate feature set. The FS method based on fuzzy entropy measures can facilitate the classification task to be performed faster with an increased classification accuracy. Besides, the method is highly computational efficient with

a readily comprehensible framework, which can be easily adapted to different applications. Therefore, this chapter mainly investigates the fuzzy entropy-based FS method.

The fuzzy entropy-based FS method proposed by Pasi Luukka mainly consists of three fundamental components, i.e., ideal vector calculation, similarity measurement, and fuzzy entropy calculation [138]. The similarity-based classification procedure is used as a classifier within the framework, while the fuzzy entropy-based FS technique is utilized as the FS process. Based on Luukka's work, this chapter comprehensively compares different measurements for each of these components and to other state-of-the-art FS methods. The main work of this chapter is the performance optimization of a fuzzy entropy-based FS method, including: (1) implementation of three different measurements for the critical components in the framework; (2) comprehensively comparison of the performance on different combinations of the measures; (3) comparison of different feature elimination processes for the classification models.

## 3.2 Methodology

Based on the method in Luukka's paper [138], a data-driven framework is implemented by incorporating various ideal vector calculations, fuzzy entropy functions, and fuzzy similarity measures together. The overall structure of the framework is illustrated in Fig. 3.1.

The implemented method aims to classify a total number of $S$ samples into the number of $C$ different classes $c_k, k \in [1, C]$ by their feature vector $\vec{x}_i$. The value $i$ represents the index of the samples. The number of features of $\vec{x}_i$ is denoted as $N$. Procedures of the implemented method are described below.

Step 1: For the training set, normalize each feature value into the range between 0 and 1 using the min-max normalization process [88]. The maximum value of each feature needs to be carefully determined to avoid using outliers. The outliers are identified if they

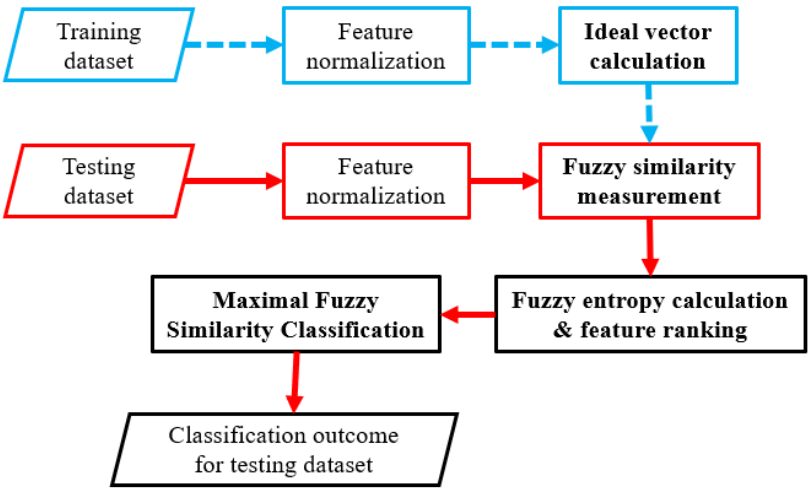



Fig. 3.1 Flowchart of fuzzy entropy-based FS framework. The blue dashed lines and red solid lines represent the data flows for the training and testing processes, respectively.

exceed three times the standard deviation from the mean value for each feature. Both the training and testing process undertake the same normalization procedure.

Step 2: Based on the normalized values in Step 1, calculate ideal vector $\vec{v}_k$ for the $k$th class.

Step 3: Apply the same normalization process of Step 1 to the testing dataset.

Step 4: Calculate the fuzzy similarity values between the feature vector $\vec{x}_i$ of the testing samples and the ideal vector $\vec{v}_k$ in Step 2.

Step 5: Based on the similarity values in Step 4, construct a similarity matrix with the dimensions of $S \times C \times N$. Calculate each feature's fuzzy entropy value (column of the matrix) and subsequently rank the features according to the values.

Step 6: Select the features and classify the testing set based on the ranked feature sequence from Step 5.

There are four main parts in the framework: ideal vector calculation, fuzzy similarity measurement, fuzzy entropy-based FS, and maximal fuzzy similarity classification. The detailed descriptions of these procedures are introduced in the following subsections.

### 3.2.1 Ideal Vector Calculation

The ideal vector is designed to be representative of the corresponding class. The vector can be user-defined or calculated from some samples which belong to the class. One of the simple ways to achieve this is to the generalized mean [138]. Therefore, the ideal vector represents the "mean" property of the samples in each class. Different methods can be used to calculate the ideal vector, namely, arithmetic means, geometric means, and harmonic means, as expressed in Equation 3.1, 3.2 and 3.3, respectively. In those equations, $C_k$ indicates the number of samples for the $k$th class; the value $j, j \in [1,N]$ represents the feature index. The performance comparing different ideal vector calculations is reported in Section 3.3.2.

1. **Arithmetic mean**: Given the $k$th class and $j$th feature, the arithmetic mean $\vec{v}_k^A(j)$ is calculated in Equation 3.1.

$$\vec{v}_k^A(j) = \frac{\sum_{i=1}^{C_k} \vec{x}_i(j)}{C_k} \tag{3.1}$$

2. **Geometric mean**: Given the $k$th class and $j$th feature, the geometric mean $\vec{v}_k^G(j)$ is calculated in Equation 3.2.

$$\vec{v}_k^G(j) = \sqrt[C_k]{\prod_{i=1}^{C_k} \vec{x}_i(j)} \tag{3.2}$$

3. **Harmonic mean**: Given the $k$th class and $j$th feature, the harmonic mean $\vec{v}_k^H(j)$ is calculated in Equation 3.3.

$$\vec{v}_k^H(j) = \frac{C_k}{\sum_{i=1}^{C_k} [1/\vec{x}_i(j)]} \tag{3.3}$$

### 3.2.2 Fuzzy Similarity Measurement

In this section, the similarity measurement is presented in the form of generalized Łukasiewicz algebra [178]. It measures the similarity between the $j$th element of feature vector $\vec{x}_i$ and the corresponding $j$th element of each class's ideal vector. The calculation process is described mathematically in Equation 3.4.

$$Sim\langle \vec{x}_i, \vec{v}_k, j\rangle = \sqrt[p]{1 - |\vec{x}_i(j)^p - \vec{v}_k(j)^p|},\ p > 0 \tag{3.4}$$

where $p$ is a hyper parameter to be optimized in Section 3.3.2. A similarity value is calculated for each feature in each class for each sample. Subsequently, a similarity matrix **P** with the dimensions of $(S \times C) \times N$ is constructed, as shown in Table 3.1. The next subsection will describe the fuzzy entropy calculation process for each feature.

Table 3.1 The formal denotation of a similarity matrix

| **Data** | **Feature 1** | **Feature 2** | **...** | **Feature N** |
|---|---|---|---|---|
| $\vec{x}_1$ | $Sim\langle \vec{x}_1, \vec{v}_1, 1\rangle$ | $Sim\langle \vec{x}_1, \vec{v}_1, 2\rangle$ | ... | $Sim\langle \vec{x}_1, \vec{v}_1, N\rangle$ |
| $\vec{x}_1$ | $Sim\langle \vec{x}_1, \vec{v}_2, 1\rangle$ | $Sim\langle \vec{x}_1, \vec{v}_2, 2\rangle$ | ... | $Sim\langle \vec{x}_1, \vec{v}_2, N\rangle$ |
| ... | ... | ... | ... | ... |
| $\vec{x}_1$ | $Sim\langle \vec{x}_1, \vec{v}_C, 1\rangle$ | $Sim\langle \vec{x}_1, \vec{v}_C, 2\rangle$ | ... | $Sim\langle \vec{x}_1, \vec{v}_C, N\rangle$ |
| $\vec{x}_2$ | $Sim\langle \vec{x}_2, \vec{v}_1, 1\rangle$ | $Sim\langle \vec{x}_2, \vec{v}_1, 2\rangle$ | ... | $Sim\langle \vec{x}_2, \vec{v}_1, N\rangle$ |
| ... | ... | ... | ... | ... |
| $\vec{x}_S$ | $Sim\langle \vec{x}_S, \vec{v}_C, 1\rangle$ | $Sim\langle \vec{x}_S, \vec{v}_C, 2\rangle$ | ... | $Sim\langle \vec{x}_S, \vec{v}_C, N\rangle$ |

### 3.2.3 Fuzzy Entropy based Feature Selection

In order to reduce the dimensionality and discard the non-important features, the fuzzy entropy-based FS process [138] is used to rank the features. Fuzzy entropy is the basic definition of the fuzzy information process and is widely used to measure the degree of vagueness in the various areas [114].

Based on the previously constructed similarity matrix, each feature's fuzzy entropy value (each column of the matrix **P**) is calculated using the fuzzy entropy functions described

below. The matrix $\mathbf{P}(r, j)$ represents the value of the $r$th row and $j$th column in the similarity matrix. These similarity values are utilized as the fuzzy set's membership function during the fuzzy entropy calculation process. Three different fuzzy entropy functions from Section 2.2.2 are implemented as expressed below.

1. *Non Probabilistic Entropy (Luca's method)*

$$H_1(j) = -\sum_{r=1}^{S\times C}[(\mathbf{P}(r,j)\log\mathbf{P}(r,j)) + (1-\mathbf{P}(r,j))\log(1-\mathbf{P}(r,j))] \tag{3.5}$$

2. *Weighted Measures of Fuzzy Entropy (Parkash's method)*

$$H_2(j) = \sum_{r=1}^{S\times C}[sin\frac{\pi\mathbf{P}(r,j)}{2} + sin\frac{\pi(1-\mathbf{P}(r,j))}{2} - 1] \tag{3.6}$$

3. *Geometry of Fuzzy Set and Entropy (Kosko's method)*

$$H_3(j) = \frac{\sum_{r=1}^{S\times C}(\mathbf{P}(r,j)\wedge(1-\mathbf{P}(r,j)))}{\sum_{r=1}^{S\times C}(\mathbf{P}(r,j)\vee(1-\mathbf{P}(r,j)))} \tag{3.7}$$

Subsequently, the fuzzy entropy values are used for feature ranking and selection. A fuzzy similarity-based classification process is then performed using the selected features.

### 3.2.4 Maximal Fuzzy Similarity Classification

The classification method is based on the maximal fuzzy similarity measures proposed by Luukka *et al.* [139]. Three similarity measurements are implemented for the ideal vector calculation.

1. *Similarity measure based on arithmetic mean*

$$Sim^A\langle\vec{x}_i,\vec{v}_k\rangle = \frac{1}{N'}\sum_{j=1}^{N'}\sqrt[p]{1-|\vec{x}_i(j)^p-\vec{v}_k(j)^p|} \tag{3.8}$$

2. *Similarity measure based on geometric mean*

$$Sim^G\langle \vec{x}_i, \vec{v}_k \rangle = \sqrt[N']{\prod_{j=1}^{N'} \sqrt[p]{1 - |\vec{x}_i(j)^p - \vec{v}_k(j)^p|}} \tag{3.9}$$

3. *Similarity measure based on harmonic mean*

$$Sim^H\langle \vec{x}_i, \vec{v}_k \rangle = \frac{N'}{\sum_{j=1}^{N'} \frac{1}{\sqrt[p]{1-|\vec{x}_i(j)^p - \vec{v}_k(j)^p|}}} \tag{3.10}$$

In Equation 3.8, 3.9 and 3.10, $\vec{x}_i$ represents the feature vector of the $i$th sample in the testing set after feature selection. $\vec{v}_k(j)$ stands for the recalculated ideal vector with the reduced dimension in the training set. $N'$ is the number of the selected features. The parameter $p$ is the same as that in Equation 3.4. Each testing sample is then classified into the class that produces the highest similarity value. It is noteworthy to mention that, based on the reduced feature subset, other classifiers can also be applied and compared, e.g., random forest, support vector machine, etc. The comparison between different classifiers is not the main focus of this chapter.

## 3.3 Experiments & Results

### 3.3.1 Materials

The experiments are based on commonly used numerical datasets as an initial test for method illustration and comparison. Three publicly available datasets are chosen from the UCI machine learning repository [46], which were all extracted from real-world problems with various features, samples, and sample distribution. The three datasets are all widely used in the data mining area for evaluating the performance of different methods. Hence, this research

work utilized these datasets to assess and compare the proposed methods' performance. The general properties of the datasets are shown in Table 3.2.

Table 3.2 Description of the experimental datasets

| **No.** | **Datasets** | *#C* | *#F* | *#S* | *#S* Distribution over *#C* |
|---|---|---|---|---|---|
| 1 | WBC | 2 | 9 | 682 | 239 / 443 |
| 2 | WDBC | 2 | 30 | 569 | 212 / 357 |
| 3 | Parkinsons | 2 | 22 | 195 | 48 / 147 |

### 3.3.2 Evaluation on Different Combinations

The combination of three ideal vector calculations and three similarity measures for classification were colour-coded and listed in Table 3.3. The experiments were performed using different combinations of the methods listed for ideal vector calculation and classification.

Table 3.3 Different combinations for classification

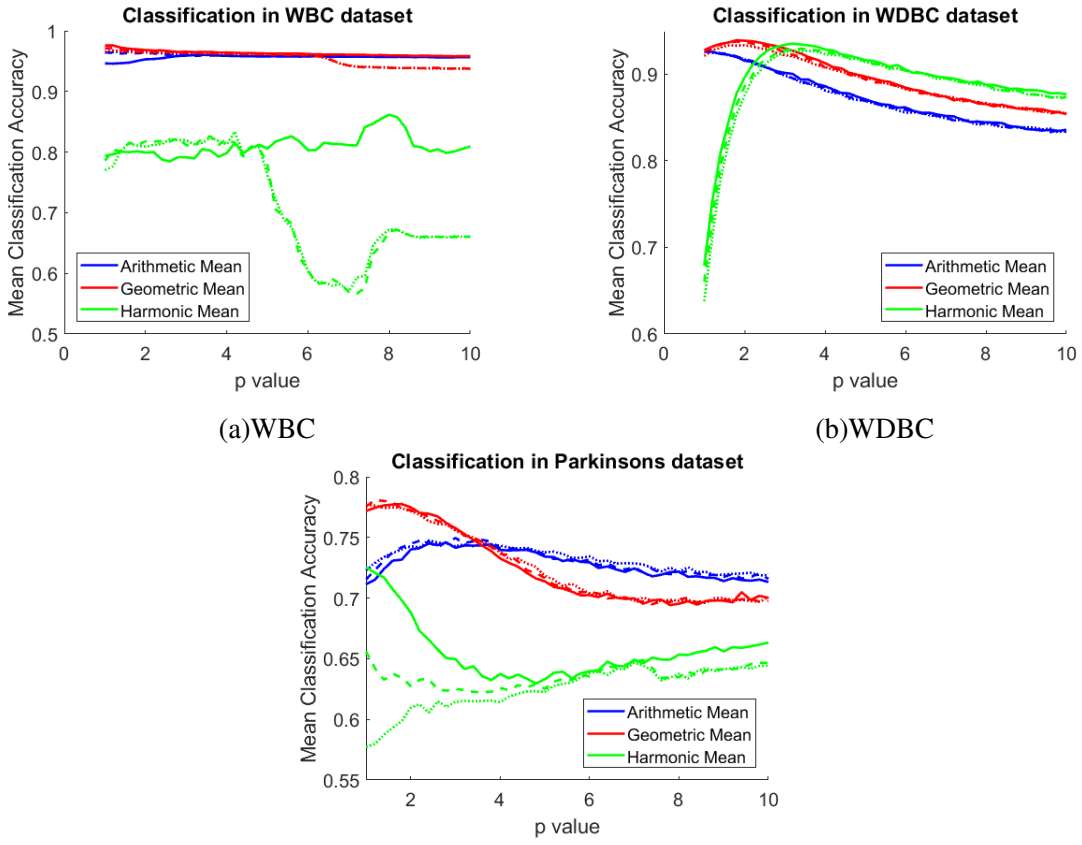
| **Ideal vector** | **Classification methods** | **Name** | **Line** |
|---|---|---|---|
| Arithmetic mean | Arithmetic mean | A-A | — |
| | Geometric mean | A-G | - - - |
| | Harmonic mean | A-H | ····· |
| Geometric mean | Arithmetic mean | G-A | — |
| | Geometric mean | G-G | - - - |
| | Harmonic mean | G-H | ····· |
| Harmonic mean | Arithmetic mean | H-A | — |
| | Geometric mean | H-G | - - - |
| | Harmonic mean | H-H | ····· |

**Experiment:** In this experiment, the complete sets of features were used for both training and testing without eliminating any features, which allows a fair comparison of different ideal vector calculations combined with different fuzzy similarity measures and the $p$ value (in Equation 3.4) optimization. The data samples in the three datasets (summarized in Table 3.2) are generally balanced for each class. Therefore, classification accuracy is utilized as the evaluation metric because it is easy to understand and performs well on balanced datasets.

The classification accuracy is defined as the number of correctly classified samples divided by the total number of samples.

Same as the evaluation in Luukka's work [138], the datasets were divided into two halves. One half was used for training and the other half for testing. Additional to the experiment in Luukka's paper [138], the experiments were repeated 1000 times for each $p$ value (in Equation 3.4) with random two-half group splitting. Note that all the remaining experiments in this chapter for classification accuracy calculation were tested based on the same evaluation mechanism, if not explicitly described.

**Result & Discussion:** The mean classification accuracy curves of the aforementioned combinations on three experimental datasets were plotted in Fig. 3.2.




(a)WBC (b)WDBC

(c)Parkinsons

Fig. 3.2 Mean classification accuracies with different $p$ values

Fig. 3.2-(a) shows the results of the mean classification accuracies in the WBC dataset. In this case, the ideal vector calculation using the arithmetic and geometric mean produced similar results, much higher and more stable than the harmonic mean. The curves of the harmonic mean methods vary dramatically when the $p$ value is greater than 5.

Fig. 3.2-(b) shows the mean classification accuracies for the WDBC dataset at different $p$ values. Different classification methods with the same ideal vector function produced similar performances. The accuracy using arithmetic and geometric mean methods in the ideal vector calculation process decreased slowly when the $p$ value increased. However, in the harmonic mean method, the accuracy increased sharply and peaked at $p = 3$. Subsequently, the mean accuracies decreased slowly along with the other two ideal vector calculation methods.

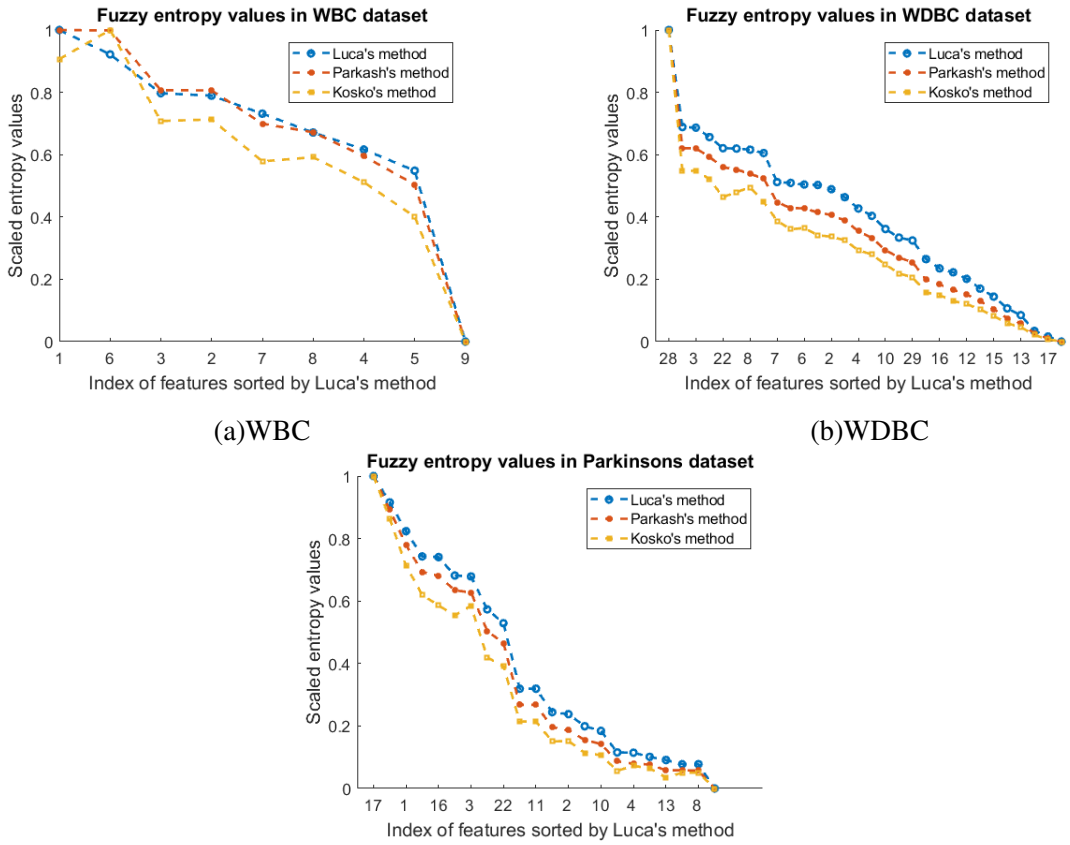
Fig. 3.2-(c) presents the results in Parkinsons dataset. The arithmetic means method on calculating the ideal vector produced a stable mean classification accuracy of around 0.73. The accuracy of the methods using the geometric mean for ideal vector calculation was maximized at the value around 0.78 and dropped quickly when $p$ was greater than 2. The harmonic mean methods on ideal vector calculation produced the worst and most inconsistent results.

Overall, the geometric mean method for calculating the ideal vector produced the maximal classification accuracies when $p$ was around two on all the experimental datasets. It may be because the geometric mean is a relative measure and gives more weight to the small feature values. When $p$ was around two, the geometric mean can represent the ideal vector's characteristics well, leading to better performance on classification accuracy. There are not many differences by using the three different similarity functions for classification. Therefore, geometric mean methods were utilized in the following experiments for ideal vector calculation and maximal similarity classification. The value $p$ in Equation 3.4 and 3.9 was set to be 2.

### 3.3.3 Evaluation on Fuzzy Entropy Methods

**Experiment:** This experiment compares the feature ranking sequences produced by three different fuzzy entropy methods in Section 3.2.3. The fuzzy entropy values ranked the features from the highest to the lowest. The entropy values were normalized into the range between 0 and 1 using min-max normalization to compare different methods.

**Result & Discussion:** For ease of comparison, Luca's method was chosen as the reference ranking sequence in the horizontal axis of Fig. 3.3. All the indices of features were sorted based on the reference ranking.




(a)WBC (b)WDBC

(c)Parkinsons

Fig. 3.3 Scaled entropy values of the sorted features

It is observed from Fig. 3.3 that different fuzzy entropy functions produced similar ranking sequences for the three datasets. Luca's and Parkash's methods resulted in an almost

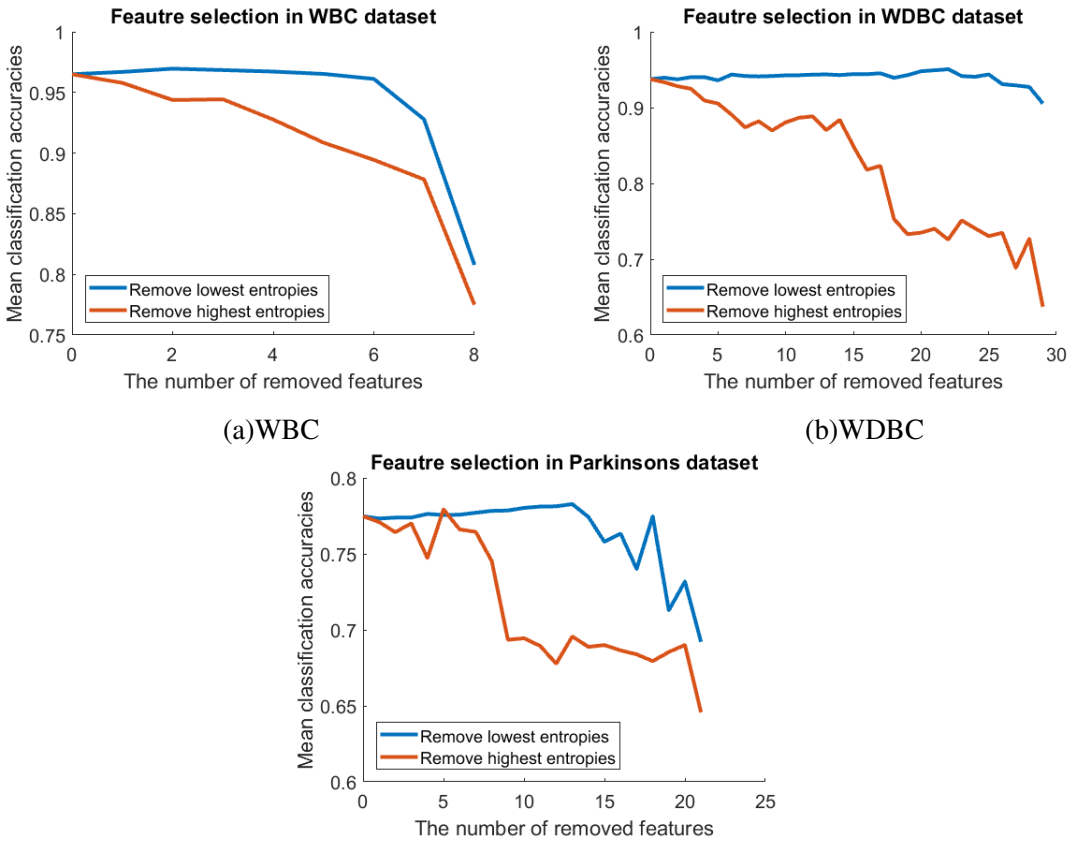
identical ranking sequence for all the datasets. Kosko's method showed disagreement at multiple points with the other two, especially in the Parkinsons dataset. In general, the three methods' ranking differences did not significantly impact the final classification performance from the experimental results. Therefore, to facilitate the performance comparison, Luca's method of the fuzzy entropy function was chosen in the final framework, which can produce a similar ranking sequence and consistent entropy values to the other two methods.

### 3.3.4 Evaluation on Removing Order of Fuzzy Entropy Methods

**Experiment:** To explore the optimal FS process, different feature removal orders and the FS approaches based on the entropy values were implemented and compared. One method removed the feature with the highest entropy value each time. On the contrary, the other method removed the feature with the lowest entropy value each time.

**Result & Discussion:** The mean classification accuracy curves with two different feature removal orders on the three experimental datasets are plotted in Fig. 3.4.

From Fig. 3.4, it can be seen that the method which removed the feature with the lowest entropy value each time produced a higher performance even when half of the features were removed on all the experimental datasets. In contrast, the performance dropped significantly on the experimental datasets once the feature with the high entropy value was removed. It is expected, as a higher entropy value indicates a larger variation in the feature values and more information than those with lower entropy values. Hence, the features with higher fuzzy entropy values are more informative and essential, and vice versa. Therefore, it can be concluded that the FS approach, which eliminates the feature with the lowest entropy value each time, produces better performance.




(a)WBC (b)WDBC

(c)Parkinsons

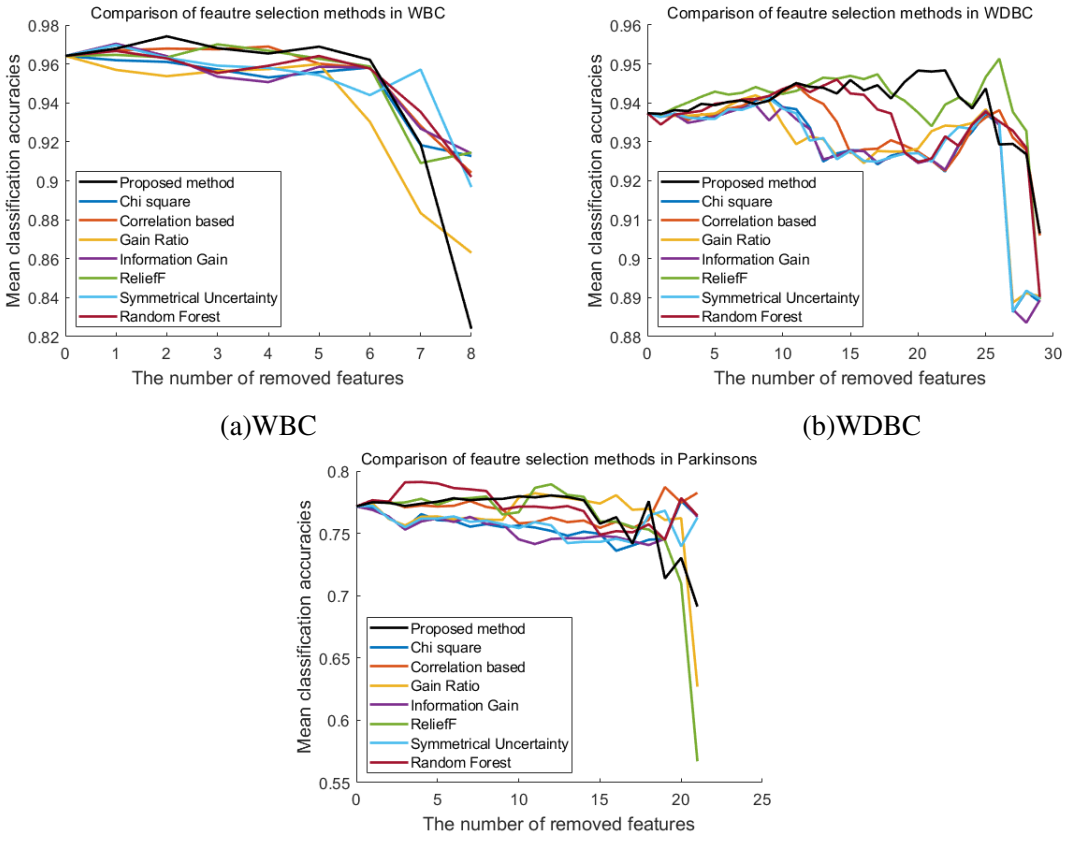
Fig. 3.4 Comparison of the different feature removing orders based on fuzzy entropy values

### 3.3.5 Performance Comparison with Other FS Methods

Based on the previous experiments' results, the optimized combination among different methods in the proposed FS and classification framework was found. The optimal choice and settings are the geometric mean method for ideal vector calculation and classification function with $p = 2$, and Luca's method for fuzzy entropy calculation. The proposed method was compared with six state-of-the-art filter-based FS methods and an embedded FS method after the paratermeters tuning process, which are Chi-Square based [95], Correlation based [73] , Gain Ratio based [104], Information Gain based [123], ReliefF based [134], Symmetrical Uncertainty based [227] and Random Forest (embedded FS method) [122].

**Experiment 1:** An experiment is performed by evaluating the mean classification accuracy while gradually removing the lowest-ranked feature each time. Two-fold cross-validation is implemented, while half of the dataset is for training and half data for testing. All the FS methods for comparison have been tuned with the comparatively optimal parameters and settings within the training process. The chosen compared FS methods rank the features from the higher to lower values. The same maximal similarity classifier with the geometric method was used for the downstream decision-making task after the FS process for all the compared methods.

**Result & Discussion 1:** The mean classification accuracy curves of different FS methods on three datasets are presented in Fig. 3.5.




(a)WBC (b)WDBC

(c)Parkinsons

Fig. 3.5 Comparison on mean classification accuracies of different FS methods

Fig.3.5-(a) shows that in the WBC dataset, the proposed method produces the highest classification accuracies among all the methods, with the number of removed features increasing from 0 to 6. In the WDBC dataset (Fig.3.5-(b)), the top two performers are the proposed method and the ReliefF method. The classification accuracies keep increasing even when about 20 features are removed using the proposed method. For the Parkinsons dataset (Fig. 3.5-(c)), the proposed method produces a stable performance with the arguably highest classification accuracies until 14 features are removed.

Another important observation is that the proposed method's classification accuracy generally follows the trend of gradually increasing, achieving peak performance, and decreasing when features were gradually removed for all three datasets. It is a good indication that the features were ranked reasonably well from the least to the most important. However, the performances of other methods changed dramatically when features were gradually removed (Fig. 3.5-(b) and Fig. 3.5-(c)).

**Experiment 2:** There is still a lack of standardised metrics for the performance comparison within different FS methods. One option from the literature is to report the highest classification accuracy despite the number of selected features. Alternatively, the classification accuracies are compared based on the same number of selected features. Arguably, if the classification result is essential, the first option should be applied. In this section, different FS methods were compared, where the compactness, representative, and relevance of the selected features were more important in this case. Therefore, the second option was adopted to compare different FS methods.

**Result & Discussion 2:** The proposed method was used as the reference method to compare with each of the other competitors. The number of selected features (denoted as $N_S$) with the highest mean classification accuracy was used as the reference. For other methods, the highest mean classification accuracies were reported with the selected number of features

less than or equal to $N_S$. For comparison, higher classification accuracy indicates better FS performance. The mean classification accuracies (Acc.) and the selected number of features (Nb.) of the three experimental datasets are shown in Table 3.4.

Table 3.4 Classification accuracy and feature numbers of different FS methods

| **Methods** | **WBC** | | **WDBC** | | **Parkinsons** | |
|---|---|---|---|---|---|---|
| | Acc.(%) | Nb. | Acc. (%) | Nb. | Acc. (%) | Nb. |
| Proposed | **96.97** | **7** | 94.86 | 8 | 78.23 | 9 |
| Chi Square | 95.86 | 7 | 93.67 | 5 | 77.70 | 2 |
| CFS | 96.95 | 7 | 93.86 | 4 | 78.72 | 3 |
| Gain Ratio | 95.83 | 6 | 93.73 | 5 | 78.09 | 6 |
| Info. Gain | 96.53 | 7 | 93.71 | 5 | 77.43 | 2 |
| ReliefF | 96.96 | 7 | **95.21** | **4** | 78.26 | 8 |
| Sym. Unc. | 95.84 | 7 | 93.68 | 5 | 77.19 | 3 |
| Random Forest | 96.52 | 7 | 94.13 | 8 | **79.10** | **4** |

Additionally, McNemar's test [13] was applied to test the statistical significance of the binary classification results for each of the two compared methods. The $P$ values of McNemar's test for the pairwise tests between the proposed method and each of the competitors are presented in Table 3.5.

Table 3.5 $P$ values of McNemar's test for the pairwise tests between the proposed method and each of the competitors

| **Methods** | **WBC** | **WDBC** | **Parkinsons** |
|---|---|---|---|
| Chi square | <0.01 | <0.01 | <0.01 |
| CFS | 1.00 | <0.01 | <0.01 |
| Gain Ratio | <0.01 | <0.01 | <0.01 |
| Info. Gain | <0.01 | <0.01 | <0.01 |
| ReliefF | 1.00 | <0.01 | <0.01 |
| Sym. Unc. | <0.01 | <0.01 | <0.01 |
| Random Forest | <0.01 | <0.01 | <0.01 |

For the WBC dataset results in Table 3.4, it is observed that the proposed method produced the best mean classification accuracy compared with other methods with $S = 7$. In Table 3.5, the proposed method is statistically better than Chi-Square, Gain Ratio, Information Gain, Symmetrical Uncertainty and Random Forest methods but no statistical differences to the CFS and ReliefF methods. For the WDBC dataset, the proposed method produced the

second-best classification accuracy with $S = 8$. The other methods produced an individually higher performance using about 4 or 5 features rather than eight features.

