# Granular Computing Approach for Intelligent Classifier Design

By

Mohammed Al-Shammaa

A Thesis Submitted in Partial Fulfilment of the
Requirements for the Degree of
Doctor of Philosophy

Department of Electronic and Computer Engineering
College of Engineering, Design and Physical Sciences
Brunel University London

September 2016

# ABSTRACT

Granular computing facilitates dealing with information by providing a theoretical framework to deal with information as granules at different levels of granularity (different levels of specificity/abstraction). It aims to provide an abstract explainable description of the data by forming granules that represent the features or the underlying structure of corresponding subsets of the data.

In this thesis, a granular computing approach to the design of intelligent classification systems is proposed. The proposed approach is employed for different classification systems to investigate its efficiency. Fuzzy inference systems, neural networks, neuro-fuzzy systems and classifier ensembles are considered to evaluate the efficiency of the proposed approach. Each of the considered systems is designed using the proposed approach and classification performance is evaluated and compared to that of the standard system.

The proposed approach is based on constructing information granules from data at multiple levels of granularity. The granulation process is performed using a modified fuzzy c-means algorithm that takes classification problem into account. Clustering is followed by a coarsening process that involves merging small clusters into large ones to form a lower granularity level. The resulted granules are used to build each of the considered binary classifiers in different settings and approaches.

Granules produced by the proposed granulation method are used to build a fuzzy classifier for each granulation level or set of levels. The performance of the classifiers is evaluated using real life data sets and measured by two classification performance measures: accuracy and area under receiver operating characteristic curve. Experimental results show that fuzzy systems constructed using the proposed method achieved better classification performance.

In addition, the proposed approach is used for the design of neural network classifiers. Resulted granules from one or more granulation levels are used to train the classifiers at different levels of specificity/abstraction. Using this approach, the classification problem is broken down into the modelling of classification rules represented by the information granules resulting in more interpretable system. Experimental results show that neural network classifiers trained using the proposed approach have better classification performance for most of the data sets. In a similar manner, the proposed approach is used for the training of neuro-fuzzy systems resulting in similar improvement in classification performance.

Lastly, neural networks built using the proposed approach are used to construct a classifier ensemble. Information granules are used to generate and train the base classifiers. The final ensemble output is produced by a weighted sum combiner. Based on the experimental results, the proposed approach has improved the classification performance of the base classifiers for most of the data sets. Furthermore, a genetic algorithm is used to determine the combiner weights automatically.

To my wife and daughter,

Zahraa and Zainab.

# ACKNOWLEDGEMENTS

Firstly, I would like to express my sincere gratitude to my supervisor Dr Maysam F. Abbod for the continuous support of my Ph.D study and related research, for his patience and motivation. His guidance helped me in all the time of research and writing of this thesis.

I would also like to thank my sponsors, the Higher Committee for Education Development in Iraq (HCED), for their continuous help and support.

# DECLARATION

I declare that this thesis is my own work and is submitted for the first time to the Post-Graduate Research Office. The study was originated, composed and reviewed by myself and my supervisors in the Department of Electronic and Computer Engineering, College of Engineering, Design and Physical Sciences, Brunel University London UK. All the information derived from other works has been properly referenced and acknowledged.

SIGNED: .................................................. DATE: ........................................

# TABLE OF CONTENTS

# LIST OF TABLES

**TABLE** **Page**

# LIST OF FIGURES

**FIGURE** **Page**

# LIST OF ABBREVIATIONS

| | |
|---|---|
| **ANFIS** | Adaptive-Network-based Fuzzy Inference Systems. |
| **AUC** | Area Under Curve. |
| **CI** | Computational Intelligence. |
| **DBSCAN** | Density-Based Spatial Clustering of Applications with Noise. |
| **FCM** | Fuzzy C-Mean. |
| **FCMGr** | FCM-based Granulation. |
| **FN** | False Negative. |
| **FP** | False Positive. |
| **GrC** | Granular Computing. |
| **GrCNNC** | GrC approach for NN Classifier design. |
| **LDA** | Linear Discriminant Analysis. |
| **NBC** | Naive Bayes Classifier. |
| **QDA** | Quadratic Discriminant Analysis. |
| **RBF** | Radial Basis Function. |
| **ROC** | Receiver Operating Characteristic. |
| **STD** | Standard Deviation. |
| **TN** | True Negative. |
| **TP** | True Positive. |

# CHAPTER 1

# INTRODUCTION

RESEARCH background, motivation, aim and objectives are presented in this chapter. Research background is presented in Section 1.1 while Section 1.2 contains the motivation of this thesis. The aim and objectives of this thesis are stated in Section 1.3 and its organisation is described in Section 1.4. Publications are listed in Section 1.5

## 1.1 Background

Classification is a supervised learning method that aims to group or cluster objects into predefined groups (classes) based on certain criteria. It has a wide range of applications as in financial forecast [1, 2, 3], medical diagnosis [4, 5, 6], fault diagnosis [7, 8], image classification [9, 10] and text classification [11, 12]. In most applications, neither the classification process can be described algorithmically nor the relationship between input and output can be derived analytically. Therefore, in the paradigm of machine learning, classification is usually achieved by learning the relationship that exists between a set of input (feature) variables and an output variable (target class label). The problem of classification can be stated as follows [13]:

"Given a set of training data points along with associated training labels, determine the class label for an unlabeled test instance."

Thus, classification algorithms are generally comprised of two phases [13, 14]:

- Training Phase: In this phase, the construction of a classification model from the training data set is performed. The data set consists of sample input data with a preassigned class labels. A machine learning method is used to extract knowledge from the labelled training data. One important desirable characteristic of classification systems is their ability to interpret the data structures underlying the classification model (e.g. the decision rules) [15].

- Testing Phase: In this phase, the constructed model is used to classify new testing data relying on knowledge extracted in the training stage.

The main objective of designing a classifier is to achieve the highest classification accuracy (the lowest error rate) in the testing phase. In other words, achieving highest possible level of generalisation, i.e. the classifier's ability to generalise the classification process learnt from the training phase to classify new data in the testing phase [16].

The task of classification is addressed by numerous methods such decision trees, rule-based methods, neural networks, Support Vector Machine (SVM) methods, and nearest neighbour methods. There are two types of outputs a classification algorithm may result in [13]:

- Discrete Label: A class label is assigned to the input instance.

- Numerical Score: Numerical scores are computed to indicate the degree to which the input instance belong to each class. A discrete label for an input instance can be produced from its numerical scores by choosing the class with the highest score.

In terms of the number of classes, there are two types of classification: binary