However, from Fig. 3.5-(b), the proposed method still produced the second-best performance if five features were used (the value corresponds to 26 in the horizontal axes of Fig. 3.5-(b)). According to Table 3.5, the proposed method was statistically worse than the ReliefF method but statistically better than the other methods. For the Parkinsons dataset, the proposed method ($N_S = 9$) ranked fourth, which was statistically worse than the CFS (3 features), ReliefF method (8 features) and Random Forest (4 features). The other methods were statistically worse than the proposed method.

In principle, the fuzzy entropy values can provide objective evidence on the features' importance. It will benefit and guide the FS process, especially when combined with the fuzzy similarity classifier. The features with higher fuzzy entropy values are found to represent the information of the data better than the features with lower fuzzy entropy values. Consequently, it results in a better performance for classification. Moreover, the calculation of fuzzy entropy values is independent among different features, which leads to a much more stable performance when calculating the classification accuracy.

## 3.4 Summary

In this chapter, based on Lukka's fuzzy entropy FS framework, different methods were implemented using the framework's key components, including the combinations of using three ideal vector calculations, three maximal similarity classifiers, and three fuzzy entropy functions. All the evaluations were performed on three widely used publicly available datasets with different data sparsity. The experiments were thoroughly evaluated by evenly and randomly splitting the dataset into the training and testing group and repeated 1000 times. The experimental results have shown that using the geometric method for ideal vector calculation ($p = 2$), the geometric method for similarity classifier ($p = 2$), and Luca's

method for fuzzy entropy calculation have produced the most stable performance and highest classification accuracy. Additionally, it can achieve better performance by removing the features with the lowest entropy values.

The proposed method was further compared with the other seven state-of-the-art filter-based FS methods. The mean classification accuracies were evaluated using the same number of selected features. McNemar's test was applied to evaluate the statistical differences for the pairwise comparisons. The proposed method produced the highest classification accuracy for the WBC dataset, the 2nd and 4th best for WDBC and Parkinson's datasets. The proposed method, ReliefF and Random Forest were the top performers among the compared methods. More importantly, the proposed method produced the stablest performance on the experimental datasets when the features were gradually removed.

To sum up, this chapter represents a very early work in this thesis and relates to other chapters on various aspects. The fuzzy-entropy based FS method was an initial investigation to develop an FS method for a given dataset. The method has produced a reasonably good and stable performance on classification. However, there are still many limitations in the research on finding the most suitable feature ranking sequence for a given dataset. Firstly, it is still not straightforward to evaluate and compare the overall performance of different FS methods based on multiple classification accuracy values produced by different feature subsets. Therefore, it suggests the need to establish better evaluation methods for method comparison. This leads to the proposed evaluation methods in Chapter 4. Secondly, the proposed method does not always achieve the best performance on all tested datasets, which could be improved by introducing more advanced techniques. Hence, other kinds of FS frameworks are explored in the later chapters. For instance, by combining different FS methods into a single framework, ensemble learning is investigated in Chapter 5. Moreover, the idea of selecting the best combination method in this chapter leads to the proposed meta-learning framework in Chapter 6.

# Chapter 4

# Evaluation of Feature Selection Methods

## 4.1 Introduction

As demonstrated in Chapter 3, a good evaluation method is an essential part for comparing different FS methods and subsequently guiding the development of effective FS algorithms. In the literature, evaluation metrics are used to assess the FS methods' performance, and various measurements have been proposed to evaluate different algorithms from multiple aspects, such as effectiveness and usefulness. Research findings indicate that FS methods' performance varies that is highly dependent on the applied evaluation metrics [188]. The evaluation and comparison process among different FS methods can help better understand the internal mechanism and recommend a suitable algorithm. Besides, the research on evaluation methods can also help establish a theoretical and practical basis to improve the FS methods' performance. Thus, selecting a suitable evaluation metric becomes essential for choosing the most appropriate FS method. Many previous studies normally focus on evaluating an algorithm from a single aspect, such as predictive performance, stability, etc [181, 151]. Some other issues also remain to be addressed, as discussed below.

Firstly, an FS algorithm's predictive performance is typically dependent on a specific classifier. By gradually removing the least important features, the performance curves are

constructed with a reducing number of features. In practice, various metrics have been utilized to evaluate the overall FS methods' performance, such as the maximum and average [187] of the measured values of the whole performance curve. In this case, these methods are not sensitive enough to capture the dynamic performance of the feature ranking sequence.

Secondly, some evaluation measures can help assess the reliability and consistency of the FS methods' output using different training subsets sampled from the same data distribution [102]. In the literature, various measurements have been proposed to evaluate the FS algorithms' stability, such as weights-scores, ranks or a selected feature subset [102]. Research on stability measures mainly focuses on evaluating the variation of the results produced by the algorithm on multiple runs on randomly sampled data. Instability arises when little agreement occurs over the selected features in these multiple runs, which prevents a correct and sound interpretation. Besides, as it reduces the trust towards the selected features, instability can also have a strongly impact to the validation process by domain experts. In the biomedical field, the experts may prefer a more stable FS algorithm over an unstable and even slightly more accurate one [102, 179, 71]. Practically, a reliable FS method should produce a consistent feature ranking sequence even using a data subset [70]. However, to the author's best knowledge, the ability to maintain consistent results across different data sizes has not been proposed as an evaluation metric.

Thirdly, most metrics only evaluate FS algorithms from independent aspects. However, an FS method with good predictive performance may not necessarily lead to superior performance on stability. Those disparate measurements and evaluation metrics are insufficient and inefficient to assess FS methods' overall performance. In the case of choosing only one FS algorithm out of a set of candidate techniques, the decision can not only be taken based on a single metric, such as accuracy, as it may have low stability at the same time [191]. It becomes challenging to compare and conclude FS's performance using multiple independent

assessments. Hence, it is in demand to quantify the methods' performance from different aspects but represented by a single measurement value.

As declared in Section 1.2, one of the objectives in this thesis is to review the existing evaluation metrics for FS method comparison and propose new and suitable metrics when necessary. After reviewing the existing evaluation metrics for FS method comparison, this chapter supplements FS evaluation metrics' research with the following contributions: (1) proposed a weighted accuracy evaluation metric to comprehensively measure the FS methods' predictive performance within a normalized value range; (2) proposed a novel measurement to infer the FS methods' robustness based on the evaluation using different size of datasets; (3) designed a multi-criteria evaluation metric to comprehensively measure the FS methods' overall performance on three independent aspects: accuracy, stability and robustness.

## 4.2 Datasets

### 4.2.1 Real-world datasets

Many real-world datasets with different characteristics and properties are employed to evaluate and compare FS methods' performance. Those datasets are drawn from the UCI machine learning repository for their applicability in the literature [46]. Twenty widely used public datasets with a different number of classes, features and samples are utilized in this research, with the overall descriptions shown in Table 4.1. The total number of classes (#C), number of features (#F), number of samples (#S) and the sample distribution per class (#S Distribution over #C) are shown in the table.

Table 4.1 General description of the real-world datasets

| No. | Datasets | *#C* | *#F* | *#S* | *#S Distribution over #C* |
|---|---|---|---|---|---|
| 1 | Banknote | 2 | 4 | 1372 | 610 / 762 |
| 2 | Mammographic | 2 | 5 | 830 | 403 / 427 |
| 3 | PIMA | 2 | 8 | 768 | 268 / 500 |
| 4 | WBC | 2 | 9 | 682 | 239 / 443 |
| 5 | CMSC | 2 | 18 | 540 | 46 / 494 |
| 6 | Statlog Heart | 2 | 13 | 270 | 120 / 150 |
| 7 | WDBC | 2 | 30 | 569 | 212 / 357 |
| 8 | Sports Articles | 2 | 59 | 1000 | 365 / 635 |
| 9 | Appendicitis | 2 | 7 | 106 | 21 / 85 |
| 10 | BCC | 2 | 9 | 116 | 52 / 64 |
| 11 | Wine | 3 | 13 | 178 | 48 / 59 / 71 |
| 12 | Parkinsons | 2 | 22 | 195 | 48 / 147 |
| 13 | Glass | 6 | 9 | 214 | 9 / 13 / 17 / 29 / 70 / 76 |
| 14 | Spectfheart | 2 | 44 | 267 | 55 / 212 |
| 15 | Breast Tissue | 6 | 9 | 106 | 14 / 15 / 16 / 18 / 21 / 22 |
| 16 | Dermatology | 6 | 34 | 358 | 20 / 48 / 48 / 60 / 71 / 111 |
| 17 | Sonar | 2 | 60 | 208 | 97 / 111 |
| 18 | Musk | 2 | 166 | 476 | 207 / 269 |
| 19 | Colon Cancer | 2 | 2000 | 62 | 22 / 40 |
| 20 | Lung | 5 | 3312 | 203 | 6 / 17 / 20 / 21 / 139 |

Many practical datasets are applied to evaluate the FS methods in diverse application scenarios. A brief introduction of these datasets is listed below.

1. *Banknote:* This dataset originated from the banknote-like specimens' images. The features were extracted by wavelet transform tools using statistical characteristics of the transformed images, such as variance, skewness, entropy, etc [46].

2. *Mammographic:* This dataset contains the patients' age, an assessment, and three attributes from BI-RADS [1]. It can be used to predict the severity of a mammographic mass lesion [52].

[1]BI-RADS stands for Breast Imaging Reporting and Data System

3. *PIMA (Pima Indians Diabetes Dataset):* This dataset contains the information of persons with or without diabetes based on the medical predictor features, such as the number of pregnancies, BMI, insulin level, age, etc [192].

4. *WBC (Wisconsin Breast Cancer):* This dataset predicts benign or malignant cancer with nine visually assessed features. After removing the NaN valued samples inside, the number of samples becomes 682 afterwards [216].

5. *CMSC (Climate Model Simulation Crashes):* This dataset contains the records of simulation crashes encountered. Its goal lies in predicting the climate model simulation outcomes using the normalized climate model input parameters [137].

6. *Statlog Heart:* This dataset contains information on the presence of heart disease in the patient. It is the reduced-sized version of Heart Disease databases [24].

7. *WDBC (Wisconsin Diagnostic Breast Cancer):* This dataset consists of the features from a digitized image of a breast mass to describe the cell nuclei's characteristics in a fine needle aspirate [197].

8. *Sports Articles:* This dataset presents 1000 sports articles that Amazon Mechanical Turk labels as objective or subjective. Their corresponding raw texts, extracted features, and URLs were retrieved and provided in the data [72].

9. *Appendicitis:* This dataset presents seven medical measures which are taken from 106 patients. The class labels represent whether the patient has appendicitis or not [46].

10. *BCC (Breast Cancer Coimbra):* This dataset provides the observed clinical features for 64 patients with breast cancer and 52 healthy controls [158].

11. *Wine:* In the dataset, the origin of wines is estimated using chemical analysis. Thirteen features are extracted from three types of wines by the analysis process [2].

12. *Parkinsons:* This dataset consists of the biomedical voice measurements from people with or without Parkinson's disease (PD). The data can be used to discriminate the persons with PD from the healthy people [133].

13. *Glass:* This dataset introduces six types of glass in terms of their oxide content, such as refractive index, sodium, magnesium, aluminium, silicon, etc [53].

14. *Spectfheart:* This dataset describes and diagnoses cardiac SPECT [2] images. The data consists of 2 classes, 267 samples with 44 continuous features, which are extracted and summarized from the images [121].

15. *Breast Tissue:* This dataset comprises the impedance measurements on freshly excised breast tissue at the frequencies of 15.625, 31.25, 62.5, 125, 250, 500, and 1000 KHz [98].

16. *Dermatology:* This dataset contains 34 features that all share the clinical characteristics of erythema and scales with minor differences. The samples are classified into six different groups based on the diseases [66].

17. *Sonar:* This dataset consists of two major classes, which are mines and rocks. Each sample contains 60 features with a value ranging between 0 and 1. The data values represent the energy using a specific frequency band integrated over a given time period [63].

18. *Musk:* In this dataset, whether new molecules are musks or non-musks can be learned and predicted. The molecules are described using 166 features which depend upon the exact shape or the conformation [42].

19. *Colon Cancer:* It is collected from patients with colon cancer, where tumour biopsies show tumour negative and normal positive biopsies originated from health parts of colons of the same patients. The dataset comprises 40 colon tumour samples and 22 normal colon tissue samples, which are analyzed using an Affymetrix oligonucleotide array [4].

20. *Lung:* This dataset contains 203 samples within five classes. Initially, there are 12600 genes within each sample. After a preprocessing step that removes genes with standard deviations smaller than 50 expression units, a dataset with 3312 features is produced afterwards [28].

[2] SPECT stands for Single Proton Emission Computed Tomography.

### 4.2.2 Synthetic Datasets

This section introduces synthesized datasets to evaluate and compare the evaluation metrics' performance in different situations. Madelon datasets are chosen on account of their high flexibility and variability [69]. The detailed generation procedures are described below.

**1. Parameters within Madelon Dataset** Madelon datasets are designed to cover a wide range of values for a different number of classes, features, and samples. By implementing the methodology which is firstly proposed in the NIPS 2003 feature selection challenge [67], various kinds of Madelon datasets are generated by varying 11 different parameters, as listed in Table 4.2. The constructed datasets present high flexibility in the choices of the number of classes, features, and samples.

Table 4.2 Parameters for data synthesis using Madelon dataset

| Alias | Meaning |
|---|---|
| P1 | Number of Classes |
| P2 | Number of Useful Features *(initially drawn to explain the concept)* |
| P3 | Number of Redundant Features *(linearly dependent upon the useful features)* |
| P4 | Number of Repeated Features *(repeating P2 and P3 at random)* |
| P5 | Number of Useless Features *(Drawn at random regardless of class label)* |
| P6 | Number of Samples per Cluster |
| P7 | Number of Cluster per Class |
| P8 | Random Seed |
| P9 | Factor multiplying the hypercube dimension (class separation) |
| P10 | Fraction of y labels to be randomly exchanged (flip y) |
| P11 | Flag to enable or disable random permutations |

**2. Generation of Madelon dataset** A Madelon dataset is generated from an empty matrix by gradually adding useful, redundant, repeated, and useless features. As illustrated in [18], the data values are then distorted by adding the noise, flipping labels, shifting and rescaling during the generation process. The detailed generation procedures are shown in Fig. 4.1. In the framework, all the steps can be enabled or disabled independently. The generation process takes the following steps to draw a randomized dataset.

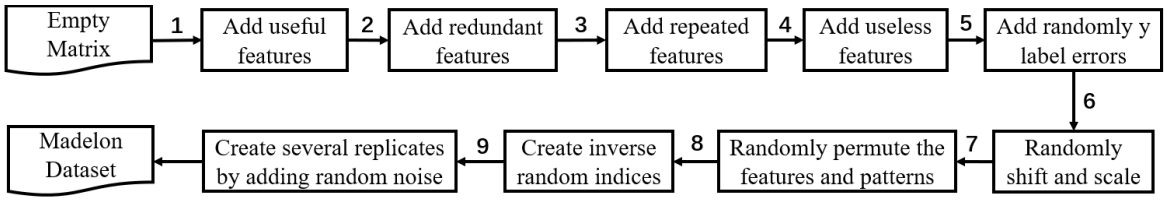



Fig. 4.1 The generation procedures of Madelon dataset

Step 1: Construct useful features:

- Add the empty matrix with a number of Gaussian clusters which are on the vertices of a hypercube in a subspace of dimensions using P2.
- Add some covariance to the previous generated Gaussian clusters by multiplying a random matrix with uniformly distributed random numbers between -1 and 1.
- Each Gaussian cluster is placed randomly on the hypercube vertices by adding or subtracting P9.

Step 2: To construct the redundant features, multiply the previous generated Gaussian clusters of P2 by another random matrix with uniformly distributed random numbers between -1 and 1.

Step 3: Draw repeated features randomly from useful and redundant features.

Step 4: Draw useless features with the number of P5 at random using the Gaussian distribution.

Step 5: Add y label errors randomly. A random fraction P10 of labels is exchanged.

Step 6: The features are shifted and rescaled randomly.

Step 7: The features and patterns are randomly permuted.

Step 8: Inverse random indices are created and applied on both features and labels.

Step 9: Create several replicates by adding random noise. Some replicates are introduced with a bit of random noise.

## 4.3 Review of Performance Evaluation Methods

Performance evaluation methods are the fundamental tools in the FS research. A reliable evaluation method is required to fairly compare different FS algorithms based on their performance. Hence, this section reviews different evaluation metrics from various aspects of the literature, such as predictive and stability measures.

### 4.3.1 Review on Predictive Measures

FS algorithms are typically evaluated with machine learning methods for the predictive performance on decision-making tasks such as classification and regression. The FS methods are ultimately used to improve the downstream decision-making performance with simpler models and fewer features. Therefore, this section reviews the predictive measures for FS methods in different aspects.

**1. Commonly Used Evaluation Metrics** Various evaluation metrics are employed to measure the predictive performance, including accuracy, F1 score, precision, recall, specificity, the area under the curve (AUC), etc [116]. From the literature, some representative evaluation metrics are introduced below.

- *Accuracy:* It is defined as the proportion of correctly classified samples and frequently reported in machine learning studies [29]. However, in the cases when the labels are imbalanced, the use of accuracy may lead to biased results [89].
- *Recall:* It is defined as the proportion of the positive samples which are correctly identified. It also provides an informative measure to understand the model's correctness, especially when the false-negative samples are costly, making it essential to focus on one class. However, it could also be biased by over-predicting the positive class [181].
- *Precision:* It is defined as the proportion of samples predicted as being in a positive class that was correct predictions. Unlike the metrics such as recall, this measurement

can avoid the defect caused by over-predicting the positive class [19]. However, the metric can also be maximized by predicting the positive class for a few of the highest confidence samples [181].

- *F1:* It is defined as the harmonic mean of precision and recall to overcome their shortcomings, respectively. On the other hand, the use of F1 may still be biased by the over-predicted positive class [19].
- *Kappa:* It is a concordance for categorical data to measure the agreement relative to what would be expected by chance. The kappa values range from -1 to 1, where 0 represents classifying randomly, and 1 indicates the perfect classification [181].
- *Root Mean Square Error (RMSE):* It measures the Euclidean distance between the predictions and the ground truth labels. Lower values mean better performance, while 0 indicates no error at all. This is commonly used to handle regression problems since it can be easily calculated for continuous labels [160].

**2. Accuracy Sequence Generation** In the FS research, the predictive performance of the algorithms is usually related to the number of selected features [90]. Different predictive performance is produced using various features for the specific FS method. Predictive accuracy sequence is normally calculated with the different numbers of removed features. The downstream decision-making technique is applied to calculate the predictive accuracy for each feature subset. The detailed procedures are generalized as below.

1. Divide the dataset into the training set and testing set by K-fold cross-validation;
2. Implement the FS method on the training set to rank the features;
3. Remove the least essential feature one by one. Train the model on the training set with the retained features and predict on the testing set;
4. For the different number of features, calculate the mean accuracy among all the folds.

The above procedures generate a series of accuracies using different numbers of features. The generated accuracy sequences can be demonstrated and illustrated by Fig. 4.2.

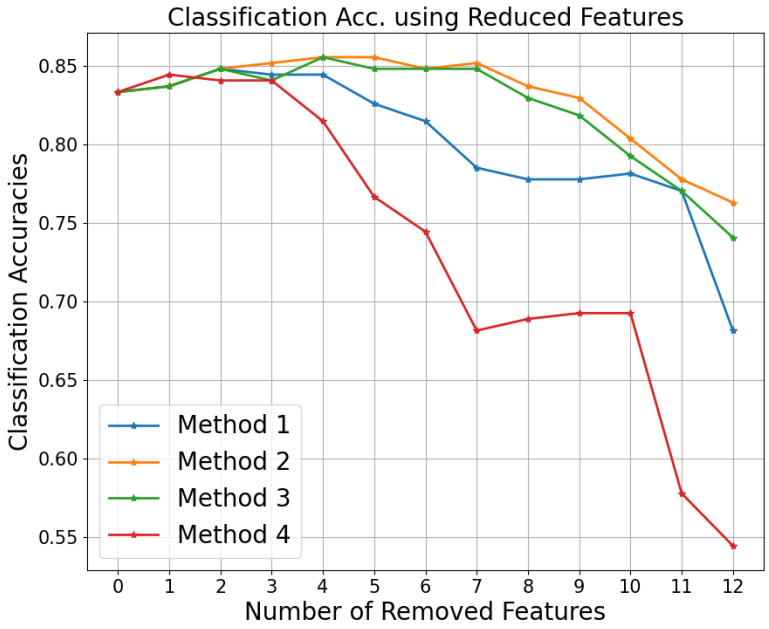



Fig. 4.2 Demonstration of accuracy sequences by different FS methods

**3. Evaluation Indices** As illustrated in Fig. 4.2, the generated accuracy sequences provide a clear and straightforward way to represent different FS methods' performance. However, it is still in great demand to evaluate the FS methods' performance using some evaluation indices. In the literature, some evaluation metrics are utilized to address these issues, such as the maximum or average [187]. Formally, given a training dataset $\mathbb{D}$ with $N$ features, the subsequent decision making result from a ranking-based FS algorithm can be represented as $Acc_0$, $Acc_1$, ..., $Acc_i$, ..., $Acc_{N-1}$, where $Acc_i$ stands for the classification accuracy with the number of $i$ features being removed.

• **Maximal Accuracy** It is defined to measure the overall maximal accuracy of the generated sequences.

$$Max\ Acc. = max\{Acc_0,\ Acc_1,\ ...,\ Acc_{N-1}\} \tag{4.1}$$

This measure helps compare the FS methods using their achieved maximal accuracy. However, many FS methods may result in the same maximal accuracy, indicating that the measure may not be effective enough to distinguish their performance.

• **Average Accuracy** It is defined to measure the overall average accuracy of the generated sequences, as expressed in the equation below.

$$Avg\ Acc. = \frac{\sum_{i=0}^{N-1} Acc_i}{N} \tag{4.2}$$

By taking the average accuracy value of the generated sequence, it can significantly reduce the possibility of producing the same evaluation index. However, the accuracy of using different feature subsets is contributed equally to the average score, which does not consider the number of retained features. Besides, those measures may not be sensitive enough to distinguish the different performances for certain datasets. Therefore, an evaluation method that can comprehensively and sensitively measure predictive performance is desirable.

### 4.3.2 Review on Stability Measures

**1. Background** Selecting a good FS method also requires a proper way to quantify the stability. In general, stability measures the sensitivity of an FS method's output on different training sets from the same distribution [102]. An FS algorithm's stability is related to the robustness of the feature preferences concerning the data's minor modifications. When the changes of the selected feature subsets are small, the method is then considered stable. On the other hand, instability refers to the fact that the selected features may drastically change even after marginal data modifications, or more generally, after some fine-tuning of the data production or analysis pipeline [74]. Unstable FS results may also easily lead to the

degraded performance in the final classifier due to the failure to identify the most relevant features [195]. Significant variations in the FS results signify the potential problems, which lead to less confidence by the domain experts.

The ranking-based FS algorithms can produce either features' ranking or weights, which typically assess the feature's importance in a predictive model. Each type of FS requires dedicated stability measures [74]. From the literature, many subset-based FS stability measures have been proposed, such as Kuncheva index [120] and Jaccard index [101]. Under such a profusion of different measures, it becomes challenging to justify a particular index and compare the results of works based on different metrics. Furthermore, the large number of available measures can also lead to inconsistent method comparison across different publications, where the researchers may select the index that makes their algorithm look the most stable [20].

**2. Evaluation Procedures** A stable method produces a consistent feature ranking result using different datasets sampled from the same distribution. Recent research mainly focuses on the stability indices and introduces various metrics such as Hamming distance, correlation coefficients, consistency and information theory [106]. A consistent feature ranking result and a fixed feature score are crucial in FS. By introducing the data perturbation using $K$-fold cross-validation, a widely used stability measure can be formalized below [188].

Firstly, the feature ranking sequence and the feature scores under different data subsets are firstly generated using $K$-fold cross-validation ($K$ is set as 10 in this research). Then, the average correlation values among the ranking sequence or scores are calculated by employing different evaluation metrics. The detailed procedures of the algorithm are shown as follows.

1. Divide the data into the training set and testing set using $K$-fold cross-validation.
2. Implement the FS method on the training set and produce the feature scores or feature ranking sequence.

3. Calculate the correlation between the feature scores or feature ranking sequences among different folds.

$$SV = \frac{\sum_{i=1}^{K-1}\sum_{j=i+1}^{K} S(\mathbf{f}_i,\, \mathbf{f}_j)}{K(K-1)/2} \tag{4.3}$$

where $S(.,.)$ represents the correlation coefficient using the chosen evaluation metric. $\mathbf{f}_i$ and $\mathbf{f}_j$ stand for the feature scores or feature ranking indices of the $i$th and the $j$th sample, respectively.

4. Average all pair-wise correlation values as the final result of stability.

**3. Similarity Measurement** The choice of a suitable correlation measurement to compare FS results' similarity remains one of the critical issues. For the ranking-based FS methods, the feature importance can be expressed using a series of feature scores. The ranking sequence is obtained by sorting the feature scores in descending order. More formally, given that a training dataset $\mathbb{D}$ with $N$ features, the result from a filter type FS algorithm can be expressed below [102].

$$\begin{aligned} &Scoring : \mathbf{s} = (s_1,\, s_2,\, ...,\, s_N),\; s_i \in S \subset \mathbb{R}^N,\; 1 \leq i \leq N \\ &Ranking : \mathbf{r} = (r_1,\, r_2,\, ...,\, r_N),\; r_i \in \{1,\, 2,\, ...,\, N\} \end{aligned} \tag{4.4}$$

where $s_i$ and $r_i$ represent the feature score and feature ranking index of the $i$th feature, respectively. Based on those different morphologies of FS results, the evaluation metrics are normally categorized into two approaches, which are feature scores based and feature ranking based. The former approach utilizes the feature scores in the similarity calculation process, such as the Pearson correlation coefficient. The latter approach evaluates the correlation values between the feature ranking indices to quantify the stability, such as Spearman's

rank correlation coefficient and Canberra distance [106]. The detailed information on those evaluation metrics is shown below.

• **Pearson Correlation Coefficient (PCC)** It is a statistical metric to evaluate the linear correlation between two variables using the following definition [11].

$$PCC(\mathbf{s},\mathbf{s}') = \frac{\sum_{i=1}^{N}(s_i-\bar{\mathbf{s}})(s_i'-\bar{\mathbf{s}}')}{\sqrt{\sum_{i=1}^{N}(s_i-\bar{\mathbf{s}})^2\sum_{i=1}^{N}(s_i'-\bar{\mathbf{s}}')^2}} \tag{4.5}$$

where $\mathbf{s}$ and $\mathbf{s}'$ represent two different feature score sequences with the mean values $\bar{\mathbf{s}}$ and $\bar{\mathbf{s}}'$. The values of *PCC* range between -1 and 1, while -1 and 1 indicating the perfectly anti-correlated and correlated respectively. The value 0 means no correlation.

• **Spearman's Rank Correlation Coefficient (SRCC)** It is a non-parametric approach to measuring the correlation between the ranking sequences and assessing the features' relationship using a monotonic function. The formal definition of this approach is shown below [232].

$$SRCC(\mathbf{r},\mathbf{r}') = 1 - 6\sum_{i=1}^{N}\frac{(r_i-r_i')}{N(N^2-1)} \tag{4.6}$$

where $\mathbf{r}$ and $\mathbf{r}'$ represent two feature ranking sequences with full sorted lists. The values of SRCC are in the range of [-1, 1], which represent perfectly anti-correlated (-1) and correlated (1), respectively. The value 0 means no correlation.

• **Canberra Distance (CD)** It is a weighted version of the classic $L1$ distance by measuring the ranked list's disarray. After dividing the absolute differences between ranks by their sums, the sum is calculated on $N$ different features, as shown in Equation 4.7 [99].

$$CD(\mathbf{r},\mathbf{r}') = \sum_{i=1}^{N} \frac{|r_i - r_i'|}{r_i + r_i'} \tag{4.7}$$

where the values of *CD* range from 0 to infinity. The smaller values mean a higher similarity between the ranked lists.

In general, the PCC assesses the linear relationship between the feature scores. The CD evaluates the absolute difference between the feature ranking sequences without a normalized value range. In comparison, the SRCC measures the rank order of the features with a monotonic relationship. Because of its normalized value range and robustness to the outlier, the SRCC is frequently applied as the evaluation metric in this research.

## 4.4 The Proposed Evaluation Methods

FS methods ultimately aim to improve the downstream decision-making performance (e.g., classification accuracy) with simpler models and fewer features. After the pre-processing process, the feature ranking sequence is obtained from the most to the least significant. FS methods' predictive performance is measured with different decision-making techniques by gradually eliminating the unimportant features. Different measures have been proposed, as reviewed in the previous section. However, many of them suffer from various shortages. Nevertheless, this section describes the proposed new evaluation metrics, which aim to comprehensively measure and evaluate the predictive performance of different FS methods. Specifically, the proposed measure is expected to have the following properties. (1) Uniformity: all the values are normalized into the range between -1 and 1 for a better comparison among different datasets. (2) Sensitivity: it is sensitive to the changes and variations of different feature ranking sequences [31].

### 4.4.1 Weighted Accuracy Measure

**1. General Procedures** In general, the proposed weighted accuracy is calculated in two steps. Firstly, as illustrated in Section 4.3.1, the performance sequences are generated by the decision-making algorithms on gradually reduced feature subsets. Then, the overall predictive performance is subsequently calculated in a weighted manner, which is defined as:

$$\begin{aligned} Weighted\ Acc. &= \frac{\sum Acc. * \%RemovedFeatures}{\sum \%RemovedFeatures} \\ &= \frac{\sum_{i=1}^{N} Acc_i * i}{\sum_{i=1}^{N} i} \end{aligned} \tag{4.8}$$

where $\%RemovedFeatures$ and $Acc.$ stand for the proportion of the removed features and the predictive accuracy, respectively. As illustrated in the "Accuracy Sequence Generation" step in Section 4.3.1, the predictive accuracy sequence is produced by removing the features from the least significant to the most. In Equation 4.8, higher weights or larger $\%RemovedFeatures$ values are assigned to the predictive accuracies with more important features. An FS method produces a higher weighted sum when the features are ranked correctly. Compared with the commonly used statistical measurements such as mean and maximum, the proposed evaluation metric becomes more reasonable and robust in evaluating the overall predictive performance.

**2. Overall Algorithm** The overall algorithm to measure the predictive performance is illustrated in Algorithm 3.

**Algorithm 3** The general algorithm for predictive accuracy evaluation

**Input:**

$N$: Total number of features;
$K$: Total number of folds for cross validation;
$FS$: The feature selection method to be evaluated;
$ML$: The down-stream machine learning method;

**Output:** $WA$: Weighted predictive accuracy

**Begin:**
**for** $n = 1$ **to** $N-1$ **by** 1 **do**:
    Split the data into $K$ equal size subset;
    **for** $k = 1$ **to** $K$ **by** 1 **do**:
        *Testing set*: The $k$th data subset; *Training set*: The remaining data subsets.
        $FR \leftarrow$ Implement $FS$ on *Training set* and produce the feature ranking sequence.
        *Reduced training/testing set* $\leftarrow$ Remove $n$ least important features by FR.
        Model with $ML$ on *Reduced training set* and predict on *Reduced testing set*.
        $Acc_{n,k} \leftarrow$ Calculate the predictive accuracy on *Reduced testing set*.
    $Acc_n = \sum_{k=1}^{K} Acc_{n,k}/K$
$WA = (\sum_{n=1}^{N-1} n * Acc_n)/(\sum_{n=1}^{N-1} n) = 2\sum_{n=1}^{N-1}[n * Acc_n]/[(N-1)N]$
**Return** $WA$

In general, weighted accuracy is informative and convenient for calculation. Because the accuracy values range between 0 and 1, it guarantees the uniformity and normalization of the measures. Besides, the principle within the proposed method is also simple and straightforward with high interpretability.

### 4.4.2 Robustness Measure

Unlike the stability measure introduced in Section 4.3.2 which evaluates the methods' equilibrium and tendency on recovering from perturbations, the robustness is defined to evaluate the ability of the FS methods on resisting bad cases or situations. From the perspective of domain experts, stable and robust FS algorithms are preferred when minor changes occur in the dataset. Robust methods can give domain experts more confidence in the selected features. Practically, a suitable FS method produces a consistent feature ranking sequence even using a data subset [70, 188]. Therefore, a new metric is proposed to measure FS methods' robustness using different data subsets.

There are two steps in the calculation process of robustness measures. Firstly, a similarity sequence is generated using the gradually reduced data's proportion. Afterwards, a single value is calculated using a weighted sum of the similarity sequence to represent the FS methods' performance on robustness. The detailed instructions are shown below.

**1. Similarity Sequence Generation** The similarity sequence is defined to evaluate the correlation coefficient of the produced feature scores or the ranking sequences between the full-sized and reduced-sized data. Various correlation metrics are utilized in the calculation process. Detailed procedures are listed below.

1. Randomly sample and construct the reduced datasets by gradually eliminating the proportion of $1-p$ samples, where $p = 1,\ 0.9,\ ...,\ 0.1$.
2. Repetitively apply the FS methods on the reduced sized data and calculate the corresponding FS results, i.e. feature score or feature rank sequence.
3. Calculate the similarity between the FS results on the full sized data ($p = 1$) and the reduced sized data ($p = 0.9,\ ...,\ 0.1$) using different evaluation metrics.
4. Repeat the calculation process $N$ times on the random permutations, and take the average as the output.

In the fourth step of the similarity sequence generation process, the calculation process is repeated $N$ times to avoid bias and occasional cases. More repeated times of the calculation process will lead to more objective results. By considering the efficiency requirement, this research empirically set $N$ to 10. Through implementing the above procedures, a series of correlation coefficients can be obtained using the different proportions of the removed data. Fig. 4.3 shows an example of the generated similarity sequence by using Spearman's rank correlation coefficient.

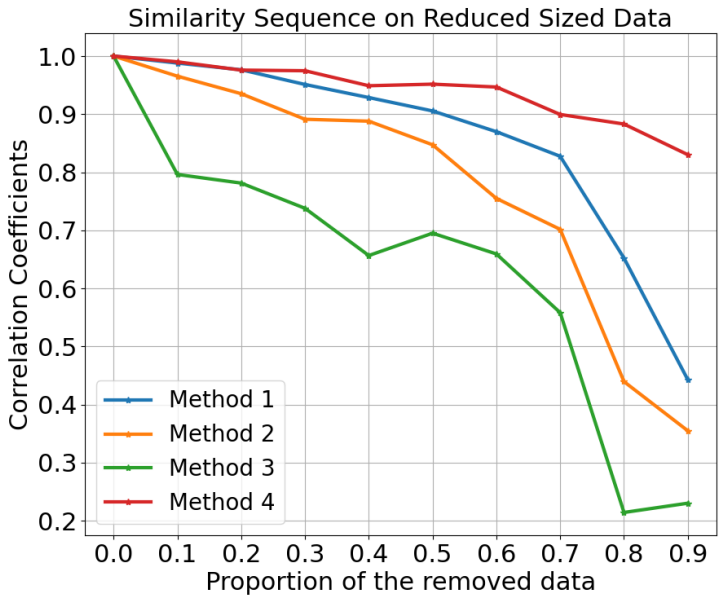



Fig. 4.3 Demonstration of similarity sequences using different proportions of removed data

**2. Weighted Robustness Calculation** A single value is computed to represent the overall performance on robustness with the following definition.

$$Weighted\ Robustness = \frac{\sum Corr * \%RemovedData}{\sum \%RemovedData} \tag{4.9}$$

where $\%RemovedData$ and $Corr$ represent the proportion of the removed data $(1-p)$ and the corresponding correlation values, respectively. Based on Equation 4.9, the correlation values with a larger proportion of the removed data will hold higher weights.

**3. Overall Algorithm** The procedures to measure the robustness of an FS method are shown in Algorithm 4.

This algorithm provides a new solution to measure FS methods' ability of producing stable FS results on various sizes of datasets. The correlation coefficient values range between 0 and 1. The evaluation process is also easy to be understood and can provide intuitive results, which proves to be interpretable.

**Algorithm 4** The general algorithm for robustness evaluation

**Input:**

$N$: Total number of the features
$K$: Total number of folds for cross validation
$FS$: Feature Selection method
$CorrEval$: Evaluation Metrics

**Output:** $WR$: Weighted robustness value

**Begin:**
**for** $random = 1$ **to** 10 **by** 1 **do**:
    Randomly permutate the data samples
    **for** $p = 1$ **to** 0.1 **by** -0.1 **do**:
        *data subset* $p \leftarrow$ The reduced datasets with the number of $p \times N$ samples
        feature ranks[p], feature scores[p] $\leftarrow$ implement $FS$ method on *data subset* $p$
        $corr[p]$ = $CorrEval$(feature ranks or scores[1], feature ranks or scores[$p$])
CorrM = mean value of $corr$ over different random permutations
$WR = [\sum_{p=0.1}^{1} CorrM[p] * (1-p)]/[\sum_{p=0.1}^{1}(1-p)]$
**Return** $WR$

### 4.4.3 Application Scenario

It can be seen that the proposed weighted accuracy and robustness measures are only applicable when the feature ranking sequence is generated. In the case of the weighted accuracy measure, the accuracy sequence is calculated by gradually removing the least essential feature one by one. Then the classification model is trained on the training set with the retained features and predicted on the testing set. Hence, the overall predictive performance is subsequently calculated in a weighted manner. Similar to the weighted accuracy, the robustness measure also needs to generate the feature ranking or feature score sequence during the similarity calculation process. Hence, for the method without a feature ranking process (baseline), the proposed evaluation metrics cannot be applied. This is also discussed in the method limitation section in Chapter 7.

## 4.5 Evaluation and Results

This section evaluates the performance of the proposed evaluation metrics. In comparison, the evaluation metrics are also illustrated and examined on both real-world and synthetic datasets. Some other widely-used standard measurements are also included and compared in this research.

### 4.5.1 Evaluation on Weighted Accuracy Measure

Based on the definition provided in equation (4.8), the accuracy values range from 0 to 1 ($0 \leq Acc. \leq 1$) and the following inequation can be derived.

$$0 = \frac{\sum_{i=1}^{N} 0 * i}{\sum_{i=1}^{N} i} \leq \frac{\sum_{i=1}^{N} Acc_i * i}{\sum_{i=1}^{N} i} \leq \frac{\sum_{i=1}^{N} 1 * i}{\sum_{i=1}^{N} i} = 1 \tag{4.10}$$

In Equation 4.10, when all the accuracy values become 0, the weighted accuracy is 0. On the other hand, the weighted accuracy becomes 1 when all the accuracy values equal 1.

**1. Case Study on the Iris Dataset** The Iris dataset is chosen to illustrate the advantages of using the weighted accuracy. This dataset contains three classes of 50 samples each, where each class refers to a type of iris plant. Four distinct features exist, including sepal length (F1), sepal width (F2), petal length (F3), and petal width (F4) [58]. To illustrate the results of the generated accuracy sequence, this section firstly implements 5 FS methods from the literature, including Infinite based FS (IFS) [175], Variance-based FS (VFS) [128], Trace-Ratio Criteria [149], Mutual Information based FS (MIFS) [231] and Correlation-based FS [73]. The feature ranking sequence (alias as "Set": 4, 3, 2, 1) is also utilized for performance comparison. During the experiments, KNN is implemented for the subsequent classification process, where K = 5. The corresponding accuracies using different FS methods are shown in Table 4.3.

Table 4.3 Feature ranking results on Iris dataset by different FS methods

| **Methods** | **Ranking** | | | | **Acc. with Removed F.** | | | | **Max.** | **Avg.** | **WA** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 0 | 1 | 2 | 3 | | | |
| IFS | 3 | 2 | 1 | 4 | .980 | .960 | .953 | .953 | .980 | .962 | .954 |
| VFS | 3 | 1 | 4 | 2 | .980 | .973 | .953 | .953 | .980 | .965 | .957 |
| TR | 3 | 4 | 2 | 1 | .980 | .967 | .960 | .953 | .980 | .965 | .958 |
| MIFS | 3 | 4 | 1 | 2 | .980 | .973 | .960 | .953 | .980 | .967 | .959 |
| Set | 4 | 3 | 2 | 1 | .980 | .967 | .960 | .960 | .980 | .967 | .961 |
| CFS | 4 | 3 | 1 | 2 | .980 | .973 | .960 | .960 | .980 | .968 | .962 |

In Table 4.3, the column "Ranking" represents the ranking sequence of the FS method from the most important to the least. The column "Acc. with Removed F." stands for the corresponding classification accuracy using KNN on the gradually removed feature subset. The column "Max." indicates the maximal accuracy of "Acc. with Removed F.". The column "Avg." stands for the average accuracy, while the column "WA" illustrates the proposed weighted accuracy for each of the FS methods.

From Table 4.3, it can be seen that different FS methods produce different feature ranking results. All the ranking results have the same highest accuracy with 0 feature removed. Hence, it indicates that the maximal accuracy is not able to distinguish the difference between different FS methods. Besides, the average value of the accuracy sequence can somehow avoid the previous issue to a certain degree. It averages the generated accuracy sequence and assigns each value using the same weights. However, the average accuracy shares the same result for the MIFS and Set methods. By inspecting the detailed accuracy values, the Set is better than MIFS because it produces an accuracy of 0.96 even when three features are removed. A similar problem can be observed for the methods of VFS and TR. In comparison, by assigning the accuracy values using different weights, the proposed weighted accuracy can significantly reduce the chance of repetition and score the feature ranking sequence sensibly. The results indicate that the weighted accuracy can distinguish all the differences between the feature ranking sequences in this example dataset.

**2. Comparison of Sensitivity on the Real-world Datasets** Several public datasets from the UCI machine learning repository are used to compare the evaluation metrics' performance. General information of those datasets is provided in Section 4.2.1. The datasets with only 5 to 8 features are chosen for illustration purposes by considering the computational efficiency, including Mammographic (Mammo.), Vertebral, Wholesale and PIMA. Given a dataset with $N$ features, there are $N!$ different feature ranking sequences (*#P.*), as shown in Table 4.4. In order to assess the sensitivity of different evaluation metrics on the accuracy, those measurements are applied to the datasets using all features' total permutations. Three widely-used machine learning methods are then employed in the subsequent decision-making process: K-Nearest Neighbors (KNN), Logistic Regression (LR) and Naive Bayes (NB). Then, the number of distinct values by the different evaluation metrics is counted and shown in Table 4.4.

Table 4.4 Numbers of distinct values using different evaluation metrics

| No. | Datasets | # F. | # P. | Classifier | Max | Avg. | WA |
|---|---|---|---|---|---|---|---|
| **1** | Mammo. | 5 | 120 | KNN | 8 | 104 | 113 |
| | | | | LR | 7 | 109 | 118 |
| | | | | NB | 3 | 99 | 109 |
| **2** | Vertebral | 6 | 720 | KNN | 6 | 347 | 536 |
| | | | | LR | 3 | 260 | 376 |
| | | | | NB | 6 | 300 | 513 |
| **3** | Wholesale | 7 | 5040 | KNN | 11 | 1148 | 2334 |
| | | | | LR | 15 | 928 | 2068 |
| | | | | NB | 13 | 1336 | 2669 |
| **4** | PIMA | 8 | 40320 | KNN | 27 | 25364 | 37198 |
| | | | | LR | 15 | 21686 | 36247 |
| | | | | NB | 43 | 23093 | 36803 |

In Table 4.4, the column "#F." represents the number of features for the dataset. The column "#P." indicates the number of total permutations of the feature ranking sequence for each dataset, which equals the factorial of the features' number. Based on the changes in total permutations, the number of distinct average accuracy values has increased drastically from around 100 to more than 40 thousand. However, the maximal accuracy could only produce a

tiny proportion of the total permutation numbers. The average accuracy has produced a much higher proportion of the distinct values in comparison. Moreover, the proposed weighted accuracy is able to produce significantly more distinct values than the maximal and average accuracy. It has increased more than 50% of the distinct values compared to the average accuracy.

**3. Comparison of Sensitivity on the Synthetic Datasets** This section utilizes the synthetic datasets to evaluate and compare the sensitivity performance of the weighted accuracy measures. As introduced in Section 4.2.2, Madelon datasets are employed with a different number of useful or useless features.

**3.1 Comparison using different numbers of useful features** Firstly, the sensitivity performance of different measures is evaluated on Madelon datasets with a different number of useful features. By fixing other parameters and varying the number of useful features from 5 to 25, various Madelon datasets are constructed correspondingly. Then, the standard deviation of the evaluation results is compared on those Madelon datasets using three accuracy measures, including maximal, average and weighted accuracy, as shown in Figure 4.4.

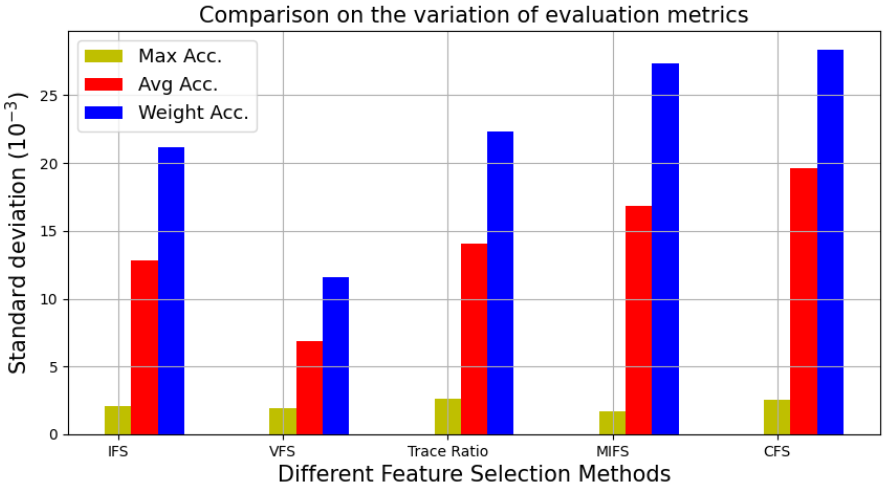



Fig. 4.4 Comparison of accuracy metrics on Madelon datasets with different useful features

**3.2 Comparison using different numbers of useless features** Secondly, the sensitivity performance of accuracy measures is also compared on the Madelon datasets using various useless features. Same with the case in the useful features, this experiment constructed different kinds of Madelon datasets by fixing the other parameters and varying the number of useless features from 0 to 30. Figure 4.5 compares the standard deviation of the accuracy measures using various FS methods on the Madelon datasets with 0 to 30 useless features.

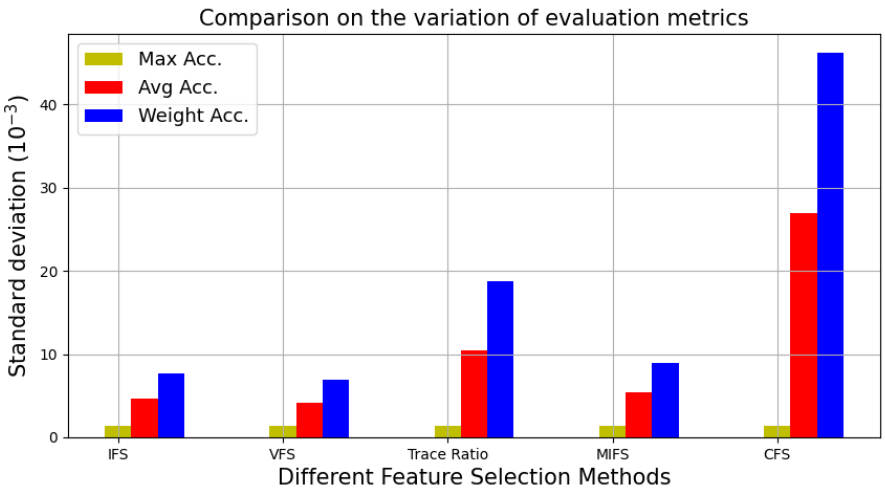



Fig. 4.5 Comparison of accuracy metrics on Madelon datasets with different useless features

It can be seen that the proposed weighted accuracy measure produced a much larger standard deviation using the 5 FS methods on different Madelon datasets with various useful and useless features. This indicates that the results of weighted accuracy can vary significantly with a wide value range. To a certain degree, the weighted accuracy measure proves to be more sensitive to the changes across different datasets.

### 4.5.2 Evaluation on Robustness Measures

The evaluation metric on stability measures the similarities between the ranked features across different folds in K-fold cross-validation in the literature. However, most measurements only focus on the performance variation in a fixed data size, which cannot depict the FS methods'

performance on the data with different sizes. Therefore, a new metric is proposed to evaluate the FS methods' performance using different sizes of the data subsets, which indicates the robustness of an FS method when handling data with varying sizes. The values' range of robustness is also dependent on the adopted measurements. Here, similarity measures are used to indicate the feature ranking differences when different sized data is used for evaluation. Many similarity measures range from -1 to 1 by definition, including the Pearson correlation coefficient and Spearman's rank correlation coefficient. Then the following inequation can be derived.

$$-1 \leq \frac{\sum_{p=0.1}^{1} CorrM[p] * (1-p)}{\sum_{p=0.1}^{1} 1-p} \leq 1 \tag{4.11}$$

From Equation 4.11, the robustness values range from -1 to 1, which ensures the measures' uniformity and normalization. $p$ is the proportion of data included in evaluation (e.g. $p$=0.1,0.2,0.3...,1). The following sections will discuss the sensitivity characteristics of the measurement.

**1. Case Study on the Iris Dataset** As introduced before, the performance on stability (often used in the literature) is not necessarily identical to robustness. Therefore, this section evaluates and compares the robustness measure with stability using different K values (i.e. the number of folds in cross-validation). Iris dataset is chosen as a case study on account of its simplicity and popularity. Five FS methods are included for comparison: IFS, VFS, Trace-Ratio Criteria, MIFS and CFS. Spearman's rank correlation coefficient (SRCC) is utilized to measure similarity. Table 4.5 compares the stability and robustness measures on the iris dataset. The stability within the FS methods is evaluated using a different number of folds (i.e. 2, 4, 5, and 10).

It can be seen that all 5 FS methods have achieved relatively high performance on both stability and robustness. The majority of their performance on stability remains to be one,

Table 4.5 Comparison of the stability and robustness on Iris dataset

| Methods | Stability with K | | | | Robust |
|---|---|---|---|---|---|
| | 10 | 5 | 4 | 2 | |
| IFS | 1 | 1 | 1 | 0.800 | 0.992 |
| VFS | 1 | 1 | 1 | 1 | 0.993 |
| TR | 1 | 1 | 1 | 1 | 0.996 |
| MIFS | 0.907 | 0.880 | 0.900 | 0.800 | 0.886 |
| CFS | 1 | 1 | 1 | 1 | 0.956 |

even using different K values. So from the perspective of stability, the chosen 5 FS methods have achieved very similar (some of them are identical) performance on stability. On the other hand, by measuring the performance using the different proportions of the removed data, the robustness measure can provide quantitative numbers to distinguish the performance of those FS methods. Therefore, the robustness measure proves to be a more sensitive and distinguishable evaluation metric for the iris dataset.

**2. Comparison of Sensitivity on the Real-World Datasets** The stability (Sta) and robustness (Rob) measures are further compared quantitatively using four public datasets from the UCI machine learning repository. Five FS methods, including IFS, VFS, TR, MIFS and CFS, are implemented for comparison. Stability is calculated using 10-fold cross-validation. To better quantify the sensitivity performance of the evaluation metrics, the standard deviation (STD) of the measures is reported, as shown in Table 4.6. Therefore, the larger the STD, the more sensitive one method is to the changes. The more sensitive, the more likely to distinguish the different performances across different methods.

In Table 4.6, it can be seen that different FS methods have produced diverse kinds of stability and robustness values. The robustness measure has always produced a larger STD value than the stability measure, which indicates a better capability of distinguishing the differences across different methods.

Table 4.6 Performance comparison of stability and robustness using different FS methods

| FS Methods | Datasets | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Mammo. | | Vertebral | | Wholesale | | PIMA | |
| | Sta | Rob | Sta | Rob | Sta | Rob | Sta | Rob |
| IFS | 1.00 | .940 | .989 | .947 | .976 | .887 | .958 | .911 |
| VFS | .940 | .811 | .973 | .956 | .981 | .966 | 1.00 | 1.00 |
| TR | .849 | .626 | .853 | .806 | 1.00 | .986 | .987 | .909 |
| MIFS | .947 | .881 | .797 | .580 | .891 | .821 | .817 | .671 |
| CFS | .964 | .925 | .844 | .360 | .929 | .654 | .992 | .930 |
| **STD** | .056 | .128 | .085 | .257 | .044 | .134 | .076 | .125 |

**3. Comparison of Sensitivity on the Synthetic Datasets** This section utilizes the synthetic datasets to compare the sensitivity performance of the robustness measures. Same as before, the Madelon datasets introduced in Section 4.2.2 are implemented here, with different numbers of useful and useless features.

**3.1 Comparison using different numbers of useful features** This section evaluates the sensitivity performance of stability and robustness measures on Madelon datasets with different numbers of useful features. By fixing the other parameters and varying the number of useful features from 5 to 25, various Madelon datasets are constructed accordingly. Figure 4.6 shows the standard deviation values of stability and robustness using different FS methods on the Madelon datasets with 5, 10, 15, 20 and 25 useful features.

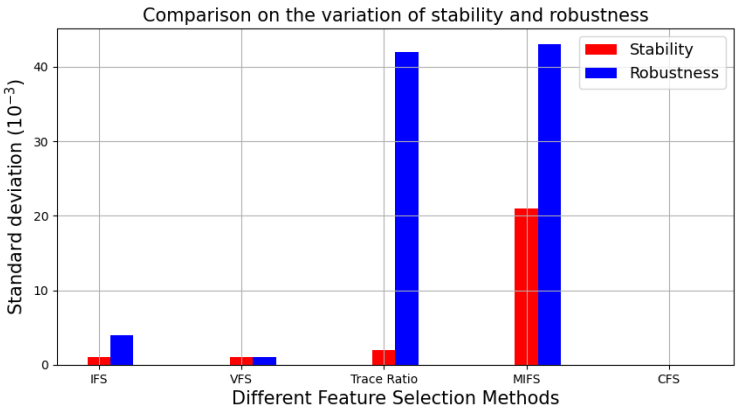



Fig. 4.6 Comparison of stability and robustness using different number of useful features

**3.2 Comparison using different numbers of useless features** The sensitivity performance of stability and robustness is also compared on the Madelon datasets with different numbers of useless features. Various Madelon datasets are generated by fixing the other parameters and varying the number of redundant features from 0 to 30 (i.e. 0, 5, 10, 15, 20, 25 and 30). Figure 4.7 shows the comparison of the standard deviation values produced by different FS methods.

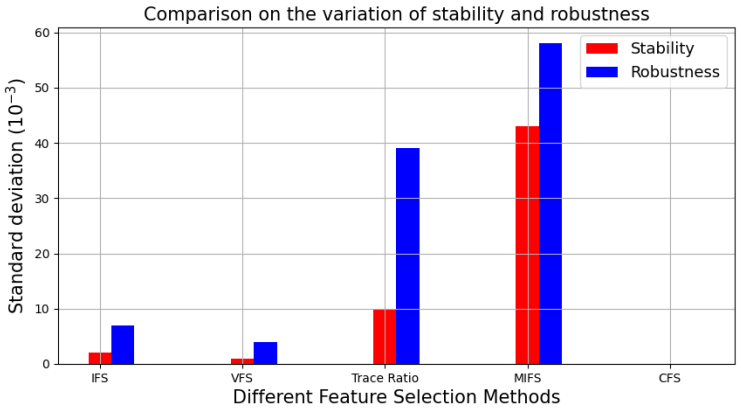



Fig. 4.7 Comparison of stability and robustness using different number of useless features

In Figure 4.6 and Figure 4.7, the robustness measure has achieved a significantly higher standard deviation for the methods of IFS, VFS, Trace Ratio and MIFS methods. Besides, the CFS method has consistently produced zero STD values for both stability and robustness measures on the Madelon datasets. This is due to the CFS method producing exactly the same feature ranking sequence in the experiments. In conclusion, the proposed robustness measure has a better capability than the stability measure in capturing the differences between different FS methods in different training scenarios.

## 4.6 Multi-Criteria Measurement and Runtime Analysis

### 4.6.1 Background

The previously proposed evaluation metrics only provide individual aspects of evaluating FS algorithms. However, those disparate measurements and individual criteria evaluation metrics are insufficient and inefficient to illustrate FS methods' overall performance. Hence, it is in demand to quantify the methods' performance to drill down into different aspects at various levels of detail. For this purpose, a multi-criteria measurement is proposed to evaluate the comprehensive performance of FS methods. Therefore, a multi-criteria metric is developed in relation to each measurement. By integrating the separate evaluation metrics, this section utilizes a radar chart to evaluate and compare different FS methods' performance comprehensively, as shown in Fig. 4.8.

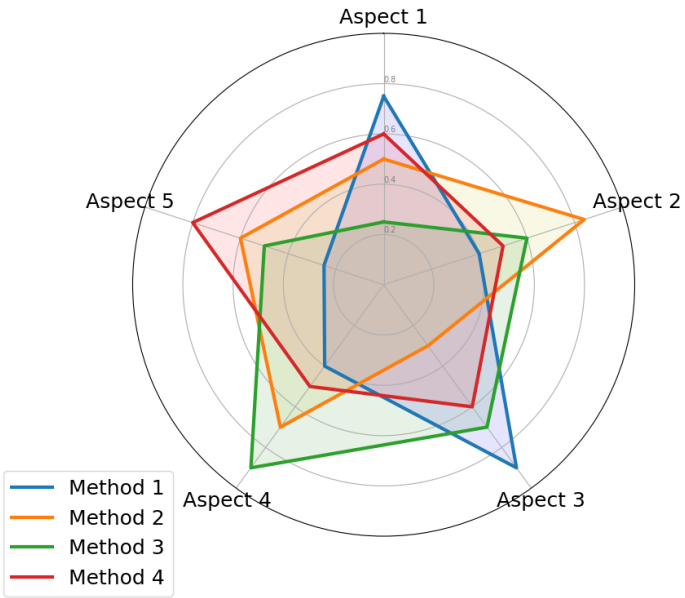



Fig. 4.8 Multi-criteria measurement using the radar chart

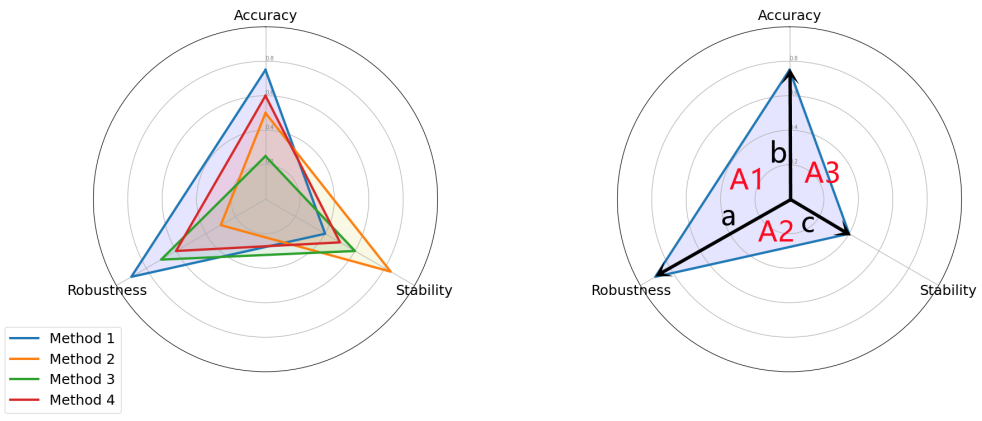
A radar chart is a two-dimensional chart representing the multivariate data using the axes from the same starting point [165]. In Fig. 4.8, the FS methods' performance is represented by the polygons with different colours, and each axis indicates a one side aspect of the

method. By comprehensively considering the multiple aspects of FS algorithms, the overall performance is measured using the polygon area.

### 4.6.2 General Procedures of Multi-Criteria Measure

Radar charts are used to compare the performance of FS methods on multi-dimensional aspects. When different radar charts' axes represent a single property, the relative performance distributions of the FS methods are characterized using the geometries of their property profiles [156]. In this research, FS methods' performance is depicted using triangles with three independent axes, i.e., accuracy, stability, and robustness. The detailed instructions are shown below.

**Step 1: Radar Chart Construction** In the beginning, the FS methods are implemented on the datasets to calculate the corresponding three independent evaluation metrics in Section 4.4. The radar chart is constructed subsequently using the independent axes, as shown in Fig. 4.9. In Fig. 4.9, the left subgraph demonstrates the performance comparison among different FS methods, and the right subgraph inspects a single radar chart in detail.




(a) Multiple Methods (b) Single Method

Fig. 4.9 Multi-criteria measurement using the aspects of accuracy, stability, and robustness

**Step 2: Normalized Area Calculation** The normalized area of a radar chart is used to represent the comprehensive performance of the FS method. From Fig. 4.9-(b), the total area of the radar chart consists of three small triangles, which are $A1$, $A2$, and $A3$. The area of those small triangles formed from the meeting edges with vertices is calculated below ($A1$ for example).

$$Area(A1) = \frac{a * b * sin(2\pi/3)}{2} \tag{4.12}$$

where $a$ and $b$ denote the two sides of the triangle. For each small triangle, the intersection angle is $2\pi/3$. Hence, the normalized area with the values ranging between 0 and 1 is calculated in Equation 4.13.

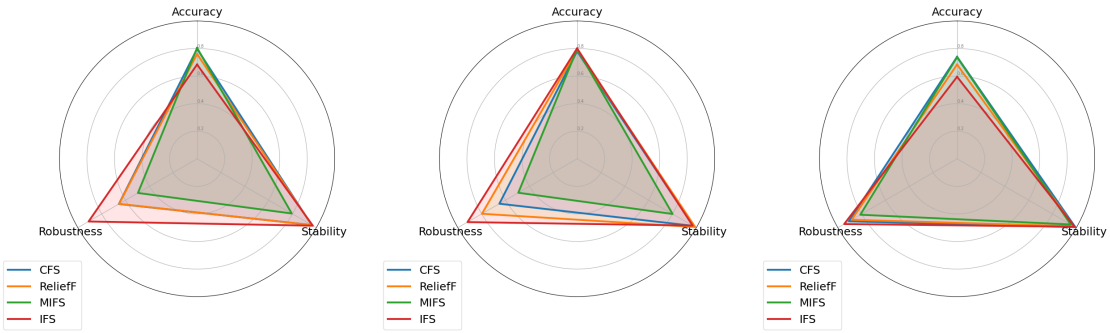
$$norm\ Area = \frac{1/2 * (a * b + b * c + a * c) * sin(2\pi/3)}{1/2 * 3 * sin(2\pi/3)} = \frac{ab + bc + ac}{3} \tag{4.13}$$

For the given FS algorithm, the area's normalized value reflects the method's overall quality. A higher value indicates a better FS performance from a comprehensive perspective.

### 4.6.3 Demonstration of Multi-Criteria Measurement

Four state-of-the-art FS methods were implemented on three public datasets (Statlog Heart, Parkinsons and Dermatology) to investigate the utility of the multi-criteria measurement, which includes CFS, ReliefF, MIFS and IFS methods. KNN is chosen as the classifier (K = 5). Spearman's rank correlation coefficient was used as the similarity measurement. The graphical representation of the constructed radar charts is shown in Fig. 4.10.

In Fig. 4.10, different FS methods propose the radar charts using various shapes. Hence, the normalized area for each radar chart is calculated to compare and discriminate the corresponding performance of the different FS approaches, as shown in Table 4.7.




(a) StatlogHeart (b) Parkinsons (c) Dermatology

Fig. 4.10 Radar charts of multi-criteria performance on different datasets

Table 4.7 Performance comparison of the normalized area of radar chart

| **FS Method** | **Datasets** | | |
|---|---|---|---|
| | Statlog Heart | Parkinsons | Dermatology |
| CFS | 0.557 | 0.602 | **0.791** |
| ReliefF | 0.567 | 0.703 | 0.700 |
| MIFS | 0.434 | 0.449 | 0.716 |
| IFS | **0.695** | **0.764** | 0.644 |

In Table 4.7, each FS method under the different datasets has scored a value to represent their overall performance. The normalized area evaluation metric has provided a new and comprehensive solution to compare the different FS methods' performance.

The multi-criteria evaluation metric is adopted as a critical evaluation index in the later chapters. It is utilized to comprehensively evaluate the performance of different FS methods. Considering the multiple aspects of the FS methods, such as accuracy, stability, and robustness, it provides a comprehensive view when comparing the FS methods' performance. However, many issues still need to be addressed to thoroughly investigate the proposed normalized area measure's utility. It would require additional work to verify it fully, which will be discussed in Chapter 7.

### 4.6.4 Run Time Analysis

This section evaluates and analyses the run-time of the proposed evaluation metrics, including weighted accuracy, robustness, and multi-criteria performance. Then, the performance of the CFS method is evaluated on the Madelon datasets with different numbers of useful features, where the other parameters are fixed. The programs were implemented using Python and ran on a laptop with 2.6GHz, Intel(R) Core(TM) i7-10750H CPU, and 16GB RAM. The average execution time (second) is presented in Figure 4.11 respectively, by running the evaluation process ten times for each generated Madelon dataset.

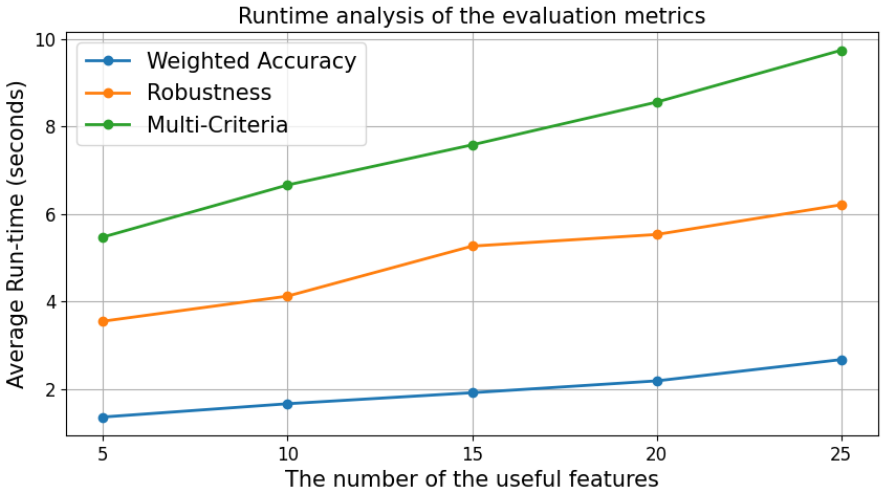



Fig. 4.11 Runtime analysis on Madelon datasets with different numbers of useful features

It can be seen that the calculation of the weighted accuracy takes the least time, followed by robustness and multi-criteria performance. It should be mentioned that the run-time of the weighted accuracy also depends on the implemented classifier. With the addition of useful features in Madelon datasets, the run-time of all the evaluation metrics increases linearly. Due to its straightforward calculation process and low computational cost, it can be easily applied and scaled to higher-dimensional datasets.

## 4.7 Summary

This chapter has introduced the materials and evaluation methods of this research. After reviewing the existing performance evaluation metrics for FS method comparison, some new evaluation metrics are proposed in this chapter. In the beginning, Section 4.2 provides a general description of both the real-world and synthetic datasets. For the real-world datasets, twenty real-world public datasets are included with different numbers of classes, features, and samples. The Madelon dataset is introduced and discussed as the synthetic dataset. Then, Section 4.3 reviews the performance evaluation methods on predictive and stability measures. Two new evaluation metrics are proposed in Section 4.4, including weighted accuracy and robustness. Afterwards, their performance has been thoroughly investigated and compared in Section 4.5. In order to comprehensively measure the performance of different FS algorithms, a multi-criteria evaluation method is also introduced based on radar charts in Section 4.6. The run time of all the proposed metrics is also provided. In general, the proposed evaluation methods provide better evaluation capabilities in differentiating different FS methods than the existing methods in the literature. Chapters 5 and 6 will apply the proposed methods for method evaluation and comparison.

# Chapter 5

# Ensemble Learning Framework for Aggregating Different FS Methods

## 5.1 Introduction

Ensemble learning is a general framework in the machine learning area to seek better predictive performance. Recently, it has become a prolific field of machine learning based on the assumption that the combination of multiple models' outputs is better than a single one. The rationale of ensemble learning is to build a set of hypotheses using various methods and combine them to produce better results afterwards [25]. Generally, the base learners in the ensemble framework should be as accurate and diverse as possible to achieve a good performance [239].

A typical approach in an ensemble method consists of two steps. The first step is to apply several base learners sequentially or in parallel. Then the base learners are combined, where the majority voting and weighted averaging are the most popular combination schemes for the cases of classification and regression, respectively [239]. The use of ensemble learning proves its effectiveness over recent years. In the literature, there are different approaches, such as boosting and bagging. The boosting approach utilizes a set of base learners to

improve the model's performance. The central idea behind boosting is the application of homogeneous algorithms sequentially [59]. On the other hand, bagging aims to improve the predictive performance by applying the base algorithms parallelly. The critical point of bagging is to use multiple base learners trained separately with random samples in the training set [23].

This chapter investigates and explores the use of ensemble learning to improve the performance of FS methods. By combining the output of several feature selectors, the ensemble FS method can provide a solution to better understand the feature importance, which represents the inherent characteristics of the data well and achieves comprehensively good performance. Furthermore, the users can also be released from the pressure of choosing a single method.

## 5.2 Methodology

Unlike the homogeneous or heterogeneous approaches (explained in Section 2.3.2) in the bagging process, this chapter proposes a new framework that simultaneously utilizes both homogeneous and heterogeneous approaches. The proposed method aims to take the advantages of the data's information by data variation and the algorithm's knowledge by function variation. The performance of different combination methods is also evaluated and compared. In general, three main steps are introduced in the framework, which include: (1) distribution generation of feature importance; (2) distribution ensemble using aggregation methods; (3) defuzzification for feature ranking. Several different approaches are also included within the main steps, as shown in Fig. 5.1.

As illustrated in Fig. 5.1, the first step of the proposed method generates the distribution of the feature importance from different FS algorithms using three approaches: score-based, rank-based, and fuzzy-based. Then, those distributions by different FS methods are combined and aggregated for each feature, respectively. The weighted combination and

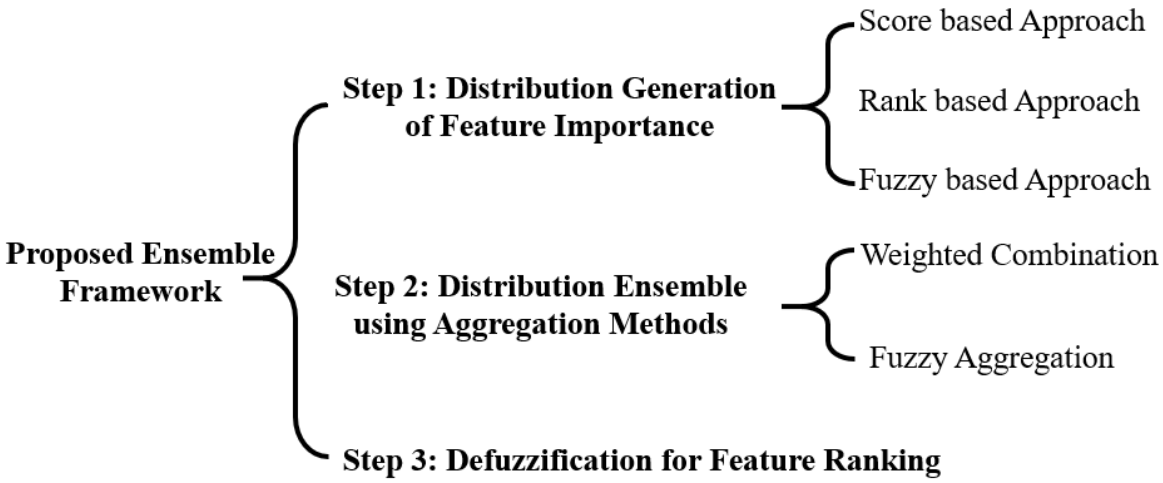



Fig. 5.1 Overview of the proposed ensemble framework

fuzzy aggregation methods are utilized during the process. In the last, after aggregating the distributions from different FS methods in step 2, the third step calculates each feature's final importance score using the defuzzification approach. At last, the corresponding feature ranking sequence can also be produced afterwards.

To be specific, for a given dataset with $N$ features, the proposed method aims to rank the features from the most to the least significant by combining $M$ different filter-based FS methods. The feature ranking result is then used for decision-making tasks (e.g., disease discrimination). The overall procedure of different steps and their relationships are shown in Fig. 5.2, with the detailed instructions of each step described as follows.

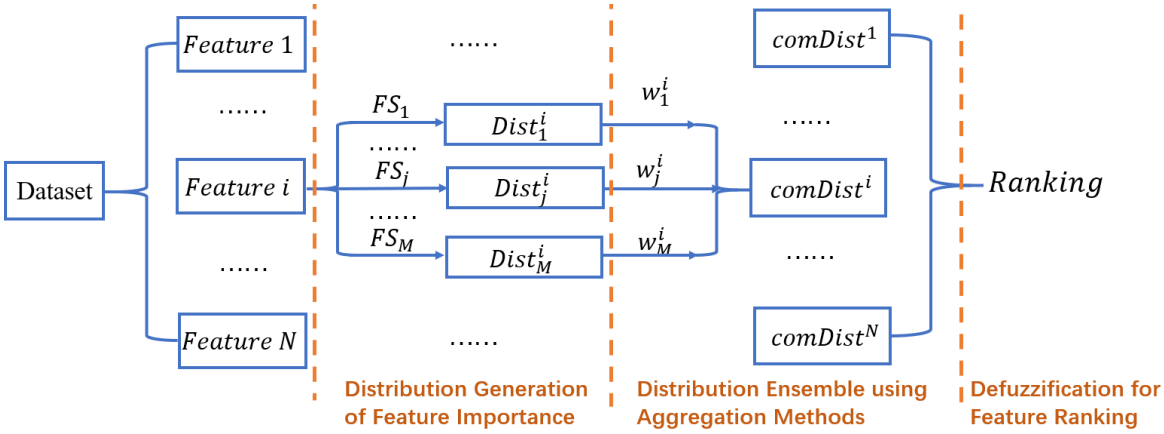



Fig. 5.2 Schematic diagram of the proposed ensemble method

### 5.2.1 Distribution Generation of Feature Importance

The bootstrap aggregation process is firstly applied to construct a number (denoted as $L$) of data subsets to increase FS methods' generalizability. $M$ different FS methods (denoted as $FS_j, j \in [1,...,M]$) are then applied on these subsets to generate the feature scores for each subset per method. Distributions are constructed using $L$ bootstrap subsets to represent the importance of each feature. Three approaches are proposed to generate the distributions, respectively: score-based, rank-based, and fuzzy-based. An overview of the distribution generation of feature importance is illustrated in Fig. 5.3. The detailed procedures and demonstration examples are shown as follows.

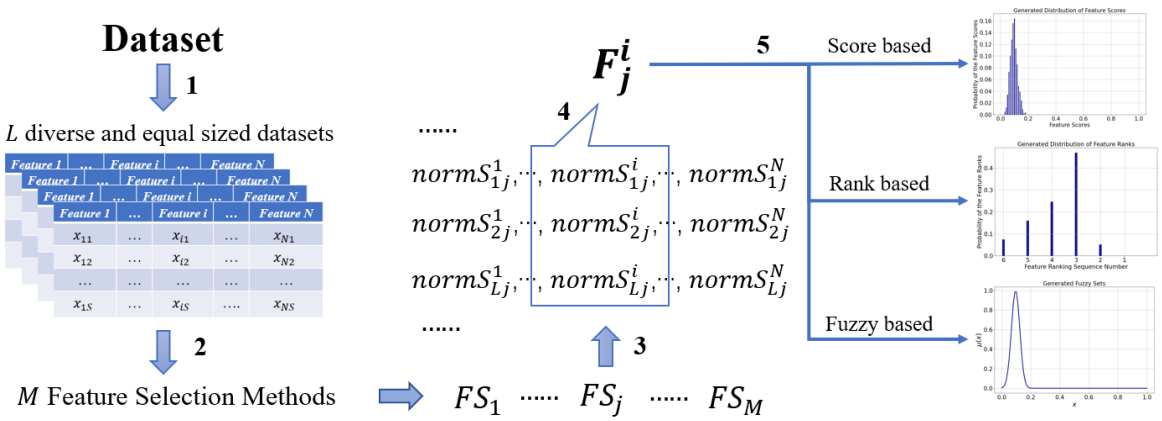



Fig. 5.3 Illustration of the distribution generation of feature importance process

**1. Bootstrap Aggregating Process** Randomly select the samples and generate $L$ diverse and equal-sized data subsets with replacement (denoted as *Subset* $l$, $(l \in [1,...,L])$).

**2. FS Method Deployment** Apply FS method $FS_j$ on *Subset* $l$, and calculate the feature score (denoted as $S^i_{j,l}$) of the $i$th feature $(i \in [1,...,N])$. For FS method $FS_j$, the calculated feature scores of different features and subsets are represented in Table 5.1.

**3. Feature Score Normalization** Normalize the feature scores of each FS method into the range of [0, 1] using min-max normalization [87], as expressed in Equation 5.1.

Table 5.1 Generated feature scores by the FS method $FS_j$

| **Feature Scores** | **1** | **2** | **...** | **i** | **...** | **N** |
|---|---|---|---|---|---|---|
| 1 | $S^1_{j,1}$ | $S^2_{j,1}$ | ... | $S^i_{j,1}$ | ... | $S^N_{j,1}$ |
| 2 | $S^1_{j,2}$ | $S^2_{j,2}$ | ... | $S^i_{j,2}$ | ... | $S^N_{j,2}$ |
| ... | ... | ... | ... | ... | ... | ... |
| $l$ | $S^1_{j,l}$ | $S^2_{j,l}$ | ... | $S^i_{j,l}$ | ... | $S^N_{j,l}$ |
| ... | ... | ... | ... | ... | ... | ... |
| L-1 | $S^1_{j,L-1}$ | $S^2_{j,L-1}$ | ... | $S^i_{j,L-1}$ | ... | $S^N_{j,L-1}$ |
| L | $S^1_{j,L}$ | $S^2_{j,L}$ | ... | $S^i_{j,L}$ | ... | $S^N_{j,L}$ |

$$normS^i_{j,l} = \frac{S^i_{j,l} - min\{S_j\}}{max\{S_j\} - min\{S_j\}} \tag{5.1}$$

where $min\{S_j\}$ and $max\{S_j\}$ represent the minimum and maximum values of the calculated feature scores using the $FS_j$ method on all features.

**4. Feature Discretization** After the FS method deployment and normalization process, the feature scores by different FS methods become non-uniformity with various data density and value ranges, which prevents them to be aggregated fairly. Thus, a feature discretization process is introduced to ensure that different feature distributions are in a similar value range and comparable. In this section, each feature space is divided into 101 equal-sized interval scores (set as $U$).

$$U = \{0, 0.01, 0.02, ..., 0.98, 0.99, 1\} \tag{5.2}$$

The previously normalized feature scores are discrete and mapped into the element of set $U$ (denoted as $dS^i_{j,l}$).

**5. Feature Importance Representation** This section explores the suitable solutions to represent the feature importance. Thus, the distributions or fuzzy sets which describe the

features' importance are constructed. Three approaches are introduced based on the generated distributions' morphology, e.g., score, rank, and fuzzy-based approaches.

**5.1 Score based approach** Feature scores are produced by many FS methods to represent certain features' importance. They are usually directly correlated with the features' importance. Higher feature score values typically indicate more essential features. In practice, it may vary significantly when the features are assigned to the particular feature scores across different FS methods. Hence, a score-based approach is proposed to describe and model the features' importance and variations. By assembling the feature scores for each feature, the probability density function (PDF) of the score-based method is constructed. The PDF provides a comprehensive and detailed description of the distribution of the feature score values for each feature and each FS method. It has also been widely applied in many machine learning areas and therefore been adopted here.

1. Given the FS method $FS_j$ and the $i$th feature, combine the normalized feature scores from $L$ different data subsets into a list $Fscore_j^i$.

$$Fscore_j^i = \{dS_{j,l}^i | l \in [1, ..., L]\} \tag{5.3}$$

2. Represent the distribution $Dist_j^i$ based on $Fscore_j^i$ in Equation 5.4.

$$Dist_j^i = \{(x, PDF_j^i(x)) | x \in U\} \tag{5.4}$$

where the probability density function $PDF_j^i(x)$ is calculated in Equation 5.5. $\forall x \in U$, $freq(x, Fscore_j^i)$ represents the number of $x$'s occurrence in $Fscore_j^i$.

$$PDF_j^i(x) = \begin{cases} freq(x, Fscore_j^i)/L & x \in Fscore_j^i \\ 0 & x \notin Fscore_j^i \end{cases} \tag{5.5}$$

The distribution $Dist_j^i(x)$ represents the feature scores and indicates the corresponding feature importance.

**5.2 Rank based approach** In practice, the feature score distributions vary a lot across different FS methods. When combining the results from various FS algorithms, it is assumed that the probability of scoring all feature values is evenly distributed for all FS methods. Otherwise, some biases could be introduced when combining the feature scores from different FS methods. The rank-based approach is proposed as the rank indices are consistent across different FS methods, which could minimize the bias in the aggregation process. The detailed procedures of this approach are described as below.

1. Based on the calculated feature scores, rank the features and produce the ranking index (denoted as $R_{j,l}^i$) of the $i$th feature on FS method $FS_j$ and data *Subset* $l$.
2. For the FS method $FS_j$ and the $i$th feature, combine the feature ranking indices from $L$ subsets into a list $Frank_j^i$.

$$Frank_j^i = \{R_{j,l}^i | l \in [1, ..., L]\} \tag{5.6}$$

3. Constitute the distribution $Dist_j^i$ based on $Frank_j^i$ using Equation 5.7.

$$Dist_j^i = \{(x, PDF_j^i(x)) | x \in U\} \tag{5.7}$$

where the probability density function $PDF_j^i(x)$ is calculated in Equation 5.8. $\forall x \in U$, $freq(x, Frank_j^i)$ represents the number of $x$'s occurrence in $Frank_j^i$.

$$PDF_j^i(x) = \begin{cases} freq(x, Frank_j^i)/L & x \in Frank_j^i \\ 0 & x \notin Frank_j^i \end{cases} \tag{5.8}$$

**5.3 Fuzzy based approach** The score-based and rank-based approaches face some limitations and disadvantages in practice. Firstly, the calculation process of the different distributions generated by the score-based approach is time-consuming. It takes time and storage to save and combine those different PDF distributions. Secondly, there are limited numbers of aggregation methods based on these distributions. In this case, a fuzzy-based approach is proposed here. The fuzzy-based method could be exceptionally efficient and space-saving by utilizing the Gaussian distributions for representing the feature importance. Besides, many aggregation methods from the literature can be applied in the aggregation process.

The FS calculation of different subsets is regarded as independent and identically distributed (i.i.d) experiments on account of the bootstrap process. Based on Bernoulli's law of large numbers [64], for any positive number $\varepsilon$, we have

$$\lim_{L\to\infty} P\{|\frac{freq(x,F_j^i)}{L} - \mu| < \varepsilon\} = 1 \tag{5.9}$$

Equation 5.9 indicates that when $L$ becomes large enough, $freq(x,F_j^i)/L$ can be used to represent the membership function values $\mu_j^i(x)$, as shown in Equation 5.10.

$$\mu_j^i(x) = \begin{cases} \lim\limits_{L\to\infty} \frac{freq(x,F_j^i)}{L} & x \in F_j^i \\ 0 & x \notin F_j^i \end{cases} \tag{5.10}$$

Membership function $\mu_j^i(x)$ represents the feature scores that indicate the corresponding feature's importance. The feature scores with higher membership values contribute more for the features' importance calculation. A fuzzy-based approach is proposed to represent the features' importance in this situation. The detailed instructions are described as below.

1. For FS method $FS_j$ and the $i$th feature, calculate the mean value $mean_j^i$, standard deviation $\delta_j^i$ and height $height_j^i$ from the probability density function $PDF_j^i(x)$.

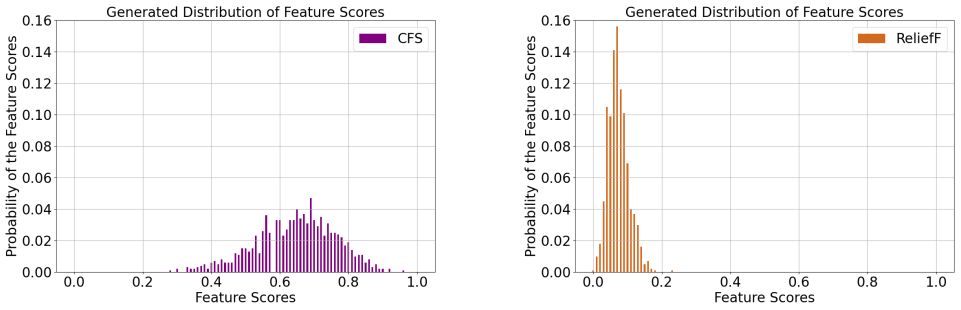
2. Constitute a normalized type-1 fuzzy set to represent the distribution of importance on the $i$th feature and FS method $FS_j$, as expressed in Equation 5.11.

$$Dist_j^i = \{(x, \mu_j^i(x)) | x \in X\} \tag{5.11}$$

Based on Bernoulli's law of large numbers, distributions of membership function $\mu_j^i(x)$ are depicted as Gaussian shaped. The membership function $\mu_j^i$ can be constructed using Equation 5.12.

$$\mu_j^i(x) = \frac{1}{\sqrt{2\pi}\delta_j^i} exp(-\frac{(x - mean_j^i)^2}{2(\delta_j^i)^2}) \tag{5.12}$$

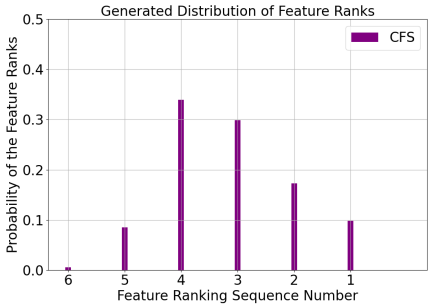
**6. Illustration of the Distribution Generation Process** Two state-of-the-art FS methods are implemented on feature 1 of Vertebral dataset [46], which are correlation-based FS (CFS) and ReliefF. In the experiment, $L$ is set as 1000. The generated distributions of CFS and ReliefF using score, rank and fuzzy-based approaches are shown in Fig. 5.4, 5.5 and 5.6, respectively.



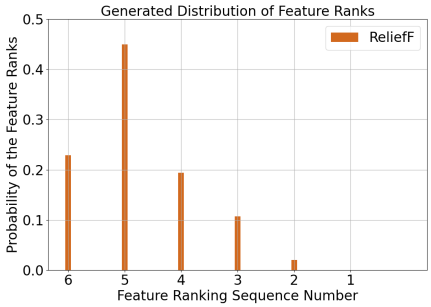

(a) CFS (score based) (b) ReliefF (score based)

Fig. 5.4 Generated distributions using the score based approach

From Fig. 5.4, 5.5 and 5.6, it can be seen that the distributions of different FS methods vary significantly on the morphology and some general characteristics such as mean, height, and width. Hence, it becomes clear that different FS methods score features differently. On the one hand, it may be on account that those FS methods give different weights to the

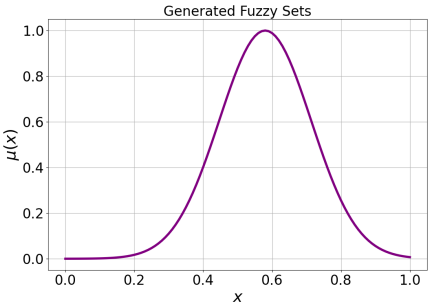



(c) CFS (rank based)

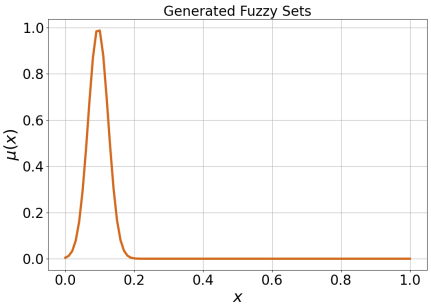



(d) ReliefF (rank based)

Fig. 5.5 Generated distributions using the rank based approach




(e) CFS (fuzzy based)




(f) ReliefF (fuzzy based)

Fig. 5.6 Generated distributions using the fuzzy based approach

features. On the other hand, it may also be due to that those methods consider the different properties in terms of what makes a feature important. Therefore, it can be observed that the generated distributions help in providing insights into the features' importance from the perspectives of different FS algorithms.

In addition, the later ensemble and defuzzification process can also help form an overall idea of the features' importance from a view of multiple FS algorithms. By effectively weighting, aggregating and incorporating the feature importance from different FS methods, the final generated distributions aim to provide a more comprehensive way to illustrate the features' importance.

### 5.2.2 Distribution Ensemble using Aggregation Methods

The distribution results from different FS methods are aimed to be aggregated together to construct a final feature distribution. The process of this step is illustrated in Fig. 5.7. Based on the employed techniques, the aggregation approaches include weighted combination and fuzzy aggregation methods.

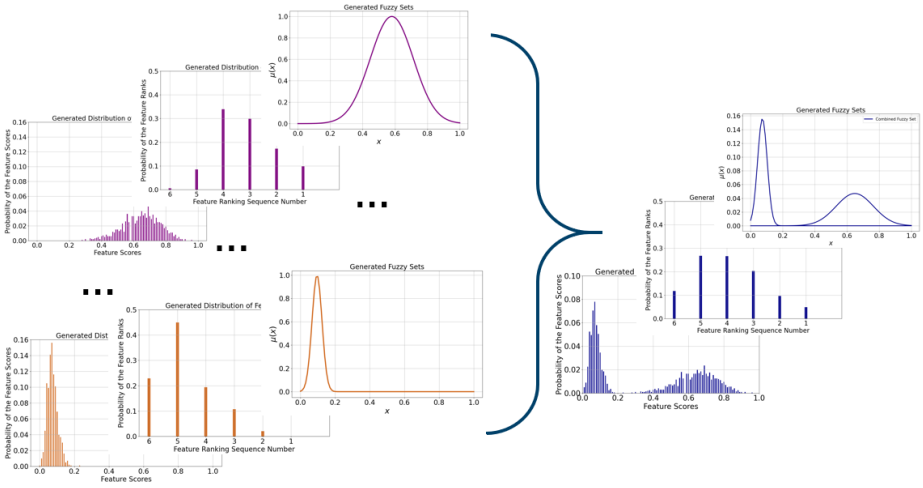



Fig. 5.7 Illustration of the distribution ensemble using different aggregation methods

**Approach 1: Weighted Combination Methods** For the $i$th feature, the distributions from $M$ different FS methods are aggregated using the weighted sum shown in Equation 5.13.

$$\begin{aligned} comDist^i &= \{(x, Dist^i_{com}(x)) | x \in U\} \\ \text{where } \ Dist^i_{com}(x) &= \sum_{j=1}^{M} w^i_j \times Dist^i_j(x) \\ \text{subject to } \sum_{j=1}^{M} w^i_j &= 1 \end{aligned} \tag{5.13}$$

Various combination methods are proposed under this aggregation framework. At first, the distribution index is calculated to quantify the inherent characteristics of FS methods. Several weights calculation functions are introduced below.

**1. Distribution Index Calculation** Distribution indices act as the independent metric to represent the dispersity of the generated distributions of score-based and rank-based approaches. Various metrics are applied to measure the distribution index using different FS methods, shown as follows.

- *Statistical Index*: The statistical index is used to measure the variations and changes of the individual data points [7]. In the statistical area, standard deviation (STD) is widely used to quantify the variation or dispersion of data. Given the hypothesis that the standard deviation of $F_j^i$ (denoted as $SD(F_j^i)$) is proportional to the uncertainty of the FS method, the distributions with higher STD are supposed to hold higher uncertainty, hence contribute less for the combined results [79].
- *Fuzzy Entropy*: Different fuzzy entropy functions introduced in Section 2.2.2 are used to calculate the entropy values of the distributions, such as weighted measures of fuzzy entropy (Parkash's Method) [155], etc. The same as the standard deviation, fuzzy entropy values of distributions are also proportional to FS methods' uncertainty. The FS methods with high uncertainty and higher fuzzy entropy values will contribute less to the combined results. In this research, weighted measures of fuzzy entropy (Parkash's method) are implemented, as shown below.

$$H(x) = \sum_x [sin\frac{\pi\mu(x)}{2} + sin\frac{\pi(1-\mu(x))}{2} - 1] \tag{5.14}$$

where $\mu(x)$ represents the membership function value of $x$, and $H(x)$ indicates the fuzzy entropy values.

**2. Weights Calculation Functions** After producing the distribution index, the weights calculation functions are used to calculate the corresponding weights in Equation 5.13.

- *Reciprocal Weights (RW)*: By using reciprocal approach, the reciprocal weights calculation function is defined in Equation 5.15.

$$w_j^i = \frac{1/Index(F_j^i)}{\sum_{j=1}^{M}[1/Index(F_j^i)]} \tag{5.15}$$

where $Index(F_j^i)$ indicates the distribution index of the feature $F_j^i$.

- *One Minus Weights (OW)*: Another weights calculation function using normalized distribution indices is shown as below [79].

$$w_j^i = \frac{1 - Index(F_j^i)}{\sum_{j=1}^{M}[1 - Index(F_j^i)]} \tag{5.16}$$

**Approach 2: Fuzzy Aggregation Methods** Fuzzy aggregation acts as the linear extensions of Boolean connectives in the scale between 0 and 1 [142]. Different fuzzy aggregation methods combine the generated fuzzy sets using the fuzzy-based approach. As introduced in Section 2.2.1, various fuzzy operators are proposed in the literature, such as S-norm and T-norms. T-norms operators typically produce the intersection among fuzzy sets and eliminate other information outside the region, leading to the loss of information in the aggregation process. On the other hand, S-norms are the generalized form of fuzzy union, which integrates different fuzzy sets. Most of the information has been retained using S-norms, which are chosen in this research. As introduced in Section 2.2.1, there are different kinds of S-norms, such as Dombi class, Yager class, etc. A parameterized family of S-norms Yager class is chosen on account that it can cover different situations in the value range (as shown in Fig. 2.2).

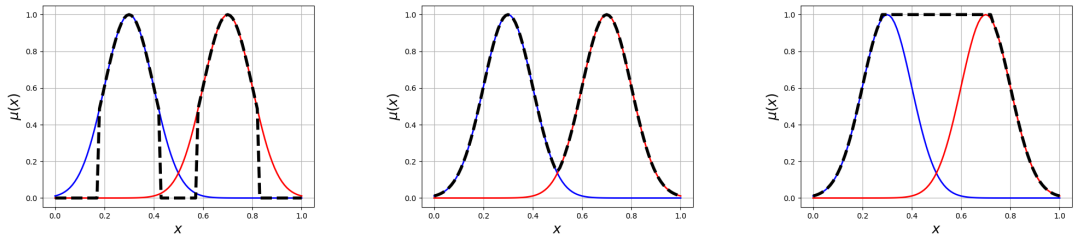
**1. Yager Class** In consideration of the computability and practicability, Yager class [223] is chosen for research with different parameter values. Given that $a$ and $b$ represent the membership function values of different fuzzy sets, the Yager class is represented below, where $\omega$ stands for the function parameter.

$$s_\omega(a,b) = min[1,(a^\omega + b^\omega)^{1/\omega}], \quad \omega \in (0,\infty) \tag{5.17}$$

**2. Drastic Sum** In the extreme case, when $\omega$ becomes 0, Yager class turns to be drastic sum S-norm $S_{ds}(a,b)$, as shown in Equation 2.19.

$$s_{ds}(a,b) = \begin{cases} a & if\ b = 0 \\ b & if\ a = 0 \\ 1 & otherwise \end{cases} \tag{5.18}$$

The combined fuzzy sets after the fuzzy aggregation process using the Yager class with different parameters ($\omega = 1000$, 10) and drastic sum S-norms are visualized in Fig. 5.8. Red and blue lines show the original fuzzy sets. The black dashed line indicates the fuzzy sets after aggregation.




(a) Yager class with $\omega = 1000$ (b) Yager class with $\omega = 10$ (c) Drastic sum

Fig. 5.8 Demonstration of the fuzzy aggregation process

### 5.2.3 Defuzzification for Feature Ranking

**1. Defuzzification** After combining different fuzzy sets, a single value is computed using defuzzification processes. Based on Section 2.2.4, centroid defuzzifier, or named center of gravity defuzzifier is applied in the process, as shown below.

$$y^* = \frac{\int_{\mathbb{V}} y\mu_{B'}(y)dy}{\int_{\mathbb{V}} \mu_{B'}(y)dy} \tag{5.19}$$

where defuzzification maps a fuzzy set $B' \in \mathbb{V} \subset \mathbb{R}^N$ to the crisp point $y^* \in \mathbb{V}$.

**2. Feature Ranking** The feature ranking sequence is obtained from the highest to the lowest value based on the final defuzzified feature scores for the score-based and fuzzy-based approaches. The feature ranking sequence is easily produced for the rank-based approaches by sorting the combined ranks from the smallest to the largest ranking numbers. Afterwards, the feature ranking sequence is utilized as guidance for the subsequent decision-making process.

## 5.3 Experiments

To further investigate the performance of different approaches in the proposed FS framework, various datasets and evaluation metrics are applied for performance comparison and analysis.

### 5.3.1 Data Repository

The data repositories for training and testing purposes are constructed from public datasets.

**1. Training data repository for methods tuning** Ten datasets from the UCI machine learning repository with a different number of classes, features and samples are chosen to tune the optimal combination or parameter in the proposed framework. Some general information about the datasets is shown in Table 5.2.

Table 5.2 Description of training data repository for methods tuning

| **No.** | **Datasets** | *#C* | *#F* | *#S* | *#S Distribution over #C* |
|---|---|---|---|---|---|
| 1 | Mammographic | 2 | 5 | 830 | 403 / 427 |
| 2 | PIMA | 2 | 8 | 768 | 268 / 500 |
| 3 | Statlog Heart | 2 | 13 | 270 | 120 / 150 |
| 4 | WDBC | 2 | 30 | 569 | 212 / 357 |
| 5 | Sports Articles | 2 | 59 | 1000 | 365 / 635 |
| 6 | Parkinsons | 2 | 22 | 195 | 48 / 147 |
| 7 | Dermatology | 6 | 34 | 358 | 20/48/48/60/71/111 |
| 8 | Sonar | 2 | 60 | 208 | 97/111 |
| 9 | Musk | 2 | 166 | 476 | 207/269 |
| 10 | Colon Cancer | 2 | 2000 | 62 | 22 / 40 |

where #C, #F and #S represent the number of classes, features, and samples, respectively. The samples' distribution over different classes is also included (*#S Distribution over #C*).

**2. Testing data repository for performance testing** Another ten datasets with a various number of classes, features and samples are chosen for independent evaluation, as shown in Table 5.3. Those datasets are aimed for final performance analysis and comparison.

Table 5.3 Description of testing data repository for performance testing

| No. | Datasets | *#C* | *#F* | *#S* | *#S Distribution over #C* |
|---|---|---|---|---|---|
| 11 | Banknote | 2 | 4 | 1372 | 610 / 762 |
| 12 | WBC | 2 | 9 | 682 | 239 / 443 |
| 13 | CMSC | 2 | 18 | 540 | 46 / 494 |
| 14 | Appendicitis | 2 | 7 | 106 | 21 / 85 |
| 15 | BCC | 2 | 9 | 116 | 52 / 64 |
| 16 | Wine | 3 | 13 | 178 | 48 / 59 / 71 |
| 17 | Glass | 6 | 9 | 214 | 9 / 13 / 17 / 29 / 70 / 76 |
| 18 | Spectfheart | 2 | 44 | 267 | 55 / 212 |
| 19 | Breast Tissue | 6 | 9 | 106 | 14 / 15 / 16 / 18 / 21 / 22 |
| 20 | Lung | 5 | 3312 | 203 | 6 / 17 / 20 / 21 / 139 |

### 5.3.2 Selection of FS Methods

Different FS methods are introduced for performance analysis and comparison, including the base selectors, proposed ensemble FS methods, and other state-of-the-art FS methods.

**1. Base Selectors** Four widely used algorithms from different filter FS categories are implemented as the base selectors. The general description of the base selectors is shown in Table 5.4.

Table 5.4 General description of the base selectors

| No. | Alias | Name | Category | Supervision |
|---|---|---|---|---|
| 1 | CFS | Correlation based FS [73] | Statistical-based FS | Supervised |
| 2 | ReliefF | ReliefF FS [171] | Similarity-based FS | Supervised |
| 3 | MIFS | Mutual Information based FS [231] | Information-based FS | Supervised |
| 4 | IFS | Infinite based FS [175] | Graph-based FS | Unsupervised |

**2. The Proposed Ensemble Methods** By incorporating various distributions of feature importance using the aggregation methods in Section 5.2, different proposed FS methods are introduced as follows. The filter-based FS methods in Table 5.4 were chosen as the base selectors. The number of bootstrap subsets $L$ was set as 1000. The bootstrap process indicates that 63.2% of data samples are implemented with replacement each time.

• **Score based Approach** By incorporating the different distribution indexes and weighting schemes, the proposed ensemble FS framework using the score-based approach is shown in Table 5.5. In the following research, the ensemble methods are represented using the column "Abbr.".

Table 5.5 Proposed ensemble FS methods using the score based approach

| No. | Abbr. | Distribution Generation | Distribution Index | Weighting Scheme |
|---|---|---|---|---|
| 1 | $S\_S_{RW}$ | Score based | Standard Deviation | RW |
| 2 | $S\_S_{OW}$ | Score based | Standard Deviation | OW |
| 3 | $S\_P_{RW}$ | Score based | Parkash's Entropy | RW |
| 4 | $S\_P_{OW}$ | Score based | Parkash's Entropy | OW |

• **Rank based Approach** The proposed ensemble FS framework using the rank-based approach is shown below by employing different distribution indexes and weighting schemes.

Table 5.6 Proposed ensemble FS methods using the rank based approach

| No. | Abbr. | Distribution Generation | Distribution Index | Weighting Scheme |
|---|---|---|---|---|
| 1 | $R\_S_{RW}$ | Rank based | Standard Deviation | RW |
| 2 | $R\_S_{OW}$ | Rank based | Standard Deviation | OW |
| 3 | $R\_P_{RW}$ | Rank based | Parkash's Entropy | RW |
| 4 | $R\_P_{OW}$ | Rank based | Parkash's Entropy | OW |

• **Fuzzy based Approach** By incorporating various fuzzy aggregation techniques such as the Yager class and drastic sum S-norms, the proposed ensemble methods using the fuzzy-based approach are shown in Table 5.7.

Table 5.7 Proposed ensemble FS methods using the fuzzy based approach

| No. | Abbr. | Aggregation Approach | Parameter Setting |
|---|---|---|---|
| 1 | *F_Yager* | S-norm (Yager Class) | 1000, 10, 0.1, 0.001 |
| 2 | *F_DS* | S-norm (Drastic Sum) | None |

**3. The FS Methods for Performance Comparison** Other state-of-the-art FS methods with different categories and supervision statuses are used for performance analysis and comparison. The detailed information on the FS methods is shown in Table 5.8.

Table 5.8 Description of the FS methods for performance comparison

| No. | Alias | Name | Category | Supervision |
|---|---|---|---|---|
| 1 | Lap Score | Laplacian Score FS [81] | Similarity-based FS | Unsupervised |
| 2 | SPEC | SPEC [235] | Similarity-based FS | Unsupervised |
| 3 | Gini Index | Gini Index based FS [61] | Statistical-based FS | Supervised |
| 4 | F Score | F Score based FS [164] | Statistical-based FS | Supervised |

### 5.3.3 Methods Tuning on the Training Data Repository

The proposed methods were first applied to the data repository for methods tuning to investigate the proposed method's performance using different combinations or parameters. Ten-fold cross-validation was used in the process. The random forest classifier was selected for the decision-making process based on the sorted features because of its bagging character and robustness to outliers and nonlinear data. Spearman's rank correlation coefficient was utilized in the stability and robustness evaluation metrics. Hence, FS methods' performance is reported using the different evaluation methods in Chapter 4: accuracy, robustness, and multi-criteria performance. The methods tuning process was implemented separately using three distribution generation approaches: score-based, rank-based, and fuzzy-based.

**1. Evaluation Criteria among Different Datasets** As for any FS method, its performance varies typically on different datasets. To better evaluate the algorithms' overall perfor-

mance on different datasets, several evaluation methods were proposed and utilized in the experiments that follow.

- *AVG*: Average performance values in the data repository;
- *Ranking Index*: It indicates the ranking index among different FS methods. Lower values mean higher ranks and better performance. When more than one method produces the same results, "joint rank" is used, which measures the average value of the ranking order. For instance, ranking index 1 indicates the FS method produced the best performance. When two methods produced the same best performance, both of them were scored 1.5, therefore.
- *ARI*: Average ranking index indicates the mean ranking index in the data repository.

**2. Methods Tuning of the Score based Approach** From Table 5.5, four ensemble FS methods with the score-based approach were implemented under different distribution indexes and weighting schemes. This section compares the performance using three evaluation metrics in Chapter 4: accuracy, robustness, and multi-criteria. The numbers in bold represent the optimal performance values for the given datasets. The numbers in the brackets indicate the method's ranking index in that dataset.

***I*. Accuracy** The performance comparison on accuracy is shown in Table 5.9.

Table 5.9 Methods tuning of the score based approach on accuracy

| No. | Datasets | Ensemble Methods using Score based Approach | | | |
|---|---|---|---|---|---|
| | | $S\_S_{RW}$ | $S\_S_{OW}$ | $S\_P_{RW}$ | $S\_P_{OW}$ |
| 1 | Mammographic | .766 (4) | **.767** (2) | **.767** (2) | **.767** (2) |
| 2 | PIMA | .716 (3) | .715 (4) | .718 (2) | **.719** (1) |
| 3 | StatlogHeart | .714 (4) | **.735** (1) | .730 (2) | .727 (3) |
| 4 | WDBC | .922 (3) | **.926** (1) | .923 (2) | .904 (4) |
| 5 | Sports Articles | .762 (4) | **.767** (1) | .765 (2) | .763 (3) |
| 6 | Parkinsons | **.793** (1) | .789 (3) | .790 (2) | .699 (4) |
| 7 | Dermatology | .683 (4) | **.719** (1) | .687 (3) | .718 (2) |
| 8 | Sonar | .479 (4) | .520 (2) | **.526** (1) | .501 (3) |
| 9 | Musk | **.495** (1) | .481 (3) | .483 (2) | .478 (4) |
| 10 | Colon Cancer | .830 (3) | **.831** (1.5) | **.831** (1.5) | .760 (4) |
| AVG (ARI) | | .716 (3.1) | **.725** (1.95) | .722 (1.95) | .703 (3) |

***II*. Robustness** The performance comparison on robustness is shown in Table 5.10.

Table 5.10 Methods tuning of the score based approach on robustness

| No. | Datasets | Ensemble Methods using Score based Approach | | | |
|---|---|---|---|---|---|
| | | $S\_S_{RW}$ | $S\_S_{OW}$ | $S\_P_{RW}$ | $S\_P_{OW}$ |
| 1 | Mammographic | .908 (4) | .950 (2) | **.954** (1) | .947 (3) |
| 2 | PIMA | .936 (3) | **.971** (1) | .898 (4) | .968 (2) |
| 3 | StatlogHeart | **.951** (1) | .876 (3) | .888 (2) | .534 (4) |
| 4 | WDBC | .884 (3) | .892 (2) | **.929** (1) | .342 (4) |
| 5 | Sports Articles | .865 (2) | **.911** (1) | .848 (3) | .660 (4) |
| 6 | Parkinsons | **.848** (1) | .813 (2) | .787 (4) | .804 (3) |
| 7 | Dermatology | .624 (2) | **.867** (1) | .553 (4) | .559 (3) |
| 8 | Sonar | **.949** (1) | .929 (3) | .931 (2) | .685 (4) |
| 9 | Musk | **.866** (1) | .797 (2) | .796 (3) | .757 (4) |
| 10 | Colon Cancer | .337 (3) | **.585** (1) | .356 (2) | .267 (4) |
| AVG (ARI) | | .817 (2.1) | **.859** (1.8) | .794 (2.6) | .652 (3.5) |

In Table 5.9, $S\_S_{OW}$ method produces the overall best performance on accuracy in the data repository for method tuning. Besides, for the aspects of robustness in Table 5.10, the $S\_S_{OW}$ method also produces the relatively optimal performance, which outperformed the other competitors on both AVG and ARI. Comparatively, the $S\_P_{RW}$ method ranks second place on the accuracy, while the $S\_S_{RW}$ method ranks second on robustness.

***III*. Multi-Criteria** Based on the evaluation metrics in Section 4.6, the methods were tuned comprehensively using multi-criteria evaluation metric, as shown in Table 5.11.

Table 5.11 Methods tuning of the score based approach on multi-criteria

| No. | Datasets | Ensemble Methods using Score based Approach | | | |
|---|---|---|---|---|---|
| | | $S\_S_{RW}$ | $S\_S_{OW}$ | $S\_P_{RW}$ | $S\_P_{OW}$ |
| 1 | Mammographic | .764 (4) | .803 (2) | **.806** (1) | .802 (3) |
| 2 | PIMA | .768 (3) | .783 (2) | .749 (4) | **.789** (1) |
| 3 | StatlogHeart | **.771** (1) | .732 (3) | .745 (2) | .432 (4) |
| 4 | WDBC | .845 (3) | .874 (2) | **.897** (1) | .424 (4) |
| 5 | Sports Articles | .731 (2) | **.782** (1) | .703 (3) | .570 (4) |
| 6 | Parkinsons | **.747** (1) | .714 (2) | .675 (3) | .661 (4) |
| 7 | Dermatology | .477 (2) | **.731** (1) | .408 (4) | .440 (3) |
| 8 | Sonar | .615 (3) | .632 (2) | **.636** (1) | .422 (4) |
| 9 | Musk | **.577** (1) | .519 (3) | .520 (2) | .493 (4) |
| 10 | Colon Cancer | .458 (2) | **.587** (1) | .439 (3) | .325 (4) |
| AVG(ARI) | | .675 (2.2) | **.716** (1.9) | .658 (2.4) | .536 (3.5) |

From Table 5.11, $S\_S_{OW}$ method produces the optimal comprehensive performance among the different competitors. It indicates a better performance to utilize the standard

deviation as the distribution index and OW as the weighting scheme than the other competitors. Therefore, this research chooses the $S\_S_{OW}$ method for further performance analysis.

**3. Methods Tuning of the Rank based Approach** From Table 5.6, four ensemble FS methods of the rank-based approach were constructed. Based on the evaluation methods in Chapter 4, this section evaluated and compared the performance on different aspects, e.g., accuracy, robustness and multi-criteria.

***I*. Accuracy** The performance comparison on accuracy is shown in Table 5.12.

Table 5.12 Methods tuning of the rank based approach on accuracy

| No. | Datasets | Ensemble Methods using Rank based Approach | | | |
|---|---|---|---|---|---|
| | | $R\_S_{RW}$ | $R\_S_{OW}$ | $R\_P_{RW}$ | $R\_P_{OW}$ |
| 1 | Mammographic | .825 (4) | **.827** (2) | **.827** (2) | **.827** (2) |
| 2 | PIMA | .717 (2.5) | **.740** (1) | .717 (2.5) | .698 (4) |
| 3 | StatlogHeart | .728 (3) | **.733** (1) | .731 (2) | .712 (4) |
| 4 | WDBC | .921 (3) | **.923** (1) | .922 (2) | .910 (4) |
| 5 | Sports Articles | .765 (3) | **.767** (1.5) | **.767** (1.5) | .763 (4) |
| 6 | Parkinsons | .794 (2.5) | .794 (2.5) | **.795** (1) | .792 (4) |
| 7 | Dermatology | .722 (2) | **.725** (1) | .714 (3) | .707 (4) |
| 8 | Sonar | .498 (4) | .566 (2) | .564 (3) | **.575** (1) |
| 9 | Musk | **.497** (1) | .491 (2) | .482 (4) | .483 (3) |
| 10 | Colon Cancer | .792 (4) | .829 (2.5) | **.830** (1) | .829 (2.5) |
| AVG(ARI) | | .726 (2.9) | **.740** (1.65) | .735 (2.2) | .729 (3.25) |

***II*. Robustness** The performance comparison on robustness is shown in Table 5.13.

Table 5.13 Methods tuning of the rank based approach on robustness

| No. | Datasets | Ensemble Methods using Rank based Approach | | | |
|---|---|---|---|---|---|
| | | $R\_S_{RW}$ | $R\_S_{OW}$ | $R\_P_{RW}$ | $R\_P_{OW}$ |
| 1 | Mammographic | .720 (4) | **.876** (1) | .756 (2) | .725 (3) |
| 2 | PIMA | .885 (2) | **.974** (1) | .866 (3) | .436 (4) |
| 3 | StatlogHeart | **.920** (1) | .897 (3) | .906 (2) | .454 (4) |
| 4 | WDBC | .913 (2) | **.917** (1) | .886 (3) | .333 (4) |
| 5 | Sports Articles | .865 (2) | **.946** (1) | .861 (3) | .828 (4) |
| 6 | Parkinsons | .911 (2) | **.913** (1) | .905 (3) | .271 (4) |
| 7 | Dermatology | .829 (3) | .863 (2) | **.871** (1) | .661 (4) |
| 8 | Sonar | .824 (4) | **.866** (1) | .856 (3) | .861 (2) |
| 9 | Musk | **.873** (1) | .800 (2) | .794 (3) | .751 (4) |
| 10 | Colon Cancer | **.652** (1) | .619 (2) | .611 (3) | .558 (4) |
| AVG(ARI) | | .839 (2.2) | **.867** (1.5) | .831 (2.6) | .588 (3.7) |

For the aspects of accuracy and robustness in Table 5.12 and 5.13, $R\_S_{OW}$ method produces the overall best performance of both AVG and ARI values in the data repository for methods tuning. Comparatively, the $R\_P_{RW}$ method ranks the second place on the accuracy, while the $R\_S_{RW}$ method ranks the second on robustness.

*III*. **Multi-Criteria** Based on the multi-criteria evaluation method introduced in Section 4.6, the performance of different ensemble FS methods of the rank-based approach is compared and shown in Table 5.14.

Table 5.14 Methods tuning of the rank based approach on multi-criteria

| No. | Datasets | Ensemble Methods using Rank based Approach | | | |
|---|---|---|---|---|---|
| | | $R\_S_{RW}$ | $R\_S_{OW}$ | $R\_P_{RW}$ | $R\_P_{OW}$ |
| 1 | Mammographic | .685 (4) | **.798** (1) | .708 (2) | .690 (3) |
| 2 | PIMA | .743 (2) | **.809** (1) | .723 (3) | .281 (4) |
| 3 | StatlogHeart | **.767** (1) | .755 (3) | .756 (2) | .346 (4) |
| 4 | WDBC | .879 (2) | **.887** (1) | .853 (3) | .290 (4) |
| 5 | Sports Articles | .745 (2) | **.807** (1) | .744 (3) | .721 (4) |
| 6 | Parkinsons | **.803** (1) | .797 (2) | .793 (3) | .322 (4) |
| 7 | Dermatology | .698 (3) | **.731** (1) | .724 (2) | .559 (4) |
| 8 | Sonar | .564 (4) | **.617** (1) | .609 (2) | .607 (3) |
| 9 | Musk | **.578** (1) | .526 (2) | .518 (3) | .486 (4) |
| 10 | Colon Cancer | **.648** (1) | .637 (2) | .632 (3) | .598 (4) |
| AVG(ARI) | | .711 (2.1) | **.736** (1.5) | .706 (2.6) | .490 (3.8) |

In Table 5.14, $R\_S_{OW}$ method produces the overall best performance on both AVG and ARI values and the best performance on 6 out of 10 datasets. It indicates a reasonably good performance utilizing the standard deviation and OW weighting scheme in the weighted combination process. Therefore, the $R\_S_{OW}$ method is chosen as the representative of the rank-based approach for further performance analysis and comparison.

**4. Methods Tuning of the Fuzzy based Approach** The fuzzy-based approach was implemented with the fuzzy aggregation methods from Table 5.7 in the data repository for methods tuning. Based on the evaluation metrics from Chapter 4, the performance was compared on three aspects: accuracy, robustness and multi-criteria performance.

*I*. **Accuracy** The performance comparison on accuracy is shown in Table 5.15.

Table 5.15 Methods tuning of the fuzzy based approach on accuracy

| No. | Datasets | *F_Yager* with different $\omega$ | | | | *F_DS* |
|---|---|---|---|---|---|---|
| | | 1000 | 10 | 0.1 | 0.001 | |
| 1 | Mammographic | .769 (4) | .746 (5) | .772 (3) | **.827** (1.5) | **.827** (1.5) |
| 2 | PIMA | .717 (4) | .706 (5) | .724 (1) | **.728** (2.5) | **.728** (2.5) |
| 3 | StatlogHeart | .705 (5) | .743 (2) | **.757** (1) | .730 (3.5) | .730 (3.5) |
| 4 | WDBC | .926 (3) | .924 (5) | .925 (4) | **.928** (1.5) | **.928** (1.5) |
| 5 | Sports Articles | .735 (5) | **.767** (1.5) | **.767** (1.5) | .761 (3.5) | .761 (3.5) |
| 6 | Parkinsons | .790 (5) | **.811** (1) | .793 (4) | .798 (2.5) | .798 (2.5) |
| 7 | Dermatology | .667 (3) | **.684** (1.5) | **.684** (1.5) | .647 (4.5) | .647 (4.5) |
| 8 | Sonar | .452 (5) | **.490** (1) | .488 (2) | .468 (3.5) | .468 (3.5) |
| 9 | Musk | **.528** (2) | .486 (4) | .444 (5) | **.528** (2) | **.528** (2) |
| 10 | Colon Cancer | .713 (5) | .716 (4) | .724 (3) | **.767** (1.5) | **.767** (1.5) |
| AVG (ARI) | | .700 (4.1) | .716 (3) | .710 (2.6) | **.718** (2.65) | **.718** (2.65) |

***II*. Robustness** The performance comparison on robustness is shown in Table 5.16.

Table 5.16 Methods tuning of the fuzzy based approach on robustness

| No. | Datasets | *F_Yager* with different $\omega$ | | | | *F_DS* |
|---|---|---|---|---|---|---|
| | | 1000 | 10 | 0.1 | 0.001 | |
| 1 | Mammographic | .638 (5) | .721 (4) | .906 (3) | **1.00** (1.5) | **1.00** (1.5) |
| 2 | PIMA | .799 (5) | .901 (4) | .925 (3) | **1.00** (1.5) | **1.00** (1.5) |
| 3 | StatlogHeart | .708 (4) | .565 (5) | .801 (3) | **.874** (1.5) | **.874** (1.5) |
| 4 | WDBC | .611 (4) | .500 (5) | .860 (3) | **1.00** (1.5) | **1.00** (1.5) |
| 5 | Sports Articles | .336 (5) | .606 (4) | **.897** (1) | .872 (2.5) | .872 (2.5) |
| 6 | Parkinsons | .491 (5) | .607 (4) | .649 (3) | **1.00** (1.5) | **1.00** (1.5) |
| 7 | Dermatology | .305 (5) | .838 (3) | .717 (4) | **.972** (1.5) | **.972** (1.5) |
| 8 | Sonar | .834 (3) | .600 (5) | .797 (4) | **1.00** (1.5) | **1.00** (1.5) |
| 9 | Musk | .992 (3) | .439 (5) | .664 (4) | **1.00** (1.5) | **1.00** (1.5) |
| 10 | Colon Cancer | .340 (3) | .018 (5) | .273 (4) | .982 (1.5) | .982 (1.5) |
| AVG(ARI) | | .605 (4.2) | .580 (4.4) | .749 (3.2) | **.970** (1.6) | **.970** (1.6) |

On the aspects of accuracy and robustness in Table 5.15 and 5.16, *F_Yager* method produces better performance with a smaller parameter value $\omega$. When $\omega$ became close to 0 ($\omega = 0.001$ in this experiment), the *F_Yager* method produces the same best results as the drastic sum method *F_DS*. Besides, the *F_DS* method outperforms the other competitors on the majority of the datasets on robustness.

***III*. Multi-Criteria** Based on the comprehensive evaluation methods in Section 4.6, the methods of the fuzzy-based approach were compared using the multi-criteria performance, as shown in Table 5.17.

Table 5.17 Methods tuning of the fuzzy based approach on multi-criteria

| No. | Datasets | *F_Yager* with different $\omega$ | | | | *F_DS* |
|---|---|---|---|---|---|---|
| | | 1000 | 10 | 0.1 | 0.001 | |
| 1 | Mammographic | .382 (5) | .648 (4) | .750 (3) | **.885** (1.5) | **.885** (1.5) |
| 2 | PIMA | .561 (5) | .727 (4) | .786 (3) | **.819** (1.5) | **.819** (1.5) |
| 3 | StatlogHeart | .548 (4) | .507 (5) | .705 (3) | **.747** (1.5) | **.747** (1.5) |
| 4 | WDBC | .550 (5) | .558 (4) | .850 (3) | **.950** (1.5) | **.950** (1.5) |
| 5 | Sports Articles | .265 (5) | .552 (4) | **.775** (1) | .735 (2.5) | .735 (2.5) |
| 6 | Parkinsons | .427 (5) | .513 (4) | .560 (3) | **.865** (1.5) | **.865** (1.5) |
| 7 | Dermatology | .238 (5) | .670 (3) | .618 (4) | **.749** (1.5) | **.749** (1.5) |
| 8 | Sonar | .514 (3) | .394 (5) | .512 (4) | **.645** (1.5) | **.645** (1.5) |
| 9 | Musk | .681 (3) | .272 (5) | .423 (4) | **.685** (1.5) | **.685** (1.5) |
| 10 | Colon Cancer | .239 (4) | .047 (5) | .330 (3) | **.835** (1.5) | **.835** (1.5) |
| AVG(ARI) | | .441 (4.4) | .493 (4.3) | .631 (3.1) | **.792** (1.6) | **.792** (1.6) |

In Table 5.17, the multi-criteria performance of the *F_Yager* method increases with the decrease of parameter $\omega$ values in Yager class S-norm. When $\omega = 0.001$, the *F_Yager* method produces the best performance in 9 out of 10 datasets, the same as the *F_DS* method. It highlights the outstanding performance in applying the drastic sum fuzzy aggregation method. In summary, the fuzzy-based approach using Yager class S-norm *F_Yager* ($\omega$ close to 0) or drastic sum S-norm method *F_DS* produce the best performance using the different evaluation methods, including accuracy, robustness, and multi-criteria performance. As the drastic sum is a more straightforward expression than Yager class S-norms, the *F_DS* method is chosen for further performance analysis and comparison.

**5. Discussion** The optimal methods were selected based on accuracy, robustness, and, more importantly, multi-criteria performance. The $S\_S_{OW}$ method produced the overall best performance in the data repository of tuning for the score-based approach. For the rank-based approach, the $R\_S_{OW}$ method achieved the best performance. It indicates a superior performance by utilizing the standard deviation as the distribution index and OW as the weighting scheme in the score and rank-based framework. This may be on account that the standard deviation can better represent the uncertainty of feature importance in different FS methods. Through aggregating the various FS methods using the one minus weights, the ensemble framework can take the most advantages of the base selectors, which leads to better

overall performance. For the fuzzy-based approach, the *F_DS* method produced the best result. This may be because the drastic sum can help aggregate the various FS methods with all the inherent information, which leads to better performance, especially on the aspects of stability and robustness. Its multi-criteria performance is also enhanced. Therefore, $S\_S_{OW}$, $R\_S_{OW}$ and *F_DS* methods are chosen as the representative for performance analysis and the later research.

### 5.3.4 Performance Analysis on the Testing Data Repository

In this section, the proposed methods with score, rank and fuzzy-based approaches are evaluated on the testing data repository to further investigate and analyze the ensemble FS framework. Firstly, to verify the effectiveness of the aggregation approach, the performance of using each of the individual base selectors after the bootstrap process is produced as the baseline. Then they are also compared with using simpler and more straightforward approaches, such as average, etc. Thirdly, to objectively and fairly assess the proposed methods' performance, several other state-of-the-art FS methods are included for further comparison and analysis. Finally, the proposed methods are evaluated and compared with other widely used ensemble approaches.

**1. Comparison with the Base Selectors after Bootstrap** To explore the effects of the bootstrap process and the aggregation approach on FS results separately, the proposed method's performance was compared with the base selectors that utilized the same bootstrap process. The "base selectors after bootstrap" indicate the FS methods which utilize the bootstrap to generate the feature distributions and defuzzify without any combinations or aggregation procedures. In Table 5.18, 5.19 and 5.20, the operators (+/-) indicate that the bootstrap process has the increased (+), decreased (-) or same (no mark) performance compared to the base selectors without bootstrap. The numbers in brackets represent the

joint ranking index of the different FS approaches. The numbers in bold indicate the best performance of the given dataset. Based on the evaluation methods in Chapter 4, the performance is measured on three aspects, including accuracy, robustness and multi-criteria.

***I*. Accuracy** Performance analysis on accuracy is shown in Table 5.18.

Table 5.18 Performance analysis with the base selectors after bootstrap on accuracy

| No. | Datasets | Base Selectors After Bootstrap | | | | Proposed Ensemble Methods | | |
|---|---|---|---|---|---|---|---|---|
| | | CFS | ReliefF | MIFS | IFS | $S\_S_{OW}$ | $R\_S_{OW}$ | $F\_DS$ |
| 11 | Banknote | .851 (4) | .757 (5) | **.864** (2) | .607 (7) | .673 (6) | **.864** (2) | **.864** (2) |
| 12 | WBC | **.947** (1) | .945$^-$ (3) | .942$^+$ (6) | .944 (5) | .945 (3) | .945 (3) | .927 (7) |
| 13 | CMSC | .915 (4) | .915 (4) | .915 (4) | .915 (4) | .915 (4) | .915 (4) | .915 (4) |
| 14 | Appendicitis | .823$^-$ (4) | **.836**$^-$ (1) | .819$^+$ (5) | .806$^+$ (7) | .817 (6) | .825 (3) | .832 (2) |
| 15 | BCC | .307 (7) | .467$^-$ (3) | **.521**$^-$ (1) | .404$^-$ (4) | .319 (6) | .394 (5) | .519 (2) |
| 16 | Wine | .857$^-$ (2) | .738$^-$ (6) | **.877** (1) | .735$^-$ (7) | .846 (4) | .845 (5) | .847 (3) |
| 17 | Glass | .349 (3) | .256 (7) | **.383**$^+$ (1) | .305$^-$ (6) | .309 (5) | .341 (4) | .364 (2) |
| 18 | Spectfheart | .786$^+$ (4) | .782 (7) | .785$^-$ (5.5) | .789 (2) | .785 (5.5) | .787 (3) | **.791** (1) |
| 19 | Breast Tissue | .267 (7) | .303$^-$ (3) | .270$^+$ (6) | .297$^+$ (5) | .301 (4) | .304 (2) | **.340** (1) |
| 20 | Lung | .681 (3.5) | .681 (3.5) | .680$^+$ (6.5) | **.682** (1) | .681 (3.5) | .681 (3.5) | .680 (6.5) |
| AVG (ARI) | | .678$^-$ (3.95) | .668$^-$ (4.25) | .706$^+$ (3.8) | .648 (4.8) | .659 (4.7) | .690 (3.45) | **.708** (3.05) |

***II*. Robustness** Performance analysis on robustness is shown in Table 5.19.

Table 5.19 Performance analysis with the base selectors after bootstrap on robustness

| No. | Datasets | Base Selectors after Bootstrap | | | | Proposed Ensemble Methods | | |
|---|---|---|---|---|---|---|---|---|
| | | CFS | ReliefF | MIFS | IFS | $S\_S_{OW}$ | $R\_S_{OW}$ | $F\_DS$ |
| 11 | Banknote | .779$^-$ (7) | .934$^+$ (5) | .996 (3) | .911$^+$ (6) | **1.00** (1.5) | .970 (4) | **1.00** (1.5) |
| 12 | WBC | .928$^+$ (2) | .812$^+$ (7) | .905$^+$ (5) | .925$^+$ (3) | .904 (6) | .924 (4) | **1.00** (1) |
| 13 | CMSC | **1.00** (1.5) | .555$^+$ (4) | .549$^+$ (5) | -.005$^+$ (7) | .479 (6) | .867 (3) | **1.00** (1.5) |
| 14 | Appendicitis | .399$^-$ (7) | .513$^-$ (4) | .580$^+$ (3) | .677$^+$ (2) | .434 (6) | .495 (5) | **1.00** (1) |
| 15 | BCC | .326$^-$ (7) | .807$^+$ (3) | .358$^-$ (6) | .913$^+$ (2) | .488 (4) | .456 (5) | **1.00** (1) |
| 16 | Wine | .770$^+$ (7) | .920$^-$ (2) | .889$^+$ (5) | .851$^+$ (6) | .899 (4) | .909 (3) | **1.00** (1) |
| 17 | Glass | .895$^-$ (6) | .965$^+$ (2) | .801$^+$ (7) | .931$^+$ (5) | .942 (4) | .949 (3) | **1.00** (1) |
| 18 | Spectfheart | .564$^+$ (7) | .784$^+$ (3) | .596$^+$ (6) | .815 (2) | .773 (5) | .783 (4) | **1.00** (1) |
| 19 | Breast Tissue | .180$^-$ (7) | .971$^-$ (2) | .818$^+$ (5) | .868$^+$ (3) | .784 (6) | .823 (4) | **1.00** (1) |
| 20 | Lung | .578$^-$ (7) | .951$^+$ (2) | .621$^+$ (6) | .941$^+$ (3) | .891 (4) | .868 (5) | **1.00** (1) |
| AVG (ARI) | | .642$^-$ (5.85) | .821$^+$ (3.4) | .711$^+$ (5.1) | .783$^+$ (3.9) | .759 (4.65) | .804 (4) | **1.00** (1.1) |

In Table 5.18, the proposed $S\_S_{OW}$ and $R\_S_{OW}$ methods produce a comparable performance with the four base selectors after bootstrap. $F\_DS$ method produces the best performance, outperforming all the base selectors and the other ensemble methods on both AVG and ARI values. Besides, it is also observed that the bootstrap aggregation for individual

base selectors did not lead to higher accuracy in many cases on different datasets and FS methods. It indicates the improved accuracy mainly results from the aggregation methods rather than the bootstrap process. In Table 5.19, comparing the performance of base selectors before and after bootstrap aggregation, the bootstrap process leads to an increase in robustness in most datasets. However, one of the proposed methods, *F_DS*, further outperformed all the other competitors and produced significantly high performance in the testing data repository on the aspect of robustness.

***III*. Multi-Criteria** Based on the evaluation methods in Section 4.6, the performance is analyzed and compared using the multi-criteria evaluation method in Table 5.20.

Table 5.20 Performance analysis with the base selectors after bootstrap on multi-criteria

| No. | Datasets | Base Selectors After Bootstrap | | | | Proposed Ensemble Methods | | |
|---|---|---|---|---|---|---|---|---|
| | | CFS | ReliefF | MIFS | IFS | $S\_S_{OW}$ | $R\_S_{OW}$ | *F_DS* |
| 11 | Banknote | $.704^{-}$ (6) | $.739^{+}$ (5) | .907 (2) | $.634^{+}$ (7) | .782 (4) | .891 (3) | **.909** (1) |
| 12 | WBC | $.905^{+}$ (3) | $.812^{+}$ (7) | $.882^{+}$ (6) | $.900^{+}$ (4) | .887 (5) | .906 (2) | **.951** (1) |
| 13 | CMSC | **.943** (1.5) | $.588^{+}$ (4) | $.585^{+}$ (5) | $-.025^{+}$ (7) | .458 (6) | .833 (3) | **.943** (1.5) |
| 15 | Appendicitis | $.504^{+}$ (6) | $.556^{-}$ (4) | $.598^{+}$ (3) | $.635^{+}$ (2) | .458 (7) | .534 (5) | **.888** (1) |
| 16 | BCC | $.226^{-}$ (7) | $.536^{+}$ (3) | $.284^{-}$ (6) | $.559^{+}$ (2) | .295 (5) | .311 (4) | **.679** (1) |
| 17 | Wine | $.744^{-}$ (6) | $.751^{-}$ (5) | $.828^{+}$ (2) | $.678^{+}$ (7) | .823 (3.5) | .823 (3.5) | **.898** (1) |
| 18 | Glass | $.507^{-}$ (3) | $.481^{+}$ (6) | $.454^{+}$ (7) | $.496^{-}$ (5) | .505 (4) | .527 (2) | **.576** (1) |
| 19 | Spectfheart | $.556^{+}$ (7) | $.711^{+}$ (3) | $.578^{+}$ (6) | .735 (2) | .695 (5) | .704 (4) | **.861** (1) |
| 20 | Breast Tissue | $.115^{-}$ (7) | $.514^{-}$ (2) | $.421^{+}$ (5) | $.452^{+}$ (3) | .412 (6) | .448 (4) | **.560** (1) |
| 20 | Lung | $.484^{-}$ (7) | $.744^{+}$ (2) | $.521^{+}$ (6) | $.740^{+}$ (3) | .696 (4) | .685 (5) | **.786** (1) |
| AVG (ARI) | | $.569^{+}$ (5.35) | $.643^{+}$ (4.1) | $.606^{+}$ (4.8) | $.580^{+}$ (4.2) | .601 (4.95) | .666 (3.55) | **.798** (1.05) |

In Table 5.20, the proposed method using the fuzzy-based approach *F_DS* produced the best performance on all the datasets in the testing data repository, which significantly outperformed all the other competitors. It indicates a superior performance in utilizing the fuzzy-based approach with drastic sum S-norm aggregation. Another proposed method, $R\_S_{OW}$, also produced a good performance, which excels the four base selectors on AVG and ARI values. The score-based approach $S\_S_{OW}$ produced the middle-level performance with the base selectors, which is lower than ReliefF and MIFS; higher than CFS and IFS on the AVG values.

**2. Comparison with Other State-of-the-art FS Methods** In order to objectively and fairly evaluate the performance of the proposed ensemble methods, four state-of-the-art FS methods were utilized for performance analysis and comparison in Table 5.8, which are Lap Score, SPEC, Gini Index, and F Score methods. Based on the particular evaluation methods in Chapter 4, the performance is measured on three different aspects, including accuracy, robustness and multi-criteria. In Tables 5.21, 5.22 and 5.23, the numbers in brackets represent the joint ranking indices of each FS method compared with the others for the given dataset.

***I*. Accuracy** Performance analysis on accuracy is shown in Table 5.21.

Table 5.21 Performance analysis with the state-of-the-art FS methods on accuracy

| No. | Datasets | The State-of-the-art FS Methods | | | | Proposed Ensemble Methods | | |
|---|---|---|---|---|---|---|---|---|
| | | Lap Score | SPEC | Gini Index | F Score | $S\_S_{OW}$ | $R\_S_{OW}$ | $F\_DS$ |
| 11 | Banknote | .580 (6) | .556 (7) | **.864** (2.5) | **.864** (2.5) | .673 (5) | **.864** (2.5) | **.864** (2.5) |
| 12 | WBC | .940 (5) | .942 (3.5) | .942 (3.5) | .939 (6) | **.945** (1.5) | **.945** (1.5) | .927 (7) |
| 13 | CMSC | **.915** (4) | **.915** (4) | **.915** (4) | **.915** (4) | **.915** (4) | **.915** (4) | **.915** (4) |
| 14 | Appendicitis | .816 (4) | .801 (6.5) | .801 (6.5) | .814 (5) | .817 (3) | .825 (2) | **.832** (1) |
| 15 | BCC | .501 (4) | .397 (5) | .556 (2) | **.564** (1) | .319 (7) | .394 (6) | .519 (3) |
| 16 | Wine | .761 (7) | .785 (6) | .813 (5) | .831 (4) | .846 (2) | .845 (3) | **.847** (1) |
| 17 | Glass | .191 (7) | .277 (5) | **.364** (1.5) | .244 (6) | .309 (4) | .341 (3) | **.364** (1.5) |
| 18 | Spectfheart | .791 (2.5) | **.793** (1) | .779 (6.5) | .779 (6.5) | .785 (5) | .787 (4) | .791 (2.5) |
| 19 | Breast Tissue | .307 (3) | .249 (7) | **.340** (1.5) | .303 (5) | .301 (6) | .304 (4) | **.340** (1.5) |
| 20 | Lung | .680 (5.5) | **.683** (1) | .680 (5.5) | .680 (5.5) | .681 (2.5) | .681 (2.5) | .680 (5.5) |
| AVG (ARI) | | .648 (4.8) | .640 (4.6) | .705 (3.85) | .693 (4.55) | .659 (4) | .690 (3.25) | **.708** (2.95) |

***II*. Robustness** Performance analysis on robustness is shown in Table 5.22.

Table 5.22 Performance analysis with the state-of-the-art FS methods on robustness

| No. | Datasets | The State-of-the-art FS Methods | | | | Proposed Ensemble Methods | | |
|---|---|---|---|---|---|---|---|---|
| | | Lap Score | SPEC | Gini Index | F Score | $S\_S_{OW}$ | $R\_S_{OW}$ | $F\_DS$ |
| 11 | Banknote | **1.00** (3) | **1.00** (3) | **1.00** (3) | .992 (6) | **1.00** (3) | .970 (7) | **1.00** (3) |
| 12 | WBC | .708 (7) | .888 (5) | .895 (4) | .873 (6) | .904 (3) | .924 (2) | **1.00** (1) |
| 13 | CMSC | .066 (7) | .101 (6) | .546 (4) | .574 (3) | .479 (5) | .867 (2) | **1.00** (1) |
| 14 | Appendicitis | .660 (3) | .812 (2) | .617 (4) | .591 (5) | .434 (7) | .495 (6) | **1.00** (1) |
| 15 | BCC | .752 (2) | .396 (7) | .572 (4) | .632 (3) | .488 (5) | .456 (6) | **1.00** (1) |
| 16 | Wine | .441 (7) | .511 (6) | .823 (5) | .911 (2) | .899 (4) | .909 (3) | **1.00** (1) |
| 17 | Glass | .866 (6) | .874 (5) | .986 (2) | .860 (7) | .942 (4) | .949 (3) | **1.00** (1) |
| 18 | Spectfheart | .509 (6) | .164 (7) | .717 (5) | .791 (2) | .773 (4) | .783 (3) | **1.00** (1) |
| 19 | Breast Tissue | .440 (6) | .184 (7) | .910 (2) | .802 (4) | .784 (5) | .823 (3) | **1.00** (1) |
| 20 | Lung | .519 (6) | .211 (7) | .682 (5) | .801 (4) | .891 (2) | .868 (3) | **1.00** (1) |
| AVG (ARI) | | .596 (5.3) | .514 (5.5) | .775 (3.8) | .783 (4.2) | .759 (4.2) | .804 (3.8) | **1.00** (1.2) |

Compared with the performance of the state-of-the-art FS methods, the proposed ensemble methods *F_DS* and *R_S*$_{OW}$ produce the best and second-best overall performance in the testing data repository on the aspects of accuracy and robustness. In Table 5.21, the *F_DS* method produces the best performance on 6 out of 10 datasets. For robustness in Table 5.22, the *F_DS* method produces the best performance on all the testing datasets. Besides, the *R_S*$_{OW}$ method gets the overall second-best performance, which is better than any of the FS methods in comparison. It indicates a superior performance in utilizing the proposed ensemble learning framework.

*III*. **Multi-Criteria** Based on the evaluation methods in Section 4.6, the performance is analyzed and compared using the multi-criteria measurement, as shown in Table 5.23.

Table 5.23 Performance analysis with the state-of-the-art FS methods on multi-criteria

| No. | Datasets | The State-of-the-art FS Methods | | | | Proposed Ensemble Methods | | |
|---|---|---|---|---|---|---|---|---|
| | | Lap Score | SPEC | Gini Index | F Score | *S_S*$_{OW}$ | *R_S*$_{OW}$ | *F_DS* |
| 11 | Banknote | .720 (6) | .704 (7) | **.909** (1.5) | .905 (3) | .782 (5) | .891 (4) | **.909** (1.5) |
| 12 | WBC | .710 (7) | .877 (4) | .873 (5) | .830 (6) | .887 (3) | .906 (2) | **.951** (1) |
| 13 | CMSC | .180 (6) | .097 (7) | .563 (4) | .599 (3) | .458 (5) | .833 (2) | **.943** (1) |
| 14 | Appendicitis | .640 (3) | .729 (2) | .590 (4) | .571 (5) | .458 (7) | .534 (6) | **.888** (1) |
| 15 | BCC | .331 (4) | .245 (7) | .419 (3) | .440 (2) | .295 (6) | .311 (5) | **.679** (1) |
| 16 | Wine | .460 (7) | .513 (6) | .695 (5) | .805 (4) | .823 (2.5) | .823 (2.5) | **.898** (1) |
| 17 | Glass | .388 (7) | .451 (5) | .570 (2) | .417 (6) | .505 (4) | .527 (3) | **.576** (1) |
| 18 | Spectfheart | .346 (6) | .255 (7) | .661 (5) | .715 (2) | .695 (4) | .704 (3) | **.861** (1) |
| 19 | Breast Tissue | .166 (6) | .146 (7) | .520 (2) | .427 (4) | .412 (5) | .448 (3) | **.560** (1) |
| 20 | Lung | .327 (6) | .277 (7) | .498 (5) | .619 (4) | .696 (2) | .685 (3) | **.786** (1) |
| AVG (ARI) | | .427 (5.8) | .429 (5.9) | .630 (3.65) | .633 (3.9) | .601 (4.35) | .666 (3.35) | **.805** (1.05) |

In Table 5.23, the proposed method *F_DS* produces the best performance on all the testing datasets compared with the other competitors. *R_S*$_{OW}$ method produces the second-best performance on the testing data repository. The *S_S*$_{OW}$ method gets the middle-level performance among the four competitors, which is better than Lap Score and SPEC; worse than Gini Index and F Score methods. It demonstrates the proposed ensemble FS framework's superior performance compared with the state-of-the-art FS methods.

**3. Overall Comparison with Simple Approaches and Other Ensemble Methods** From the previous evaluation and comparison process, it can be seen that the fuzzy-based approach with drastic sum S-norm (*F_DS* method) has achieved the overall best performance on both the training and testing data repository. Nevertheless, to demonstrate the utility and effectiveness of the proposed method, it is further evaluated and compared with simpler aggregation approaches and other state-of-the-art ensemble techniques.

**3.1 Description of the simpler approaches and other ensemble methods** Within the fuzzy-based method, some simple approaches for aggregation are included and compared. By utilizing the different parameters within the max-min averages (introduced in Section 2.2.1), the minimal ($\lambda = 0$), average ($\lambda = 0.5$), and maximal ($\lambda = 1$) aggregation within the fuzzy-based method are used for performance comparison. Besides, two other ensemble FS methods are also included (i.e. random forest [122] and gradient-boosting decision tree (GBDT) [221]).

**3.2 Performance comparison using the multi-criteria evaluation metric** This section compares and assesses the *F_DS* method with the simpler approaches and other ensemble methods on multi-criteria performance. The KNN classifier was selected in the decision-making process because it can achieve high consistent results without a specialized training phase. Spearman's rank correlation coefficient was utilized in the stability and robustness measures within the multi-criteria performance evaluation. By implementing the multi-criteria performance in Section 4.6, the evaluation and comparison of the different FS methods are reported in Table 5.24.

From Table 5.24, it can be seen that except for the case of random forest in the Banknote dataset, the proposed method produced better performance on almost all the datasets compared with the simpler approaches and the two popular ensemble methods. Besides, the proposed method also achieved the best performance in the testing data repository, which

Table 5.24 Performance analysis with simpler approaches and other ensemble methods on multi-criteria

| No. | Datasets | Simpler Approaches | | | Other Ensemble Methods | | Proposed Method |
|---|---|---|---|---|---|---|---|
| | | Minimal | Average | Maximal | Random Forest | GBDT | *F_DS* |
| 11 | Banknote | .455 (6) | .878 (4) | .741 (5) | .918 (1.5) | .908 (3) | .918 (1.5) |
| 12 | WBC | .838 (3) | .757 (5) | .615 (6) | .887 (2) | .778 (4) | .948 (1) |
| 13 | CMSC | .247 (5) | .258 (4) | .153 (6) | .622 (2) | .533 (3) | .943 (1) |
| 14 | Appendicitis | .329 (5) | .369 (4) | .201 (6) | .661 (2) | .568 (3) | .873 (1) |
| 15 | BCC | .459 (2) | .342 (5) | .287 (6) | .431 (3) | .352 (4) | .746 (1) |
| 16 | Wine | .618 (2) | .353 (6) | .403 (5) | .608 (3) | .469 (4) | .781 (1) |
| 17 | Glass | .480 (3) | .503 (2) | .319 (5) | .452 (4) | .288 (6) | .569 (1) |
| 18 | Spectfheart | .522 (4) | .585 (2) | .363 (6) | .571 (3) | .377 (5) | .806 (1) |
| 19 | Breast Tissue | .363 (3) | .287 (5) | .142 (6) | .394 (2) | .320 (4) | .547 (1) |
| 20 | Lung | .722 (2) | .473 (4) | .198 (5) | .507 (3) | .057 (6) | .793 (1) |
| AVG (ARI) | | .503 (3.5) | .480 (4.1) | .342 (5.6) | .605 (2.55) | .465 (4.2) | .792 (1.05) |

outperformed others using both AVG and ARI values. Therefore, from the perspective of the multi-criteria performance, the proposed method is demonstrated to hold a better performance for FS, even compared with the other approaches and ensemble frameworks.

**3.3 Statistical analysis of the results** Additionally, McNemar's test is applied to test the statistical significance of whether the proposed method is better than the compared method. The P values of McNemar's test for the pairwise tests between the proposed method and each competitor are shown in Table 5.25.

Table 5.25 P values of McNemar's test for the pairwise test on whether the proposed method is better than the competitor on multi-criteria

| No. | Datasets | Simpler Approaches | | | Other Ensemble Methods | |
|---|---|---|---|---|---|---|
| | | Minimal | Average | Maximal | Random Forest | GBDT |
| 11 | Banknote | <0.01 | <0.01 | <0.01 | 1.00 | <0.01 |
| 12 | WBC | <0.01 | <0.01 | <0.01 | <0.01 | <0.01 |
| 13 | CMSC | <0.01 | <0.01 | <0.01 | <0.01 | <0.01 |
| 14 | Appendicitis | <0.01 | <0.01 | <0.01 | <0.01 | <0.01 |
| 15 | BCC | <0.01 | <0.01 | <0.01 | <0.01 | <0.01 |
| 16 | Wine | <0.01 | <0.01 | <0.01 | <0.01 | <0.01 |
| 17 | Glass | <0.01 | <0.01 | <0.01 | <0.01 | <0.01 |
| 18 | Spectfheart | <0.01 | <0.01 | <0.01 | <0.01 | <0.01 |
| 19 | Breast Tissue | <0.01 | <0.01 | <0.01 | <0.01 | <0.01 |
| 20 | Lung | <0.01 | <0.01 | <0.01 | <0.01 | <0.01 |

From Table 5.25, it is observed that the proposed method produces a significantly better performance than the other competitors, except for random forest in the Banknote dataset.

**4. Discussion** In the performance analysis process with the base selectors after bootstrap, the proposed ensemble FS method *F_DS* outperforms the base selectors. The ensemble methods *F_DS* and $R\_S_{OW}$ produce the overall best and second-best performance with the use of different evaluation metrics. In the performance analysis with the other state-of-the-art FS methods, *F_DS* and $R\_S_{OW}$ methods also outperform the four other state-of-the-art FS methods in terms of accuracy, robustness and multi-criteria performance. Remarkably, the *F_DS* method's performance on robustness is significantly higher than the others on almost all the testing datasets. Afterwards, the *F_DS* method is further compared and evaluated with the simpler approaches and some other ensemble FS methods. The proposed *F_DS* method has achieved significantly better performance on almost all the testing datasets. To sum up, in the testing data repository, the proposed ensemble FS framework has produced a comprehensively outstanding performance on multi-criteria, which indicates a reasonably good decision-making performance with a stable feature ranking sequence and a high level of robustness on the reduced size datasets.

## 5.4 Discussion

Four different state-of-the-art FS methods are utilized as the base selectors within the proposed ensemble framework. Those four base selectors are chosen and carefully picked up to represent the different categories of the ranking-based filter methods. Nevertheless, the proposed ensemble method is a dynamic framework where more diverse FS methods could be included. The proposed ensemble method aims to form a generalized framework to incorporate the different kinds of algorithms together. In addition to the utilized FS methods, more state-of-the-art FS techniques could also be employed for the performance comparison.

From the experiments in Section 5.3, it can be seen that the *F_DS* method has achieved relatively good performance on multi-criteria performance. At the same time, the *F_DS* method also produced significantly high scores on robustness. It may be because the drastic

sum aggregation method can eliminate the most noise and changes within the features' distributions. Rather than combining the distributions with all information in detail, the drastic sum aggregation focuses on the values' coverage and ignores the small changes within them. Therefore, its performance on stability and robustness can be significantly enhanced.

## 5.5 Summary

This chapter has proposed an ensemble learning framework to combine the different FS results, which consists of three main steps: distribution generation of feature importance, distribution ensemble using aggregation methods, and defuzzification for feature ranking. Different approaches are utilized to represent the features' importance by generating the various distributions using the bootstrap process, such as score-based, rank-based, and fuzzy-based approaches. Next, a distribution ensemble using aggregation methods is implemented with various weighted combinations and fuzzy aggregation methods. At last, the defuzzification process is implemented for feature ranking.

The training and testing data repository are introduced for tuning the methods and the performance analysis process. The proposed method is first evaluated on a training data repository to select the best aggregation methods for the score, rank, and fuzzy-based approaches. Based on the evaluation methods in accuracy, robustness, and multi-criteria performance, the $S\_S_{OW}$, $R\_S_{OW}$ and $F\_DS$ methods produced the best performance respectively and therefore were chosen as their representatives. Subsequently, their performance was evaluated and compared on the testing data repository with the base selector after bootstrap, other FS methods, the simpler approaches and two other ensemble techniques.

From the experimental results, the proposed method with the fuzzy-based approach using drastic sum S-norms ($F\_DS$) has produced the overall best performance on the testing data repository. Specifically, the performance values of $F\_DS$ on robustness are significantly higher than other methods, which indicates an excellent capability to deal with the variations

in the distribution generation process. For classification accuracy, *F_DS* also produced the overall best performance. It is observed that the bootstrap process for individual base selectors did not lead to higher accuracies for many tested datasets. Hence, the proposed ensemble method's improved performance is not due to the bootstrap process but mainly to the method aggregation step.

As for the other methods in the ensemble FS framework, the $R\_S_{OW}$ method has produced the second-best performance on the testing data repository. It is the proposed method using a rank-based approach with standard deviation as the distribution index and OW as the weighting scheme. The $R\_S_{OW}$ method also performed better than the $S\_S_{OW}$ method, which utilized the same aggregation method. It may be because the biases could be introduced when combining the feature scores from different FS methods with a score-based approach. In comparison, the rank-based approach's rank indices were consistent among different FS methods; hence they could be aggregated better from different FS methods without introducing a bias.

# Chapter 6

# Meta Learning Framework for Recommending Suitable FS Methods

## 6.1 Introduction

As introduced in Section 2.1, there are many different FS methods in the literature. Based on the dependency with the learning algorithms, FS methods are grouped into three types, i.e., filter, wrapper, and embedded methods. There are subset search-based and ranking-based methods within the filter methods. The ranking-based filter methods could also be generally grouped into four FS categories: similarity-based, information-based, statistical-based, and graph-based [128]. However, one of the main issues for applying those different FS methods is that the FS methods' performance varies in a data-dependent manner. It is not possible to state the best FS method categorically to provide the best performance for all kinds of data [86]. The relationship between the data and their FS performance is still unclear and lacks cognition in many cases. Many learning-based FS methods suffer from data scarcity for training with poor generalizability when applying to a new dataset. It poses an interesting and challenging problem to select the most suitable FS method for a given unseen dataset [156].

In this chapter, the meta-learning method is adopted to solve this problem by choosing the best algorithm for a given dataset [22]. It is typically trained to learn the relationship between the characteristics of training datasets and their best FS methods [156], which is precious and of great importance in the machine learning area. As discussed in Section 2.4.4, different meta-learning approaches exist, such as metric learning-based, parameter training-based, gradient optimization-based, memory augmentation-based, data augmentation-based, etc. In practice, one of the key challenges for machine learning-based FS method is the lack of good training data with diverse characteristics. Data augmentation and synthetic data generation have gradually become widely-used techniques to address this issue in the literature. In 2011, Albuquerque *et al.* presented a novel framework to generate the synthetic high-dimensional datasets [3]. In 2013, Bolon-Canedo *et al.* reviewed the different synthetic datasets in the FS research area [18]. In 2014, Tomas *et al.* proposed two strategies to generate synthetic multi-label datasets under the framework "Mldatagen", which is a publicly available data generator [204]. In 2018, Avino *et al.* proposed a new method to generate synthetic but plausible healthcare record datasets [6]. In 2018, Dandekar *et al.* performed a comparative study of different synthetic data generation techniques under various machine learning techniques [37]. In 2019, Kamilaris *et al.* described the preliminary work on generating synthetic training data for the deep learning models [103].

Therefore, on the basis of the different kinds of techniques from the literature, this chapter aims to develop an FS recommendation framework to suggest a suitable FS method for a given dataset. In order to overcome the data scarcity problem, the feasibility of using synthesized data for training a meta-learning framework is investigated and explored to achieve feature ranking.

## 6.2 Methodology

The overall framework of the meta learning method is shown in Fig. 6.1. Blue lines and red lines represent the data flow for the training and testing processes, respectively.

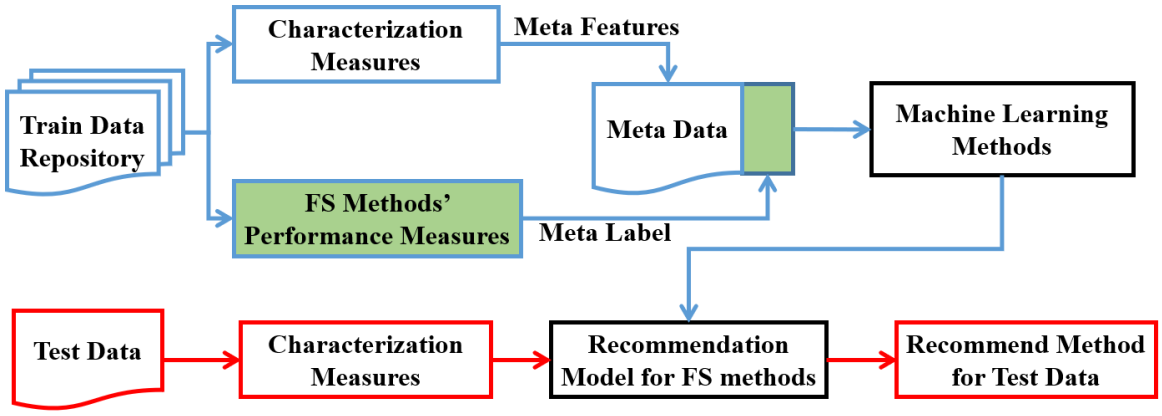



Fig. 6.1 The overall framework of the proposed meta learning method

The proposed method consists of four steps: training data repository generation, meta feature extraction, meta data construction, and recommendation modelling. The meta-features and meta labels are constructed on synthetic datasets in the training phase. After implementing and comparing different FS methods on the synthetic datasets, the meta label is generated to represent the most suitable FS method given a dataset. Another option is to recommend a ranked list of potentially suitable FS methods, making it a multi-label problem. To simplify the learning process and be more practical, only the most suitable method is recommended in the proposed method. More discussion is included in Chapter 7. Then, based on the synthetic datasets, a set of meta-features are extracted to represent the characteristics of the synthetic datasets. On account of its high flexibility, computational efficiency and good performance, a fuzzy similarity-based classifier from Chapter 3 is then adopted during the decision-making process. In the testing phase, the same meta-features are extracted from the testing dataset. By applying the trained recommendation model, the most suitable FS method is subsequently produced. The detailed information is described below.

### 6.2.1 Synthetic Training Data Repository Generation

In this section, a data repository is constructed to cover a variety of characteristics using data synthesis. Compared with the other synthetic datasets, Madelon dataset [69] is chosen on account of its high flexibility and variability. By changing the parameters, Madelon datasets can be designed to cover a wide range of values that are similar to real-world datasets. The detailed process of generating the synthetic datasets is described in Section 4.2.2.

### 6.2.2 Meta Feature Extraction

To learn meta features from the synthetic datasets, a set of meta features are extracted from $M$ different datasets $\mathbb{D}_i, i = 1, ..., M$. For the given dataset $\mathbb{D}_i$, it contains the number of $S_i$ data samples $(E_1, E_2, ..., E_{S_i})$ and $N_i$ features $(F_1, F_2, ..., F_{N_i})$. The corresponding label information is represented using class $C, (c_1, c_2, ..., c_{S_i})$ for different data samples. The overall illustration of the data structure is shown in Table 6.1. Subsequently, several different meta feature extraction methods are introduced as follows, which are derived from the $\mathbb{D}_i$ dataset [157].

Table 6.1 The structure of the generated synthetic dataset

| **Samples** | **Features** | | | | **Class** |
|---|---|---|---|---|---|
| | $F_1$ | $F_2$ | ... | $F_{N_i}$ | |
| $E_1$ | $v_{11}$ | $v_{12}$ | ... | $v_{1N_i}$ | $c_1$ |
| $E_2$ | $v_{21}$ | $v_{22}$ | ... | $v_{2N_i}$ | $c_2$ |
| **...** | ... | ... | ... | ... | ... |
| $E_{S_i}$ | $v_{S_i1}$ | $v_{S_i2}$ | ... | $v_{S_iN_i}$ | $c_{S_i}$ |

The datasets' standard measures are generally classified into different categories: simple, statistics-based, and information theory-based. These meta-features are formalized and shown as below [157]. Those features are suggested in the literature [157, 207] to best capture the characteristics of datasets from different aspects. Other feasible feature representation methods could be investigated in future work, which are discussed in Chapter 7.

**1. Simple measures** Those measures aim to describe the general characteristics of the datasets.

1. *Number of Samples (NS)*: The number of samples per feature of the dataset.
2. *Number of Features (NF)*: The number of features of the dataset.
3. *Number of Classes (NC)*: The number of classes of the dataset.
4. *Number of Samples per Feature (S/F)*: The number of samples per feature ($NS/NF$).
5. *Number of Samples per Class (S/C)*: The number of samples per class ($NS/NC$).

**2. Statistical based measures** Those measurements characterize a dataset by extracting the measures in the statistical area.

1. *Average Asymmetry of Features (AAF)*: It measures the average value of Pearson's asymmetry coefficient. The formulation quantitatively summarizes the skewness of distribution, as shown in Equation 6.1.

$$AAF(\mathbb{D}_i) = \frac{3}{N_i} \sum_{j=1}^{N_i} \frac{Mean(F_j) - Median(F_j)}{Std(F_j)} \tag{6.1}$$

   where $Mean(F_j)$, $Median(F_j)$ and $Std(F_j)$ indicate the average, median and standard deviation values of feature $F_j$ respectively. $j$ represents the index of the features.

2. *Average Correlation between Features (ACF)*: It measures the average value of Pearson's correlation coefficient between different features.

$$ACF(\mathbb{D}_i) = \frac{2}{N_i(N_i-1)} \sum_{j=1}^{N_i-1} \sum_{k=j+1}^{N_i} Pearson(F_j, F_k) \tag{6.2}$$

   where $Pearson(F_j, F_k)$ indicates the Pearson's correlation between feature $F_j$ and $F_k$.

3. *Average Coefficient of Variation of Features (ACVF)*: It measures the average coefficient of variation by the ratio of the standard deviation and the mean of the feature values.

$$ACVF(\mathbb{D}_i) = \frac{1}{N_i}\sum_{j=1}^{N_i}\frac{Std(F_j)}{Mean(F_j)} \tag{6.3}$$

**3. Information theory based measures** Those measures characterize the nominal features and their relationship to the class attribute.

1. *Feature Entropy*: The distribution's entropy of the feature in a dataset $\mathbb{D}$, which originates from the communication theory area, is defined in Equation 6.4.

$$Entropy(\mathbb{D}_i) = -\sum_{i=1}^{N_i}[p_i \times log_2(p_i)] \tag{6.4}$$

where $p_i, 1 \leq i \leq n$ represents the occurrence probability of the given feature.

2. *Average Entropy of Features (AEF)*: It measures the average amount of the information provided by each feature for the aims of class prediction.

$$AEF(\mathbb{D}_i) = \frac{1}{N_i}\sum_{k=1}^{N_i}Entropy(F_j) \tag{6.5}$$

where $Entropy(F_j)$ measures the distribution's entropy of feature $F_j$.

### 6.2.3 Meta Data Construction

Meta data consists of the extracted meta features and meta labels. Meta labels represent the FS method with the best performance.

**1. Meta label generation** This section aims to generate the meta label for each dataset in the training data repository. The derived meta-features and associated labels are used for model learning to recommend the best FS method for a given unseen dataset. In this situation, the label is set as the best FS method for a given training dataset. The comprehensive

evaluation method on multi-criteria performance (introduced in Section 4.6) is used to measure the FS performance. After implementing the FS methods ($FS_1, FS_2, ..., FS_L$) on the training data repository for different datasets $\mathbb{D}_i$, the results are illustrated in Table 6.2.

Table 6.2 Performance measures of FS methods on different datasets

| **Data** | **Different FS Methods** | | | | | | **Optimal Method** |
|---|---|---|---|---|---|---|---|
| | $FS_1$ | $FS_2$ | ... | $FS_l$ | ... | $FS_L$ | |
| $\mathbb{D}_1$ | $value_{11}$ | $value_{12}$ | ... | $value_{1l}$ | ... | $value_{1L}$ | $Opt_1$ |
| $\mathbb{D}_2$ | $value_{21}$ | $value_{22}$ | ... | $value_{2l}$ | ... | $value_{2L}$ | $Opt_2$ |
| ... | ... | ... | ... | ... | ... | ... | ... |
| $\mathbb{D}_i$ | $value_{i1}$ | $value_{i2}$ | ... | $value_{il}$ | ... | $value_{iL}$ | $Opt_i$ |
| ... | ... | ... | ... | ... | ... | ... | ... |
| $\mathbb{D}_M$ | $value_{M1}$ | $value_{M2}$ | ... | $value_{Ml}$ | ... | $value_{ML}$ | $Opt_M$ |

In Table 6.2, $value_{il}$ indicates the multi-criteria performance value on dataset $\mathbb{D}_i$ using FS method $FS_l$. The optimal method $Opt_i$ stands for the algorithm with the maximal multi-criteria performance on the dataset $\mathbb{D}_i$. It is worth noting that when an FS method is the second-best for all the datasets, it is a good choice but would not be featured in the meta-data as a possible choice. This proposed meta-learning method aims not to select a method that can generally perform well on most datasets but to select the best suitable method for a given dataset. If a method can be selected as the best option with high confidence for a given dataset, the user should not need to consider the second-best option. More discussion is added to Chapter 7 as future work for this point.

**2. Construction of meta data** Subsequently, the meta data is constructed by combining the calculated meta features $MF_p$, $(1 \leq p \leq P)$ and the meta label for each dataset $\mathbb{D}_i$. The decision-making model is trained to recommend the optimal FS method for a given dataset based on the meta data. Constructed meta data is next illustrated using Table 6.3.

Table 6.3 Demonstration of meta data

| Data | Meta Features | | | | | | Meta Target |
|---|---|---|---|---|---|---|---|
| | $MF_1$ | $MF_2$ | ... | $MF_p$ | ... | $MF_P$ | |
| $\mathbb{D}_1$ | $w_{1,1}$ | $w_{1,2}$ | ... | $w_{1,p}$ | ... | $w_{1,P}$ | $Opt_1$ |
| $\mathbb{D}_2$ | $w_{2,1}$ | $w_{2,1}$ | ... | $w_{2,p}$ | ... | $w_{2,P}$ | $Opt_2$ |
| ... | ... | ... | ... | ... | ... | ... | ... |
| $\mathbb{D}_i$ | $w_{i,1}$ | $w_{i,1}$ | ... | $w_{i,p}$ | ... | $w_{i,P}$ | $Opt_i$ |
| ... | ... | ... | ... | ... | ... | ... | ... |
| $\mathbb{D}_M$ | $w_{M,1}$ | $w_{M,1}$ | ... | $w_{M,p}$ | ... | $w_{M,P}$ | $Opt_M$ |

### 6.2.4 Recommendation using Fuzzy Similarity Measure

As introduced in Chapter 4, a fuzzy similarity measure-based framework is implemented to train the classification model using the generated meta dataset. The overall structure of the classification framework is illustrated in Fig. 6.2. The blue and red lines show data flows for the training and testing processes, respectively.

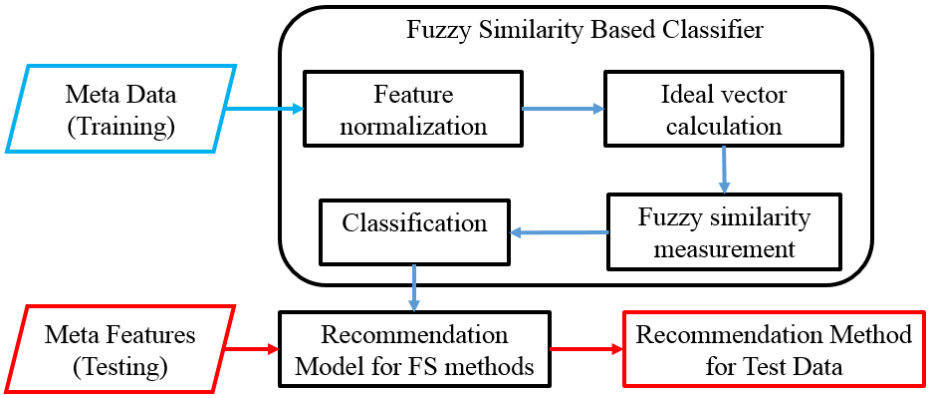



Fig. 6.2 The framework of fuzzy similarity-based classifier

The model training process aims to classify a total number of $M$ samples $D_i$ $(1 \leq i \leq M)$ into $L$ different classes $FS_l$, $(1 \leq l \leq L)$ by their feature vector $\vec{x_q}$. $q$ is the index of the data samples in each class. $C_l$ represents the number of data samples for the $l$th class. Based on the performance comparison results reported in Chapter 4, the similarity measure-based classifier is selected and implemented with the following steps.

Step 1: For the training set, standardize each feature using the normalization process [65].

Step 2: Based on the standardized values from Step 1, calculate the ideal vector $\vec{v}_l$ for the $l^{th}$ class using geometric mean.

$$\vec{v}_l(p) = \sqrt[C_l]{\prod_{q=1}^{C_l} \vec{x}_q(p)},\ 1 \leq p \leq P \tag{6.6}$$

where $p$ and $P$ represent the index and the total number of meta features respectively.

Step 3: The same standardization process from Step 1 is applied to the meta features extracted from the test dataset. Subsequently, the feature vector $\vec{y}_r$ of the meta features is produced, where $r$ indicates the index of the data samples in the test set.

Step 4: Based on the maximal fuzzy similarity measures introduced in Chapter 4, a similarity measurement in the form of generalized Łukasiewicz algebra is used. The geometric mean is used to combine the similarity measures from different features.

$$S\langle \vec{y}_r, \vec{v}_l \rangle = \sqrt[P]{\prod_{p=1}^{P} \sqrt{1 - |\vec{y}_r(p)^2 - \vec{v}_l(p)^2|}} \tag{6.7}$$

where $S\langle \vec{y}_r, \vec{v}_l \rangle$ represents the fuzzy similarity value between the feature vector of the testing set and the ideal vectors obtained from the training set.

Step 5: Classify the test datasets into the class with the corresponding ideal vector using the produced highest fuzzy similarity value.

The process can recommend the most suitable FS method for a given test dataset afterwards.

## 6.3 Experiments

### 6.3.1 Materials

This section introduces the data repositories and the FS methods for the experiments. The data repositories for parameters tuning and performance testing are firstly introduced and shown below.

**1. Data Repository for Parameters Tuning** Ten public datasets from the UCI machine learning repository with different data sparsity levels are chosen to help generate the synthesized data repository as realistic as possible for training. The general information about the datasets is shown in Table 6.4, with detailed information described in Section 4.2.

Table 6.4 Description of training data repository for methods tuning

| No. | Datasets | *#C* | *#F* | *#S* | *#S Distribution over #C* |
|---|---|---|---|---|---|
| 1 | Mammographic | 2 | 5 | 830 | 403 / 427 |
| 2 | PIMA | 2 | 8 | 768 | 268 / 500 |
| 3 | Statlog Heart | 2 | 13 | 270 | 120 / 150 |
| 4 | WDBC | 2 | 30 | 569 | 212 / 357 |
| 5 | Sports Articles | 2 | 59 | 1000 | 365 / 635 |
| 6 | Parkinsons | 2 | 22 | 195 | 48 / 147 |
| 7 | Dermatology | 6 | 34 | 358 | 20/48/48/60/71/111 |
| 8 | Sonar | 2 | 60 | 208 | 97/111 |
| 9 | Musk | 2 | 166 | 476 | 207/269 |
| 10 | Colon Cancer | 2 | 2000 | 62 | 22 / 40 |

where #C, #F and #S represent the number of classes, features, and samples, respectively. The data density or sparsity is calculated using the ratio between the number of samples, features, and classes ($\#S/\#C/\#F$).

**2. Data Repository for Performance Testing** Another ten datasets with various data sparsity are chosen for independent evaluation, as shown in Table 6.5. Those datasets are aimed for final performance analysis and comparison.

Table 6.5 Description of testing data repository for performance testing

| No. | Datasets | *#C* | *#F* | *#S* | *#S Distribution over #C* |
|---|---|---|---|---|---|
| 11 | Banknote | 2 | 4 | 1372 | 610 / 762 |
| 12 | WBC | 2 | 9 | 682 | 239 / 443 |
| 13 | CMSC | 2 | 18 | 540 | 46 / 494 |
| 14 | Appendicitis | 2 | 7 | 106 | 21 / 85 |
| 15 | BCC | 2 | 9 | 116 | 52 / 64 |
| 16 | Wine | 3 | 13 | 178 | 48 / 59 / 71 |
| 17 | Glass | 6 | 9 | 214 | 9 / 13 / 17 / 29 / 70 / 76 |
| 18 | Spectfheart | 2 | 44 | 267 | 55 / 212 |
| 19 | Breast Tissue | 6 | 9 | 106 | 14 / 15 / 16 / 18 / 21 / 22 |
| 20 | Lung | 5 | 3312 | 203 | 6 / 17 / 20 / 21 / 139 |

**3. Selection of FS methods** Four representative algorithms from different ranking-based filter FS categories were chosen and implemented in this experiment. They are carefully selected from different categories as a representative method for each category. The four FS methods are utilized as the preliminary test and evaluation of the framework. It helps to perform FS from different perspectives. Potentially, more methods could be included in future work, as the proposed meta-learning framework is sufficiently generic to include more methods. The general description is provided in Table 6.6.

Table 6.6 General description of the implemented FS methods

| No. | Alias | Name | Category | Supervision |
|---|---|---|---|---|
| 1 | CFS | Correlation based FS [73] | Statistical-based FS | Supervised |
| 2 | ReliefF | ReliefF FS [171] | Similarity-based FS | Supervised |
| 3 | MIFS | Mutual Information based FS [231] | Information-based FS | Supervised |
| 4 | IFS | Infinite based FS [175] | Graph-based FS | Unsupervised |

### 6.3.2 Synthetic Training Data Repository Generation

**1. Parameters setting of Madelon datasets** The general information and characteristics are extracted from the practical training data repository in Table 6.4, as listed as follows. (1) Number of classes ranges from 2 to 6. (2) Number of features ranges from 5 to 2000, whereas the majority of them are below 60. (3) Number of samples ranges between 62 and

1000. (4) Number of samples per class ranges from 31 to 500. (5) Number of samples per class per feature is between 0.02 to 83. Based on the extracted information, the parameter values in Madelon datasets were set correspondingly, as shown in Table 6.7.

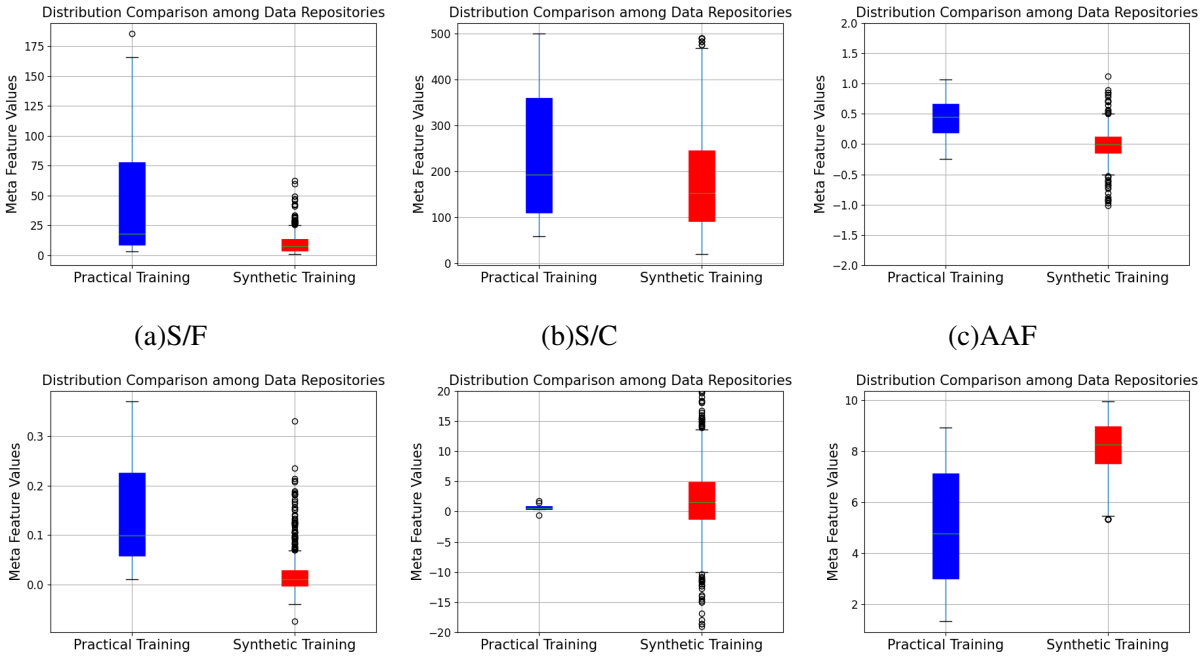
Table 6.7 Value range of the parameters in the Madelon dataset

| Alias | Meaning | Value Range |
|---|---|---|
| P1 | Number of Classes | [2, ..., 10] |
| P2 | Number of Useful Features | [4, 5,..., 20] |
| P3 | Number of Redundant Features | [0, 1,..., 20] |
| P4 | Number of Repeated Features | [0, 1,..., 20] |
| P5 | Number of Useless Features | [0, 1,..., 20] |
| P6 | Number of Samples per Cluster | [10, 11,..., 70] |
| P7 | Number of Cluster per Class | [2, 3,..., 7] |
| P8 | Random Seed | [1, 2,..., 1000] |
| P9 | Factor multiplying the hypercube dimension | [2, 3,..., 10] |
| P10 | Fraction of y labels to be randomly exchanged | [0.01, 0.02, ..., 0.1] |
| P11 | Flag to enable or disable random permutations | [0, 1] |

According to the procedures in Section 4.2.2, 1000 synthetic datasets were generated using the randomly chosen and combined parameter values in Table 6.7.

**2. Comparison of the meta feature distribution** Six meta-features were extracted from the 1000 synthetic datasets, which include Number of Samples per Feature (S/F), Number of Samples per Class (S/C), Average Asymmetry of Features (AAF), Average Correlation between Features (ACF), Average Coefficient of Variation of Features (ACVF) and Average Entropy of Features (AEF). Within the simple measures, the number of Samples per Feature (S/F) and the number of Samples per Class (S/C) are chosen to illustrate the distribution and density of the datasets. Besides, the average entropy of features (AEF), which measures the average performance on features' entropy, is utilized to represent the information theory-based measures. The same meta-features were also extracted from the practical training data repository in the meantime. Hence, the distribution comparison of the meta-features from

the practical and synthetic training data repositories is shown in Fig. 6.3. The blue and red bars represent the practical and synthetic training data repositories.




(a)S/F (b)S/C (c)AAF

(d)ACF (e)ACVF (f)AEF

Fig. 6.3 Distribution comparison of the meta features

It is seen that the distributions of the synthetic training repository cover the value range of the practical training datasets well on the meta-features S/C and ACVF. However, the synthetic data holds a slightly smaller range for the other meta-features than the practical datasets. Meta feature S/F and AEF of the synthetic datasets are lower or higher than the practical training datasets' corresponding value ranges. It can be further improved by fine adjusting the parameters in the data synthesis procedures.

**3. Comparison of the meta label distribution** Through implementing the FS methods in Table 6.6, the comprehensive evaluation metric from Section 4.6 was chosen for performance evaluation and comparison. The random forest was chosen as the downstream classifier based on its excellent performance and robustness to nonlinear datasets in the multi-criteria

performance evaluation method. Spearman's rank correlation coefficient was utilized in the stability and robustness evaluation metrics. After implementing the four FS methods on the training data repository, the number of times on achieving the best performance was 4 by CFS, 0 by ReliefF, 0 by MIFS, and 6 by IFS (a total of 10 datasets). Comparatively, the number of times on the best performance produced on the synthetic training data repository was 266 by CFS, 443 by ReliefF, 173 by MIFS, and 118 by IFS, respectively (a total of 1000 datasets). The comparison of the distributions on meta labels for practical and synthetic training data repositories is shown in Fig. 6.4. The blue and red bars represent the practical and synthetic training data repositories.

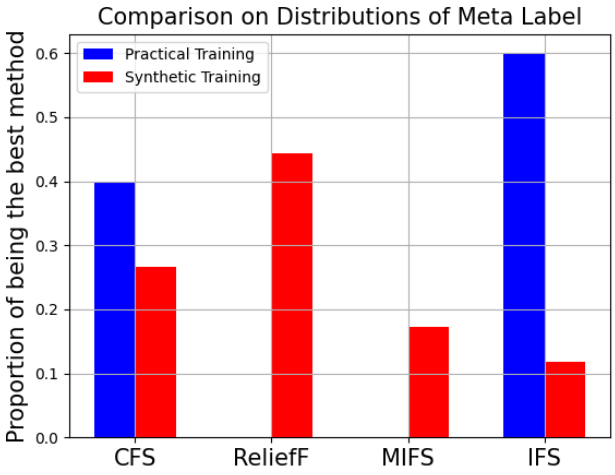



Fig. 6.4 Distribution comparison of the meta labels

From Fig. 6.4, in the practical training data repository, CFS and IFS were consistently selected as the best methods, whereas ReliefF and MIFS were never selected. However, the selected best method was distributed equally across all four methods for the synthetic datasets. It shows one of the benefits of using synthetic datasets for meta-learning rather than real datasets, as the synthetic datasets cover a wider variety of situations.

### 6.3.3 Performance Evaluation Results

**1. Recommendation of FS methods on Testing Data Repository** After constructing the meta data and applying the fuzzy similarity measure-based classifier, the meta-learning process was implemented on the testing data repository. The evaluation results on multi-criteria performance by applying each of the four FS methods to the test datasets are listed in Table 6.8. Comparatively, the FS method, which produced the highest multi-criteria performance value, was set as the ground truth ("Best Method" column). The recommended FS method suggested by the proposed meta-learning framework is listed in the last column of Table 6.8. The bold numbers indicate the best performance. The numbers with apostrophes represent the second-best performance.

Table 6.8 FS methods comparison and recommendation using multi-criteria performance

| No. | Datasets | FS Methods | | | | Best Method | Recommend |
|---|---|---|---|---|---|---|---|
| | | CFS | ReliefF | MIFS | IFS | | |
| 11 | Banknote | .705 | .707′ | **.907** | .632 | MIFS | MIFS |
| 12 | WBC | **.903** | .794 | .866 | .898 | CFS | CFS |
| 13 | CMSC | **.943** | .559 | .536 | -.038 | CFS | CFS |
| 14 | Appendicitis | .492 | .575′ | .477 | **.589** | IFS | ReliefF |
| 15 | BCC | .227 | .271 | .333′ | **.547** | IFS | ReliefF |
| 16 | Wine | .749 | **.821** | .809′ | .674 | ReliefF | MIFS |
| 17 | Glass | **.509** | .476 | .364 | .502′ | CFS | CFS |
| 18 | Spectfheart | .522 | .691′ | .390 | **.735** | IFS | IFS |
| 19 | Breast Tissue | .120 | **.535** | .388 | .415′ | ReliefF | IFS |
| 20 | Lung | .487 | **.735** | .507 | .731′ | ReliefF | ReliefF |

It can be observed that the proposed framework has successfully recommended the best method in 6 out of 10 datasets, which include Banknote, WBC, CMSC, Glass, Spectfheart and Lung. In Appendicitis, Wine, and Breast Tissue datasets, the recommended method ranked second, slightly lower than optimal. In the other dataset (i.e., BCC), the proposed method cannot accurately predict the best method. It may be on account that the distributions of the meta-features in the dataset lie outside the value ranges of the synthetic training data repository. Besides, the contribution of each meta feature to FS method recommendation may be highly dependent on whether the value distribution of the synthesized data covers the value

distribution of the real datasets. As shown in Fig. 6.3, there are mismatched distributions between the synthetic datasets and the real datasets for some of the meta-features.

However, the performance of the proposed framework can be further improved by refining the data synthesis process in Section 6.2.1. By tuning the parameters in the Madelon datasets, various kinds of synthetic datasets can be generated accordingly. It would help build a training data repository with a broader value range of the distributions within the meta-features and therefore contribute to the final performance. Nevertheless, a few aspects can be improved, which are discussed in the future work (Chapter 7).

**2. Comparison on the Recommended Best Methods** Through counting the number of the recommended best method in the testing data repository, the performance between the proposed method and the individual FS methods are displayed and compared in Fig. 6.5. It is seen that the proposed meta-learning method produced the best performance compared with the individual FS method. In the testing data repository, the proposed meta-learning framework successfully recommended the best method on 6 datasets out of 10. In contrast, the individual FS methods achieved the best performance in 3, 3, 1, and 3 cases, respectively.

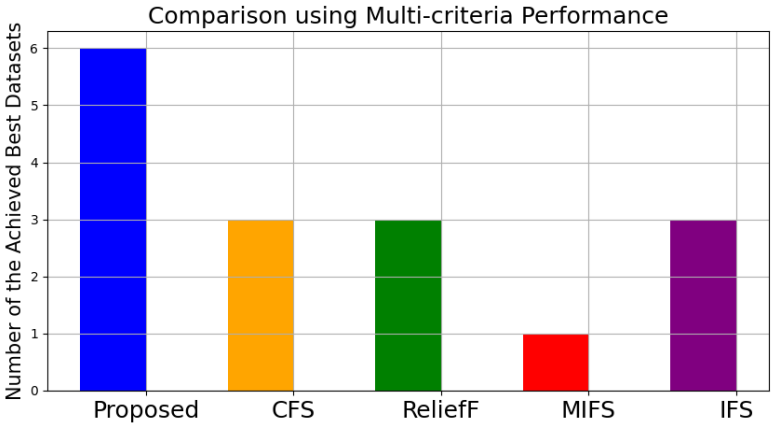



Fig. 6.5 Comparison on numbers of the achieved best methods

### 6.3.4 Evaluation on Computational Cost

This section reports and compares the execution time using FS methods and the meta-learning framework. The programs were implemented using Python and ran on a laptop with 2.6GHz, Intel(R) Core(TM) i7-10750H CPU, and 16GB RAM. By running each method ten times on the same dataset, the average execution time (s) is reported in Table 6.9.

Table 6.9 Average run-time using different methods (seconds)

| No. | Datasets | Individual Methods | | | | Meta Learning | Total Run Time |
|---|---|---|---|---|---|---|---|
| | | CFS | ReliefF | MIFS | IFS | | |
| 11 | Banknote | 5.39 | 86.91 | 6.82 | 4.35 | 0.95 | 7.77 |
| 12 | WBC | 8.59 | 33.18 | 11.03 | 8.29 | 0.02 | 8.61 |
| 13 | CMSC | 19.63 | 30.74 | 22.36 | 17.41 | 0.71 | 20.34 |
| 14 | Appendicitis | 6.00 | 6.92 | 7.11 | 6.24 | 0.02 | 6.94 |
| 15 | BCC | 7.80 | 8.76 | 9.27 | 7.90 | 0.03 | 8.79 |
| 16 | Wine | 11.49 | 13.43 | 14.18 | 11.27 | 0.05 | 14.23 |
| 17 | Glass | 8.33 | 12.74 | 11.39 | 8.14 | 0.04 | 8.37 |
| 18 | Spectfheart | 40.24 | 42.19 | 48.19 | 40.05 | 0.09 | 40.14 |
| 19 | Breast Tissue | 7.69 | 9.36 | 10.79 | 7.92 | 0.03 | 7.95 |
| 20 | Lung | 4532 | 4275 | 5180 | 6876 | 1.67 | 4277 |

In Table 6.9, the "Meta Learning" column indicates the execution time of the method selection. Subsequently, the whole FS process's total run time using the proposed method is the summation of the meta learning time and the recommended FS method run time, as shown in the column "Total Run Time" in Table 6.9.

It is seen that the meta-learning framework takes less than two seconds to run in all cases. The proposed meta-learning framework and the recommended FS method have consumed a moderate amount of time by inspecting the total run time and comparing it with the individual FS methods. It indicates a comparatively little additional computational cost incurred on implementing the meta-learning framework, which demonstrates the approach's applicability. The meta-learning method provides an efficient way to learn the potentially optimal FS method.

### 6.3.5 Discussion

The results show that the proposed method has successfully recommended the correct FS methods on 6 out of 10 testing datasets. However, it has to be admitted that the experimental result is only based on a small number of testing datasets. From the author's perspective, it is acknowledged that more datasets need to be included to investigate the proposed method's performance thoroughly. It is of great value to include the datasets from the different areas with various classes, features, samples and sample distributions.

One option is to generate synthetic datasets for evaluation. However, since the meta-learning process was performed on a large number of synthetic data, evaluation on the synthetic datasets with similar characteristics may result in over-optimistic results. On the other hand, more real-world datasets could be used to evaluate the proposed method's performance better. Because collecting and applying those real-world datasets are time-consuming, more evaluation will be performed in future work. Moreover, the code and trained model are freely available to encourage further development and evaluation on more datasets by the research community.

Another potential experiment is to add the ensemble learning method developed in Chapter 5 into the meta-learning framework. However, based on the experimental results from Chapter 5, the proposed ensemble method is able to achieve the best performance on almost all the testing datasets. Hence, it becomes unnecessary to utilize a meta-learning process to select the best FS method, which is known to be the ensemble learning method in most cases. On the other hand, on account that the ensemble learning method is time-consuming, it would take a long time to run during the training process. Therefore, ensemble learning has not been applied within the meta-learning approach in this research. The meta-learning and ensemble learning methods are treated as two alternative solutions for FS.

In addition, the fuzzy similarity-based classifier is utilized in this chapter, as it was used in previous chapters. The performance may also depend on the selection of the classifier.

However, the proposed meta-learning is a generic framework, indicating that other linear or non-linear classifiers could be used. More comparisons could be performed in future work.

As for the computational cost, the proposed method has not led to a significantly longer overall execution time, which indicates high computational efficiency. Apart from the very time-critical situations, the proposed method can be widely used and applied. Rather than randomly choosing one FS method, the pre-selection process using the meta-learning framework can produce a good performance with a minimal additional computational burden. It becomes an attractive potential method to be used when a wide variety of candidate algorithms are considered.

## 6.4 Summary

This chapter proposed a meta-learning method to recommend the most suitable FS method from four candidate algorithms. By using the data synthesis technique in constructing the training data repository, a new solution is developed within the data augmentation-based meta-learning method category, as introduced in Section 2.4.4. Instead of employing real-world datasets during the meta-learning process, 1000 different synthetic datasets were generated to form the training data repository. A practical training data repository that consists of ten datasets was used to tune the generation procedures' parameters. Six meta-features were extracted from the synthetic training data repository. The performance of the FS methods was measured using the multi-criteria measurement introduced in Chapter 4. A fuzzy similarity measure-based framework from Chapter 3 was applied to train a classification model based on the generated meta datasets. The proposed method has successfully recommended the FS method with the highest performance on six datasets by evaluating the proposed method on ten independent testing datasets from real-world applications. The performance is better than any of the individual FS methods. Besides, the proposed method is computationally efficient with little additional time cost to the FS process.

# Chapter 7

# Conclusions

This chapter concludes the thesis by summarising the main points and research outcomes. Further, this chapter presents a discussion concerning potential limitations and outlines some avenues for future work.

## 7.1 Thesis Summary

As stated in Chapter 1, the essential research aim of this thesis is to explore solutions for a better understanding of feature importance that represents the inherent characteristics of diverse datasets well and achieves good performance as assessed from a more comprehensive perspective that just performance alone. The objectives are identified as follows.

1. *Explore a ranking-based FS method that incorporates fuzzy theory to handle the various uncertainties that may be present, thereby producing good predictive performance.*

   In the popular fuzzy entropy-based FS framework proposed by Luukka, the original framework consists of three fundamental components: ideal vector calculation, similarity measurement, and fuzzy entropy calculation. Different measures and functions were introduced in the framework, and the various performance evaluated. This

novel comprehensive comparison of components, whilst not conclusive, provided valuable insights into the use of fuzzy components and how they might be used in later developments.

2. *Review the existing performance evaluation metrics for FS method comparison and propose new metrics if supplementary to the existing methods.*

   This objective led to a fundamental evaluation of various performance metrics that may be utilized to provide a more comprehensive overall picture of how different FS methods compare, taking into account robustness and stability, in addition to the standard assessment of performance.

3. *Develop an ensemble learning framework that combines different FS methods to achieve better feature ranking.*

   An ensemble learning method was developed to seek better performance. By aggregating the output of several base FS algorithms, the ensemble FS method was intended to better assess the importance of each feature, thereby achieving comprehensively good performance.

4. *As an alternative to this ensemble learning framework, develop an FS recommendation framework to suggest a suitable FS method for a given dataset. In order to overcome the data scarcity problem, the feasibility of using synthesized data for training a meta-learning framework to achieve feature ranking is also investigated and evaluated on various real datasets.*

   There are a large number of FS methods. It is impossible to conclude the optimal FS method in the sense of providing the best performance for all kinds of data. The relationship between the data and the corresponding performance of the various FS methods available is unclear. Besides, many FS methods suffer from data scarcity within the training process, resulting in poor generalization when being applied to a

new dataset. This experimentation suggested that a valuable and interesting problem would be to try to identify the most suitable FS method for any unseen dataset.

Chapter 2 reviewed background material and existing literature in four separate sections: FS methods, fuzzy theory, ensemble learning and meta-learning. FS methods aim to discover the minimal feature subset to represent the original data with different procedures and approaches. The use of FS methods leads to many benefits in real-world applications. Nevertheless, the presence of noise and uncertainty is arguably inevitable in practice. Therefore, the fuzzy theory was considered as a unified framework to model the vagueness, imprecision, and uncertainty present in this context. By providing a brief review of fuzzy sets, fuzzy entropy, fuzzy similarity, and fuzzy systems, standard notations and definitions within fuzzy theory were established. Furthermore, this chapter reviewed the commonly used ensemble learning and meta-learning from the literature to understand the feature importance better, and to develop a better understanding of combination methods for FS, together with other potential approaches to recommendation frameworks. Their background, techniques and main issues are introduced and discussed afterwards.

Chapter 3 investigated a fuzzy entropy-based FS method of Luukka's framework on three widely used public datasets. Different methods were implemented and compared in each framework's key components with three ideal vector calculations, three maximal similarity classifiers, and three fuzzy entropy functions. Based on the experimental results, the optimal method was selected using the geometric method for ideal vector calculation ($p = 2$), the geometric method for similarity classifier ($p = 2$), and Luca's method for fuzzy entropy calculation. Compared with the other six classical filter-based FS algorithms, the selected method produced a reasonably good performance. This chapter inspired selecting the best combinations in the FS framework, which led to the meta-learning method in Chapter 6. Besides, the introduced fuzzy similarity-based classification method was later applied as the downstream decision-making method used in Chapter 6.

In Chapter 4, after reviewing the existing performance evaluation metrics for the comparison of alternative FS methods, several evaluation metrics were proposed to take into account the various aspects of accuracy, robustness, and stability, together with a mechanism to combine them into an overall multi-criteria evaluation. The general calculation procedures for each proposed evaluation method were presented, together with a set of experiments to demonstrate their effectiveness. This chapter also introduced a multi-criteria evaluation method using the concept of radar charts to comprehensively measure the performance of FS algorithms in a user-accessible manner.

In Chapter 5, an ensemble learning framework was proposed to combine the FS results. The framework consists of three main steps: distribution generation of feature importance, distribution ensemble using aggregation methods, and defuzzification for feature ranking. Different distribution generation and aggregation methods were implemented for the ensemble learning and decision-making process. Ten public datasets were used as the training data repository for methods tuning. Another ten datasets were utilized as the testing data repository for independent performance testing. The optimal combination was selected for the score, rank, and fuzzy-based approaches, respectively. Subsequently, the proposed methods were evaluated on the testing data repository for performance analysis and comparison.

Chapter 6 addressed the last objective by developing an FS recommendation framework. A meta-learning method was proposed to recommend the most suitable FS algorithms from the candidate ones. Rather than utilize real-world datasets for training, 1000 synthetic datasets were generated as the training data repository. A practical training data repository that consists of ten datasets was used for parameter tuning in the generation procedures. Six meta-features were extracted afterwards. By utilizing the comprehensive performance framework introduced in Chapter 4 as the meta label and the fuzzy similarity measure-based method in Chapter 3 as the classifier, the proposed method successfully recommends the best FS method on six out of ten datasets with a reasonably high computational efficiency.

## 7.2 Contributions

The key contributions of this thesis are summarized in this section.

**1. Optimized method of a fuzzy entropy-based FS and classification framework**

Chapter 3 investigates a fuzzy entropy-based FS method with different ideal vector calculations, maximal similarity classifiers, and fuzzy entropy functions. Based on the experimental results, the optimized method is found to be based on the geometric method for ideal vector calculation ($p = 2$), the geometric method for similarity classifier ($p = 2$), and Luca's method for fuzzy entropy calculation. This chapter also concludes that the FS approach that eliminates the feature with the lowest entropy value each time produces better performance.

**2. New evaluation methods on the aspects of accuracy, robustness, and multi-criteria performance**

Various measures are used to evaluate the FS results in the literature on accuracy and stability. However, those measures could not comprehensively evaluate the performance of the various FS methods in a comprehensive manner. Furthermore, the robustness of FS techniques is a consideration that has received relatively little attention in the past.

Chapter 4 has proposed several new metrics to evaluate FS results on the accuracy, robustness, and multi-criteria performance. For the individual evaluation metrics such as accuracy and robustness, a higher weight was assigned to the evaluation values with more important features. Based on the experiments, those measurements are more reasonable and sensitive than the commonly used evaluation metrics, such as the mean and maximum. Besides, the proposed evaluation method on multi-criteria performance provides a comprehensive view of FS methods. A radar chart that incorporates the various aspects has been utilized to represent and visualize the comprehensive performance.

The evaluation methods are utilized to compare FS methods in Chapters 5 and 6. Moreover, those measurements are also used in a set of papers published in the FUZZ-IEEE conferences [188, 189].

**3. An ensemble learning framework for aggregating FS methods**

In the literature, ensemble FS methods are generally categorised into homogeneous and heterogeneous approaches. Chapter 5 proposes a novel ensemble learning framework that simultaneously incorporates homogeneous and heterogeneous approaches. It consists of three main steps: distribution generation of feature importance, distribution ensemble using aggregation methods, and defuzzification for feature ranking. The distributions of feature importance are generated using score, rank, and fuzzy-based approaches. Different aggregation methods are implemented, including weighted combination and fuzzy aggregation methods. After the method tuning process using ten real-world datasets, the optimal FS combinations were chosen based upon the score, rank, and fuzzy-based approach. The methods which utilize standard deviation as the 'distribution index' and 'one minus weights' as the weighting scheme produce the best performance for both score and rank-based approaches. The rank-based approach produces better performance than the score-based approach, which may introduce biases when integrating the feature scores from different FS methods. The fuzzy-based approach with a drastic sum S-norm produces the highest performance on the testing data repository, even compared with the other state-of-the-art FS methods.

**4. A meta learning framework based on data synthesis**

The meta-learning process learns the knowledge from a data repository in this research. As introduced in Section 2.4.3, the data repository construction becomes a fundamental issue. Collecting various datasets from real-world applications is a time-consuming process and customarily restricted by ethical issues, especially in the biomedical area. From the literature, various real-world data sources may need different types of consent, such as

informed consent, broad consent and implied consent [54, 202], which therefore decrease the availability of those datasets.

A training data repository was constructed to cover a variety of characteristics using data synthesis. The Madelon dataset was chosen based on its high flexibility and variability. It can be generated from an empty matrix by gradually adding useful, redundant, repeated, and useless features. This chapter constructs a training data repository with ten public datasets for parameter tuning in the synthetic training datasets. Therefore, the synthetic training data repository can generate a broader distribution of meta-features and nearly equally distributed meta labels. Besides, there are many parameters and steps to construct the synthetic datasets. Different synthetic datasets can be composed in the training phase by changing the parameter values, including or excluding the specific generation steps.

## 7.3 Limitations

The limitations of the work in this thesis are discussed in this section.

**1. Performance optimization of the fuzzy entropy based FS framework**

Chapter 3 remains a very early work in this thesis with some shortcomings and limitations which could be further improved in the future. The experiments were implemented only on three real-world datasets, which is insufficient to cover the datasets with different numbers of classes, features, samples, and sample distributions. Besides, the performance comparison process focuses on classification accuracy using the metrics, such as the highest achieved accuracy. The evaluation methods proposed in Chapter 4 have not been utilized and applied here. Furthermore, the evaluation results of this chapter are also not straightforward compared with those in Chapters 5 and 6.

**2. The comprehensive evaluation method for multi-criteria performance** The multi-criteria evaluation method favours the algorithm which produces a balanced performance in different aspects. Hence, a method with significantly higher performance on one aspect may not stand out by the multi-criteria performance. Using radar charts is also sensitive to the order of the coordinating axes when the number of the evaluation aspects is more than 3. This issue would need in-depth exploration, if the concept of multi-criteria radar charts was to be developed further.

Besides, it should be emphasized that the proposed evaluation metrics cannot be applied to the methods without generating a feature ranking. The weighted accuracy, robustness and multi-criteria performance measures are applied with the prerequisites that the feature ranking sequences are constructed.

On the other hand, in Section 4.6, only three real-world datasets have been used for the demonstration of the multi-criteria performance. Four state-of-the-art FS methods are utilized with kNN as the classifier and Spearman's rank correlation coefficient as the similarity measurement. Nevertheless, those choices do not sufficiently represent the wide range of potential different real-world situations. This experimentation still needs to be significantly expanded to thoroughly investigate the utility of the proposed normalized area measure.

**3. The computational cost of ensemble learning framework**

In Chapter 5, the ensemble learning method is rather time-consuming because of the bootstrap aggregation process. Even though the base selectors are chosen from the filter FS approaches, well-known for their high computational efficiency, the overall ensemble learning process still significantly increases the computational time. The running time of ensemble learning is proportional to the number of bootstraps. Many bagging processes are needed to describe feature importance distributions accurately, which inevitably will lead to an increase in computational cost.

**4. The performance of meta learning framework**

Firstly, it can be seen that the distributions of meta features generated in the synthetic training data repository could not fully and perfectly cover the value range of those found in the practical training data repository. The synthetic dataset holds a slightly smaller range than the practical datasets in respect of the different meta-features. It may prevent the proposed method from achieving the best performance. Besides, whether these features are sufficient for determining the best FS method still remains ambiguous. More research needs to be applied on whether enough information has been encoded within the datasets to allow the proper selection of the best FS method. More efforts also need to be made to generate synthetic datasets which are more similar to the real ones.

Secondly, the meta label is generated within the meta-learning framework to represent the FS method with the best performance. This option probably leads it to be overly restrictive. Nevertheless, it is worth noting that when an FS method is second-best for all the datasets, it is still a good choice but would not feature in the meta-data as a possible choice. Therefore, it would make more sense to make this into a multi-label problem and be able to recommend more than one method.

Thirdly, only four FS methods have been chosen as the representative algorithms from different ranking-based filter FS categories. The number of FS methods in the experiment thus remains quite limited. More FS algorithms could be included by principle. Therefore, to sum up, these aspects indicate that the results in Chapter 6 are still provisional and need to be further explored in the future, especially if the approach is to be more widely used.

## 7.4 Future Work

This section presents some potential new directions and future work based on this thesis.

**1. Further investigation on the fuzzy entropy based FS framework**

In Chapter 3, the fuzzy entropy-based FS methods need to be further evaluated on an extensive number of datasets with different classes, features, samples, and sample distributions. Various datasets should be included for the performance evaluation and comparison, such as different real-world or even synthetic datasets. Besides, the performance comparison process can be enhanced using the proposed evaluation methods from Chapter 4. The fuzzy entropy-based FS methods should be evaluated comprehensively using the proposed weighted accuracy, robustness and even multi-criteria performance.

**2. Further investigation on the proposed evaluation metrics**

In Chapter 4, the comprehensive and statistical properties of the proposed measurements need to be thoroughly investigated. In the future, more FS methods from various categories should be experimented upon utilizing a large volume of datasets using the proposed evaluation methods. More datasets with different classes, features, samples, and sample distributions, such as real-world and synthetic datasets, need to be included. It would help establish a comprehensive view of the FS methods, which could enhance the FS method selection in future research.

**3. Further investigation on the ensemble learning framework**

The research in Chapter 5 can be explored and investigated to further improve the performance of the methods, especially on accuracy. Other aggregation methods from the score, rank and fuzzy-based approaches could be experimented with and implemented on more extensive training and testing data repositories. Furthermore, the bootstrap aggregation process needs to be further accelerated to improve the computational efficiency of the proposed methods, and thereby reduce their computational cost.

**4. Further investigation on the meta learning framework**

The overall performance of the meta-learning frameworks in Chapter 6 can be further improved with the following procedures. Firstly, more extensive and diverse synthetic datasets should be generated by including and choosing different steps and parameters in the Madelon datasets generation process. It would help to build a training data repository with a broader range of values of the distributions and therefore contribute to the final performance. As mentioned before, the proliferation of synthetic training data repositories could provide a broader range of distributions for the meta-features. This work can also use evaluation metrics to compare the practical and synthetic training data repositories, such as the Wilcoxon sign rank test, etc. It is of great value to generate the synthetic datasets as naturally as possible and improve the performance of the meta-learning framework.

Secondly, more meta-features can be included from the literature to comprehensively evaluate the characteristics of a wider set of datasets, with respect to various aspects such as skewness, kurtosis, sparsity, Fisher's discriminant, landmark, etc. Besides, more real-world datasets in practice or from the literature could be used to evaluate and compare the performance of the proposed method.

Thirdly, other decision-making methods could be applied instead of utilizing a fuzzy similarity classifier, such as random forest, support vector machine, etc. Their predictive performance could be evaluated and compared with the fuzzy similarity classifier. Besides, other meta-learning frameworks can also be investigated and compared with the proposed method using the same training and testing data repository.

**5. Evaluation on the computational cost using the synthetic datasets**

Although the performance of FS methods is investigated thoroughly with respect to the accuracy, robustness and multi-criteria performance in this thesis, their computational efficiency properties have not been well studied yet. The characteristics of the computational cost in various FS methods from different categories are still unclear. One possible underlying problem in measuring the computational cost is that the running time of different

FS methods depends on the experimental datasets. Their inherent characteristics could also affect computational performance, such as the number of samples, features, classes, etc. A set of synthetic datasets could be generated for performance evaluation and comparison to understand their relationship better. For example, to investigate how the performance scales according to the sample size, the synthetic datasets can be generated by increasing the number of samples in a controlled manner. Therefore, an in-depth study can be introduced to analyze the performance.

# References


[1] Abeel, T., Helleputte, T., Van de Peer, Y., Dupont, P., and Saeys, Y. (2009). Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. *Bioinformatics*, 26(3):392–398.

[2] Aeberhard, S., Coomans, D., and de Vel, O. (1992). The classification performance of rda. *Dept. of Computer Science and Dept. of Mathematics and Statistics, James Cook University of North Queensland, Tech. Rep*, pages 92–01.

[3] Albuquerque, G., Lowe, T., and Magnor, M. (2011). Synthetic generation of high-dimensional datasets. *IEEE transactions on visualization and computer graphics*, 17(12):2317–2324.

[4] Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D., and Levine, A. J. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences*, 96(12):6745–6750.

[5] Altidor, W. (2011). *Stability analysis of feature selection approaches with low quality data*. Florida Atlantic University.

[6] Aviñó, L., Ruffini, M., and Gavaldà, R. (2018). Generating synthetic but plausible healthcare record datasets. *arXiv preprint arXiv:1807.01514*.

[7] Babbie, E. R. (2020). *The practice of social research*. Cengage learning.

[8] Bandemer, H. and Näther, W. (2012). *Fuzzy data analysis*, volume 20. Springer Science & Business Media.

[9] Battiti, R. (1994). Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on neural networks*, 5(4):537–550.

[10] Bellman, R. (1966). Dynamic programming. *Science*, 153(3731):34–37.

[11] Benesty, J., Chen, J., Huang, Y., and Cohen, I. (2009). Pearson correlation coefficient. In *Noise reduction in speech processing*, pages 1–4. Springer.

[12] Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828.

[13] Bennett, B. and Underwood, R. (1970). 283. note: On mcnemar's test for the 2 * 2 table and its power function. *Biometrics*, pages 339–343.

[14] Bentz, H., Hagstroem, M., and Palm, G. (1997). Selection of relevant features and examples in machine learning. *Neural Networks*, 2(4):289–293.

[15] Blumer, A., Ehrenfeucht, A., Haussler, D., and Warmuth, M. K. (1987). Occam's razor. *Information processing letters*, 24(6):377–380.

[16] Bolón-Canedo, V. and Alonso-Betanzos, A. (2019). Ensembles for feature selection: a review and future trends. *Information Fusion*, 52:1–12.

[17] Bolón-Canedo, V., Sánchez-Maroño, N., and Alonso-Betanzos, A. (2012). An ensemble of filters and classifiers for microarray data classification. *Pattern Recognition*, 45(1):531–539.

[18] Bolón-Canedo, V., Sánchez-Maroño, N., and Alonso-Betanzos, A. (2013). A review of feature selection methods on synthetic data. *Knowledge and information systems*, 34(3):483–519.

[19] Bosch, N. and Paquette, L. (2018). Metrics for discrete student models: Chance levels, comparisons, and use cases. *Journal of Learning Analytics*, 5(2):86–104.

[20] Boulesteix, A.-L. and Slawski, M. (2009). Stability and aggregation of ranked gene lists. *Briefings in bioinformatics*, 10(5):556–568.

[21] Brahim, A. B. and Limam, M. (2018). Ensemble feature selection for high dimensional data: a new method and a comparative study. *Advances in Data Analysis and Classification*, 12(4):937–952.

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

[23] Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123–140.

[24] Brown, G. (2004). *Diversity in neural network ensembles*. PhD thesis, Citeseer.

[25] Brown, G. (2011). Ensemble learning. *Encyclopedia of Machine Learning*.

[26] Brown, G., Pocock, A., Zhao, M.-J., and Luján, M. (2012). Conditional likelihood maximisation: a unifying framework for information theoretic feature selection. *The journal of machine learning research*, 13(1):27–66.

[27] Cai, Q., Pan, Y., Yao, T., Yan, C., and Mei, T. (2018). Memory matching networks for one-shot image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4080–4088.

[28] Cai, Z., Goebel, R., Salavatipour, M. R., Shi, Y., Xu, L., and Lin, G. (2007). Selecting genes with dissimilar discrimination strength for sample class prediction. In *Proceedings Of The 5th Asia-Pacific Bioinformatics Conference*, pages 81–90. World Scientific.

[29] Caruana, R. and Niculescu-Mizil, A. (2004). Data mining in metric space: an empirical analysis of supervised learning performance criteria. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 69–78.

[30] Chandrashekar, G. and Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28.

[31] Chen, C., Twycross, J., and Garibaldi, J. M. (2017). A new accuracy measure based on bounded relative error for time series forecasting. *PloS one*, 12(3):e0174202.

[32] Chouchoulas, A. and Shen, Q. (2001). Rough set-aided keyword reduction for text categorization. *Applied Artificial Intelligence*, 15(9):843–873.

[33] Cornelis, C., De Cock, M., and Radzikowska, A. M. (2008). Fuzzy rough sets: from theory into practice. *Handbook of Granular computing*, pages 533–552.

[34] Criado, F. and Gachechiladze, T. (1997). Entropy of fuzzy events. *Fuzzy Sets and Systems*, 88(1):99–106.

[35] Cruz, R. M., Sabourin, R., and Cavalcanti, G. D. (2017). Meta-des. oracle: Meta-learning and feature selection for dynamic ensemble selection. *Information fusion*, 38:84–103.

[36] Czelakowski, J. (2017). The infinite-valued łukasiewicz logic and probability. *Bulletin of the Section of Logic*, 46(1/2).

[37] Dandekar, A., Zen, R. A., and Bressan, S. (2018). A comparative study of synthetic dataset generation techniques. In *International Conference on Database and Expert Systems Applications*, pages 387–395. Springer.

[38] Dash, M. and Liu, H. (1997). Feature selection for classification. *Intelligent data analysis*, 1(3):131–156.

[39] Davis, J. C. and Sampson, R. J. (1986). *Statistics and data analysis in geology*, volume 646. Wiley New York et al.

[40] De Luca, A. and Termini, S. (1972). A definition of a nonprobabilistic entropy in the setting of fuzzy sets theory. *Information and control*, 20(4):301–312.

[41] Dernoncourt, F. (2013). Introduction to fuzzy logic. *Massachusetts Institute of Technology*, 21.

[42] Dietterich, T. G., Lathrop, R. H., and Lozano-Pérez, T. (1997). Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, 89(1-2):31–71.

[43] Dobre, C. and Xhafa, F. (2014). Parallel programming paradigms and frameworks in big data era. *International Journal of Parallel Programming*, 42(5):710–738.

[44] Dombi, J. (1982). A general class of fuzzy operators, the demorgan class of fuzzy operators and fuzziness measures induced by fuzzy operators. *Fuzzy sets and systems*, 8(2):149–163.

[45] Du, W., Cao, Z., Song, T., Li, Y., and Liang, Y. (2017). A feature selection method based on multiple kernel learning with expression profiles of different types. *BioData mining*, 10(1):4.

[46] Dua, D. and Graff, C. (2017). UCI machine learning repository.

[47] Duda, R. O., Hart, P. E., et al. (1973). *Pattern classification and scene analysis*, volume 3. Wiley New York.

[48] Duda, R. O., Hart, P. E., and Stork, D. G. (2012). *Pattern classification*. John Wiley & Sons.

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

[50] Dy, J. G. and Brodley, C. E. (2004). Feature selection for unsupervised learning. *Journal of machine learning research*, 5(Aug):845–889.

[51] Elad, M. and Aharon, M. (2006). Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image processing*, 15(12):3736–3745.

[52] Elter, M., Schulz-Wendtland, R., and Wittenberg, T. (2007). The prediction of breast cancer biopsy outcomes using two cad approaches that both emphasize an intelligible decision process. *Medical physics*, 34(11):4164–4172.

[53] Evett, I. W. and Spiehler, E. J. (1989). Rule induction in forensic science. In *Knowledge Based Systems*, pages 152–160.

[54] Facca, D., Smith, M. J., Shelley, J., Lizotte, D., and Donelle, L. (2020). Exploring the ethical issues in research using digital data collection strategies with minors: A scoping review. *Plos one*, 15(8):e0237875.

[55] Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37–37.

[56] Filchenkov, A. and Pendryak, A. (2015). Datasets meta-feature description for recommending feature selection algorithm. In *2015 Artificial Intelligence and Natural Language and Information Extraction, Social Media and Web Search FRUCT Conference (AINL-ISMW FRUCT)*, pages 11–18. IEEE.

[57] Finn, C., Abbeel, P., and Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR.

[58] Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188.

[59] Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139.

[60] Gao, H., Shou, Z., Zareian, A., Zhang, H., and Chang, S.-F. (2018). Low-shot learning via covariance-preserving adversarial augmentation networks. *Advances in Neural Information Processing Systems*, 31.

[61] Gini, C. W. (1971). Variability and mutability, contribution to the study of statistical distributions and relations. studi cconomico-giuridici della r. universita de cagliari (1912). reviewed in: Light, rj, margolin, bh: An analysis of variance for categorical data. *J. American Statistical Association*, 66:534–544.

[62] Goh, W. W. B. and Wong, L. (2016). Evaluating feature-selection stability in next-generation proteomics. *Journal of bioinformatics and computational biology*, 14(05):1650029.

[63] Gorman, R. P. and Sejnowski, T. J. (1988). Analysis of hidden units in a layered network trained to classify sonar targets. *Neural networks*, 1(1):75–89.

[64] Grimmett, G., Grimmett, G. R., Stirzaker, D., et al. (2001). *Probability and random processes*. Oxford university press.

[65] Grus, J. (2019). *Data science from scratch: first principles with python*. O'Reilly Media.

[66] Güvenir, H. A., Demiröz, G., and Ilter, N. (1998). Learning differential diagnosis of erythemato-squamous diseases using voting feature intervals. *Artificial intelligence in medicine*, 13(3):147–165.

[67] Guyon, I. (2003). Design of experiments of the nips 2003 variable selection benchmark. In *NIPS 2003 workshop on feature extraction and feature selection*, volume 253.

[68] Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182.

[69] Guyon, I., Gunn, S., Nikravesh, M., and Zadeh, L. A. (2008). *Feature extraction: foundations and applications*, volume 207. Springer.

[70] Guzmán-Martínez, R. and Alaiz-Rodríguez, R. (2011). Feature selection stability assessment based on the jensen-shannon divergence. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 597–612. Springer.

[71] Haibe-Kains, B., El-Hachem, N., Birkbak, N. J., Jin, A. C., Beck, A. H., Aerts, H. J., and Quackenbush, J. (2013). Inconsistency in large pharmacogenomic studies. *Nature*, 504(7480):389–393.

[72] Hajj, N., Rizk, Y., and Awad, M. (2019). A subjectivity classification framework for sports articles using improved cortical algorithms. *Neural Computing and Applications*, 31(11):8069–8085.

[73] Hall, M. A. (1999). Correlation-based feature selection for machine learning.

[74] Hamer, V. and Dupont, P. (2021). An importance weighted feature selection stability measure. *Journal of Machine Learning Research*, 22(116):1–57.

[75] Haque, M. N., Noman, N., Berretta, R., and Moscato, P. (2016). Heterogeneous ensemble combination search using genetic algorithm for class imbalanced data classification. *PloS one*, 11(1):e0146116.

[76] Hariharan, B. and Girshick, R. (2017). Low-shot visual recognition by shrinking and hallucinating features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3018–3027.

[77] Hariri, R. H., Fredericks, E. M., and Bowers, K. M. (2019). Uncertainty in big data analytics: survey, opportunities, and challenges. *Journal of Big Data*, 6(1):1–16.

[78] Hartmann, T., Moawad, A., Schockaert, C., Fouquet, F., and Le Traon, Y. (2019). Meta-modelling meta-learning. In *2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems (MODELS)*, pages 300–305. IEEE.

[79] Hartung, J., Knapp, G., and Sinha, B. K. (2011). *Statistical meta-analysis with applications*, volume 738. John Wiley & Sons.

[80] He, H. and Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284.

[81] He, X., Cai, D., and Niyogi, P. (2006). Laplacian score for feature selection. In *Advances in neural information processing systems*, pages 507–514.

[82] Hinton, G. E., Sejnowski, T. J., et al. (1999). *Unsupervised learning: foundations of neural computation*. MIT press.

[83] Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.

[84] Hosni, M., Idri, A., and Abran, A. (2017). Investigating heterogeneous ensembles with filter feature selection for software effort estimation. In *Proceedings of the 27th international workshop on software measurement and 12th international conference on software process and product measurement*, pages 207–220.

[85] Hosni, M., Idri, A., and Abran, A. (2021). On the value of filter feature selection techniques in homogeneous ensembles effort estimation. *Journal of Software: Evolution and Process*, page e2343.

[86] Hua, J., Tembe, W. D., and Dougherty, E. R. (2009). Performance of feature-selection methods in the classification of high-dimension data. *Pattern Recognition*, 42(3):409–424.

[87] Jain, A., Nandakumar, K., and Ross, A. (2005). Score normalization in multimodal biometric systems. *Pattern recognition*, 38(12):2270–2285.

[88] Jain, Y. K. and Bhandare, S. K. (2011). Min max normalization based data perturbation method for privacy protection. *International Journal of Computer & Communication Technology*, 2(8):45–50.

[89] Jeni, L. A., Cohn, J. F., and De La Torre, F. (2013). Facing imbalanced data–recommendations for the use of performance metrics. In *2013 Humaine association conference on affective computing and intelligent interaction*, pages 245–251. IEEE.

[90] Jensen, R. (2005). *Combining rough and fuzzy sets for feature selection*. PhD thesis, Citeseer.

[91] Jensen, R. (2008). Rough set-based feature selection: a review. *Rough computing: theories, technologies and applications*, pages 70–107.

[92] Jensen, R. and Shen, Q. (2004a). Fuzzy-rough attribute reduction with application to web categorization. *Fuzzy sets and systems*, 141(3):469–485.

[93] Jensen, R. and Shen, Q. (2004b). Semantics-preserving dimensionality reduction: rough and fuzzy-rough-based approaches. *IEEE Transactions on knowledge and data engineering*, 16(12):1457–1471.

[94] Jensen, R. and Shen, Q. (2007). Fuzzy-rough sets assisted attribute selection. *IEEE Transactions on fuzzy systems*, 15(1):73–89.

[95] Jin, X., Xu, A., Bie, R., and Guo, P. (2006). Machine learning techniques and chi-square feature selection for cancer classification using sage gene expression profiles. In *International Workshop on Data Mining for Biomedical Applications*, pages 106–115. Springer.

[96] Joachims, T. (2002). Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142. ACM.

[97] John, G. H., Kohavi, R., and Pfleger, K. (1994). Irrelevant features and the subset selection problem. In *Machine Learning Proceedings 1994*, pages 121–129. Elsevier.

[98] Jossinet, J. (1996). Variability of impedivity in normal and pathological breast tissue. *Medical and biological engineering and computing*, 34(5):346–350.

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

[100] Kalousis, A. and Hilario, M. (2001). Feature selection for meta-learning. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 222–233. Springer.

[101] Kalousis, A., Prados, J., and Hilario, M. (2005). Stability of feature selection algorithms. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, pages 8–pp. IEEE.

[102] Kalousis, A., Prados, J., and Hilario, M. (2007). Stability of feature selection algorithms: a study on high-dimensional spaces. *Knowledge and information systems*, 12(1):95–116.

[103] Kamilaris, A., Brink, C. v. d., and Karatsiolis, S. (2019). Training deep learning models via synthetic data: application in unmanned aerial vehicles. In *International Conference on Computer Analysis of Images and Patterns*, pages 81–90. Springer.

[104] Karegowda, A. G., Manjunath, A., and Jayaram, M. (2010). Comparative study of attribute selection using gain ratio and correlation based feature selection. *International Journal of Information Technology and Knowledge Management*, 2(2):271–277.

[105] Katz, G., Shin, E. C. R., and Song, D. (2016). Explorekit: Automatic feature generation and selection. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 979–984. IEEE.

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

[107] Kira, K., Rendell, L. A., et al. (1992). The feature selection problem: Traditional methods and a new algorithm. In *Aaai*, volume 2, pages 129–134.

[108] Kittler, J. (1978). Feature set search algorithms. *Pattern recognition and signal processing*.

[109] Koch, G., Zemel, R., Salakhutdinov, R., et al. (2015). Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2, page 0. Lille.

[110] Kohavi, R. and John, G. H. (1997). Wrappers for feature subset selection. *Artificial intelligence*, 97(1-2):273–324.

[111] Kolde, R., Laur, S., Adler, P., and Vilo, J. (2012). Robust rank aggregation for gene list integration and meta-analysis. *Bioinformatics*, 28(4):573–580.

[112] Koller, D. and Sahami, M. (1996). Toward optimal feature selection. Technical report, Stanford InfoLab.

[113] Kononenko, I. (2001). Machine learning for medical diagnosis: history, state of the art and perspective. *Artificial Intelligence in medicine*, 23(1):89–109.

[114] Kosko, B. (1986). Fuzzy entropy and conditioning. *Information sciences*, 40(2):165–174.

[115] Kosko, B. (1990). Fuzziness vs. probability. *International Journal of General System*, 17(2-3):211–240.

[116] Kou, G., Yang, P., Peng, Y., Xiao, F., Chen, Y., and Alsaadi, F. E. (2020). Evaluation of feature selection methods for text classification with small datasets using multiple criteria decision-making methods. *Applied Soft Computing*, 86:105836.

[117] Kück, M., Crone, S. F., and Freitag, M. (2016). Meta-learning with neural networks and landmarking for forecasting model selection an empirical evaluation of different feature sets applied to industry data. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 1499–1506. IEEE.

[118] Kuhn, M., Johnson, K., et al. (2013). *Applied predictive modeling*, volume 26. Springer.

[119] Kumar, V. and Minz, S. (2014). Feature selection: A literature review. *SmartCR*, 4(3):211–229.

[120] Kuncheva, L. I. (2007). A stability index for feature selection. In *Artificial intelligence and applications*, pages 421–427.

[121] Kurgan, L. A., Cios, K. J., Tadeusiewicz, R., Ogiela, M., and Goodenday, L. S. (2001). Knowledge discovery approach to automated cardiac spect diagnosis. *Artificial intelligence in medicine*, 23(2):149–169.

[122] Kursa, M. B. and Rudnicki, W. R. (2011). The all relevant feature selection using random forest. *arXiv preprint arXiv:1106.5112*.

[123] Lee, C. and Lee, G. G. (2006). Information gain and divergence-based feature selection for machine learning-based text categorization. *Information processing & management*, 42(1):155–165.

[124] Lemke, C., Budka, M., and Gabrys, B. (2015). Metalearning: a survey of trends and technologies. *Artificial intelligence review*, 44(1):117–130.

[125] Lemke, C. and Gabrys, B. (2010). Meta-learning for time series forecasting and forecast combination. *Neurocomputing*, 73(10-12):2006–2016.

[126] Leondes, C. T. (1998). *Fuzzy logic and expert systems applications*. Elsevier.

[127] Li, J. (2007). Feature construction, selection and consolidation for knowledge discovery.

[128] Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., and Liu, H. (2017). Feature selection: A data perspective. *ACM Computing Surveys (CSUR)*, 50(6):1–45.

[129] Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., and Liu, H. (2018). Feature selection: A data perspective. *ACM Computing Surveys (CSUR)*, 50(6):94.

[130] Li, R.-P., Mukaidono, M., and Turksen, I. B. (2002). A fuzzy neural network for pattern classification and feature selection. *Fuzzy Sets and Systems*, 130(1):101–108.

[131] Li, Y., Gao, S., and Chen, S. (2012). Ensemble feature weighting based on local learning and diversity. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.

[132] Liaw, A., Wiener, M., et al. (2002). Classification and regression by randomforest. *R news*, 2(3):18–22.

[133] Little, M. A., McSharry, P. E., Roberts, S. J., Costello, D. A., and Moroz, I. M. (2007). Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection. *Biomedical engineering online*, 6(1):23.

[134] Liu, H. and Motoda, H. (2007). *Computational methods of feature selection*. CRC Press.

[135] Liu, H. and Motoda, H. (2012). *Feature selection for knowledge discovery and data mining*, volume 454. Springer Science & Business Media.

[136] Liu, H., Sun, J., Liu, L., and Zhang, H. (2009). Feature selection with dynamic mutual information. *Pattern Recognition*, 42(7):1330–1339.

[137] Lucas, D., Klein, R., Tannahill, J., Ivanova, D., Brandon, S., Domyancic, D., and Zhang, Y. (2013). Failure analysis of parameter-induced simulation crashes in climate models. *Geoscientific Model Development*, 6(4):1157–1171.

[138] Luukka, P. (2011). Feature selection using fuzzy entropy measures with similarity classifier. *Expert Systems with Applications*, 38(4):4600–4607.

[139] Luukka, P., Saastamoinen, K., and Kononen, V. (2001). A classifier based on the maximal fuzzy similarity in the generalized lukasiewicz-structure. In *Fuzzy Systems, 2001. The 10th IEEE International Conference on*, volume 1, pages 195–198. IEEE.

[140] Mankad, K. B. (2017). An intelligent process development using fusion of genetic algorithm with fuzzy logic. In *Artificial Intelligence: Concepts, Methodologies, Tools, and Applications*, pages 245–281. IGI Global.

[141] Mendel, J. M. (1995). Fuzzy logic systems for engineering: a tutorial. *Proceedings of the IEEE*, 83(3):345–377.

[142] Mesiar, R. and Kolesárová, A. (2018). Aggregation functions in fuzzy set theory: History and some recent advances. In *2018 6th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS)*, pages 94–97. IEEE.

[143] Mika, S., Ratsch, G., Weston, J., Scholkopf, B., and Mullers, K. R. (1999). Fisher discriminant analysis with kernels. In *Neural Networks for Signal Processing IX, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop*.

[144] Minaei-Bidgoli, B., Asadi, M., and Parvin, H. (2011). An ensemble based approach for feature selection. In *Engineering applications of neural networks*, pages 240–246. Springer.

[145] Mitchell, L., Sloan, T. M., Mewissen, M., Ghazal, P., Forster, T., Piotrowski, M., and Trew, A. (2014). Parallel classification and feature selection in microarray data using sprint. *Concurrency and computation: practice and experience*, 26(4):854–865.

[146] Morán-Fernández, L., Bolón-Canedo, V., and Alonso-Betanzos, A. (2017). Centralized vs. distributed feature selection methods based on data complexity measures. *Knowledge-Based Systems*, 117:27–45.

[147] Narendra, P. M. and Fukunaga, K. (1977). A branch and bound algorithm for feature subset selection. *IEEE Transactions on computers*, (9):917–922.

[148] Ni, R., Goldblum, M., Sharaf, A., Kong, K., and Goldstein, T. (2021). Data augmentation for meta-learning. In *International Conference on Machine Learning*, pages 8152–8161. PMLR.

[149] Nie, F., Xiang, S., Jia, Y., Zhang, C., and Yan, S. (2008). Trace ratio criterion for feature selection. In *AAAI*, volume 2, pages 671–676.

[150] Nikulin, V. (2012). On the homogeneous ensembling via balanced subsets combined with wilcoxon-based feature selection. In *International Conference on Rough Sets and Current Trends in Computing*, pages 455–462. Springer.

[151] Nogueira, S. and Brown, G. (2016). Measuring the stability of feature selection. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 442–457. Springer.

[152] Novák, V. (1990). On the syntactico-semantical completeness of first-order fuzzy logic. ii. main results. *Kybernetika*, 26(2):134–154.

[153] Paige, R. F. (1997). A meta-method for formal method integration. In *International Symposium of Formal Methods Europe*, pages 473–494. Springer.

[154] Pan, S. J. and Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

[155] Parkash, O., Sharma, P., and Mahajan, R. (2008). New measures of weighted fuzzy entropy and their applications for the study of maximum weighted fuzzy entropy principle. *Information Sciences*, 178(11):2389–2395.

[156] Parmezan, A. R. S., Lee, H. D., and Wu, F. C. (2017). Metalearning for choosing feature selection algorithms in data mining: Proposal of a new framework. *Expert Systems with Applications*, 75:1–24.

[157] Parmezana, A. R. S., Leeb, H. D., and Wub, F. C. (2016). Supplementary material for metalearning for choosing feature selection algorithms in data mining: Proposal of a new framework.

[158] Patrício, M., Pereira, J., Crisóstomo, J., Matafome, P., Gomes, M., Seiça, R., and Caramelo, F. (2018). Using resistin, glucose, age and bmi to predict the presence of breast cancer. *BMC cancer*, 18(1):1–8.

[159] Pawlak, Z. (1982). Rough sets. *International journal of computer & information sciences*, 11(5):341–356.

[160] Pelánek, R. (2015). Metrics for evaluation of student models. *Journal of Educational Data Mining*, 7(2):1–19.

[161] Peng, H., Long, F., and Ding, C. (2005). Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (8):1226–1238.

[162] Pes, B., Dessì, N., and Angioni, M. (2017). Exploiting the ensemble paradigm for stable feature selection: a case study on high-dimensional genomic data. *Information Fusion*, 35:132–147.

[163] Petković, M., Kocev, D., and Džeroski, S. (2020). Feature ranking for multi-target regression. *Machine Learning*, 109(6):1179–1204.

[164] Polat, K. and Güneş, S. (2009). A new feature selection method on classification of medical datasets: Kernel f-score feature selection. *Expert Systems with Applications*, 36(7):10367–10373.

[165] Porter, M. M. and Niksiar, P. (2018). Multidimensional mechanics: Performance mapping of natural biological systems using permutated radar charts. *PloS one*, 13(9):e0204309.

[166] Ravi, S. and Larochelle, H. (2016). Optimization as a model for few-shot learning.

[167] Reif, M., Shafait, F., and Dengel, A. (2012). Meta-learning for evolutionary parameter optimization of classifiers. *Machine learning*, 87(3):357–380.

[168] Reif, M., Shafait, F., Goldstein, M., Breuel, T., and Dengel, A. (2014). Automatic classifier selection for non-experts. *Pattern Analysis and Applications*, 17(1):83–96.

[169] Rezaee, M. R., Goedhart, B., Lelieveldt, B. P., and Reiber, J. H. (1999). Fuzzy feature selection. *Pattern Recognition*, 32(12):2011–2019.

[170] Rich, E. and Knight, K. (1992). *Artificial Intelligence: Instructor's Manual*. McGraw-Hill.

[171] Robnik-Šikonja, M. and Kononenko, I. (2003). Theoretical and empirical analysis of relieff and rrelieff. *Machine learning*, 53(1-2):23–69.

[172] Roffo, G. (2017). Ranking to learn and learning to rank: On the role of ranking in pattern recognition applications. *arXiv preprint arXiv:1706.05933*.

[173] Roffo, G. and Melzi, S. (2016). Features selection via eigenvector centrality. *Proceedings of new frontiers in mining complex patterns (NFMCP 2016)*.

[174] Roffo, G., Melzi, S., Castellani, U., and Vinciarelli, A. (2017). Infinite latent feature selection: A probabilistic latent graph-based ranking approach. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1398–1406.

[175] Roffo, G., Melzi, S., and Cristani, M. (2015). Infinite feature selection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4202–4210.

[176] Rogers, R. D. (1985). The western information society. In *New Information Technologies and Libraries*, pages 11–18. Springer.

[177] Russell, S. and Norvig, P. (2002). Artificial intelligence: a modern approach.

[178] Saastamoinen, K. and Luukka, P. (2003). Testing continuous t-norm called lukasiewicz algebra with different means in classification. In *Fuzzy Systems, 2003. FUZZ'03. The 12th IEEE International Conference on*, volume 2, pages 808–813. IEEE.

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

[180] Santoro, A., Bartunov, S., Botvinick, M., Wierstra, D., and Lillicrap, T. (2016). Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pages 1842–1850. PMLR.

[181] Sanyal, D., Bosch, N., and Paquette, L. (2020). Feature selection metrics: Similarities, differences, and characteristics of the selected models. *International Educational Data Mining Society*.

[182] Schaffer, J. (2015). What not to multiply without necessity. *Australasian Journal of Philosophy*, 93(4):644–664.

[183] Segaran, T. and Hammerbacher, J. (2009). *Beautiful data: the stories behind elegant data solutions*. " O'Reilly Media, Inc.".

[184] Seijo-Pardo, B., Bolón-Canedo, V., and Alonso-Betanzos, A. (2017a). Testing different ensemble configurations for feature selection. *Neural Processing Letters*, 46(3):857–880.

[185] Seijo-Pardo, B., Porto-Díaz, I., Bolón-Canedo, V., and Alonso-Betanzos, A. (2017b). Ensemble feature selection: homogeneous and heterogeneous approaches. *Knowledge-Based Systems*, 118:124–139.

[186] Shannon, C. E. (2001). A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review*, 5(1):3–55.

[187] Shen, Z., Chen, X., and Garibaldi, J. (2018). Performance optimization of a fuzzy entropy based feature selection and classification framework. In *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1361–1367. IEEE.

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

[189] Shen, Z., Chen, X., and Garibaldi, J. M. (2020). A novel meta learning framework for feature selection using data synthesis and fuzzy similarity. In *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–8. IEEE.

[190] Shilbayeh, S. and Vadera, S. (2014). Feature selection in meta learning framework. In *2014 Science and Information Conference*, pages 269–275. IEEE.

[191] Singh, R., Kumar, H., and Singla, R. (2014). Topsis based multi-criteria decision making of feature selection techniques for network traffic dataset. *International Journal of Engineering and Technology*, 5(6):4598–4604.

[192] Smith, J. W., Everhart, J., Dickson, W., Knowler, W., and Johannes, R. (1988). Using the adap learning algorithm to forecast the onset of diabetes mellitus. In *Proceedings of the Annual Symposium on Computer Application in Medical Care*, page 261. American Medical Informatics Association.

[193] Snell, J., Swersky, K., and Zemel, R. (2017). Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30.

[194] Soheili, M., Moghadam, A.-M. E., and Dehghan, M. (2020). Statistical analysis of the performance of rank fusion methods applied to a homogeneous ensemble feature ranking. *Scientific Programming*, 2020.

[195] Somol, P. and Novovičová, J. (2010). Evaluating stability and comparing output of feature selectors that optimize feature subset cardinality. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(11):1921–1939.

[196] Stoppiglia, H., Dreyfus, G., Dubois, R., and Oussar, Y. (2003). Ranking a random feature for variable and feature selection. *Journal of machine learning research*, 3(Mar):1399–1414.

[197] Street, W. N., Wolberg, W. H., and Mangasarian, O. L. (1993). Nuclear feature extraction for breast tumor diagnosis. In *Biomedical image processing and biomedical visualization*, volume 1905, pages 861–870. International Society for Optics and Photonics.

[198] Sun, Q., Liu, Y., Chua, T.-S., and Schiele, B. (2019). Meta-transfer learning for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 403–412.

[199] Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H., and Hospedales, T. M. (2018). Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1199–1208.

[200] Tang, J., Alelyani, S., and Liu, H. (2014). Feature selection for classification: A review. *Data classification: algorithms and applications*, page 37.

[201] Tenenbaum, J., de Silva, V., and Langford, J. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):P.2319–2323.

[202] Thomas, D. R., Pastrana, S., Hutchings, A., Clayton, R., and Beresford, A. R. (2017). Ethical issues in research using datasets of illicit origin. In *Proceedings of the 2017 Internet Measurement Conference*, pages 445–462.

[203] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288.

[204] Tomás, J. T., Spolaôr, N., Cherman, E. A., and Monard, M. C. (2014). A framework to generate synthetic multi-label datasets. *Electronic Notes in Theoretical Computer Science*, 302:155–176.

[205] Turunen, E. and Turunen, E. (1999). *Mathematics behind fuzzy logic*. Physica-Verlag Heidelberg.

[206] Van Der Maaten, L., Postma, E., and Van den Herik, J. (2009). Dimensionality reduction: a comparative. *J Mach Learn Res*, 10(66-71):13.

[207] Vanschoren, J. (2018). Meta-learning: A survey. *arXiv preprint arXiv:1810.03548*.

[208] Vanschoren, J. (2019). Meta-learning. In *Automated Machine Learning*, pages 35–61. Springer, Cham.

[209] Vilalta, R. and Drissi, Y. (2002). A perspective view and survey of meta-learning. *Artificial intelligence review*, 18(2):77–95.

[210] Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al. (2016). Matching networks for one shot learning. *Advances in neural information processing systems*, 29.

[211] Wang, L.-X. (1996). *A course in fuzzy systems and control*. Prentice-Hall, Inc.

[212] Wang, L.-X. and Mendel, J. M. (1992). Generating fuzzy rules by learning from examples. *IEEE Transactions on systems, man, and cybernetics*, 22(6):1414–1427.

[213] Wang, Y.-X., Girshick, R., Hebert, M., and Hariharan, B. (2018). Low-shot learning from imaginary data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7278–7286.

[214] Whitley, D. (1994). A genetic algorithm tutorial. *Statistics and computing*, 4(2):65–85.

[215] Willett, P. (2013). Combination of similarity rankings using data fusion. *Journal of chemical information and modeling*, 53(1):1–10.

[216] Wolberg, W. H. and Mangasarian, O. L. (1990). Multisurface method of pattern separation for medical diagnosis applied to breast cytology. *Proceedings of the national academy of sciences*, 87(23):9193–9196.

[217] Wold, S., Esbensen, K., and Geladi, P. (1987). Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52.

[218] Wolpert, D. H. and Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, 1(1):67–82.

[219] Xin, B., Hu, L., Wang, Y., and Gao, W. (2015). Stable feature selection from brain smri. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

[220] Xing, E. P. (2003). Feature selection in microarray analysis. In *A practical approach to microarray data analysis*, pages 110–131. Springer.

[221] Xu, Z., Huang, G., Weinberger, K. Q., and Zheng, A. X. (2014). Gradient boosted feature selection. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 522–531.

[222] Xue, B., Zhang, M., Browne, W. N., and Yao, X. (2015). A survey on evolutionary computation approaches to feature selection. *IEEE Transactions on Evolutionary Computation*, 20(4):606–626.

[223] Yager, R. R. and RR, Y. (1980). On a general class of fuzzy connectives.

[224] Yang, F. and Mao, K. (2010). Robust feature selection for microarray data based on multicriterion fusion. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 8(4):1080–1092.

[225] Yao, Q., Wang, M., Chen, Y., Dai, W., Li, Y.-F., Tu, W.-W., Yang, Q., and Yu, Y. (2018). Taking human out of learning applications: A survey on automated machine learning. *arXiv preprint arXiv:1810.13306*.

[226] Yu, L. and Liu, H. (2003). Feature selection for high-dimensional data: A fast correlation-based filter solution. In *Proceedings of the 20th international conference on machine learning (ICML-03)*, pages 856–863.

[227] Yu, L. and Liu, H. (2004). Efficient feature selection via analysis of relevance and redundancy. *Journal of machine learning research*, 5(Oct):1205–1224.

[228] Zadeh, L. A. (1965). Fuzzy sets. *Information and control*, 8(3):338–353.

[229] Zadeh, L. A. (1968). Probability measures of fuzzy events. *Journal of mathematical analysis and applications*, 23(2):421–427.

[230] Zadeh, L. A. (1971). Similarity relations and fuzzy orderings. *Information sciences*, 3(2):177–200.

[231] Zaffalon, M. and Hutter, M. (2002). Robust feature selection by mutual information distributions. In *Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence*, pages 577–584. Morgan Kaufmann Publishers Inc.

[232] Zar, J. H. (2005). Spearman rank correlation. *Encyclopedia of Biostatistics*, 7.

[233] Zhang, D., Chen, S., and Zhou, Z.-H. (2008). Constraint score: A new filter method for feature selection with pairwise constraints. *Pattern Recognition*, 41(5):1440–1451.

[234] Zhang, R., Che, T., Ghahramani, Z., Bengio, Y., and Song, Y. (2018). Metagan: An adversarial approach to few-shot learning. *Advances in neural information processing systems*, 31.

[235] Zhao, Z. and Liu, H. (2007). Spectral feature selection for supervised and unsupervised learning. In *Proceedings of the 24th international conference on Machine learning*, pages 1151–1157.

[236] Zhao, Z., Wang, J., Sharma, S., Agarwal, N., Liu, H., and Chang, Y. (2010). An integrative approach to identifying biologically relevant genes. In *Proceedings of the 2010 SIAM International Conference on Data Mining*, pages 838–849. SIAM.

[237] Zheng, L., Chao, F., Mac Parthaláin, N., Zhang, D., and Shen, Q. (2021). Feature grouping and selection: A graph-based approach. *Information Sciences*, 546:1256–1272.

[238] Zhou, Q., Ding, J., Ning, Y., Luo, L., and Li, T. (2014). Stable feature selection with ensembles of multi-relieff. In *Natural Computation (ICNC), 2014 10th International Conference on*, pages 742–747. IEEE.

[239] Zhou, Z.-H. (2009). Ensemble learning. *Encyclopedia of biometrics*, 1:270–273.

[240] Zhou, Z.-H. (2012). *Ensemble methods: foundations and algorithms*. CRC press.

[241] Zhu, X. and Wu, X. (2004). Class noise vs. attribute noise: A quantitative study. *Artificial intelligence review*, 22(3):177–210.

[242] Zoph, B., Vasudevan, V., Shlens, J., and Le, Q. V. (2018). Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710.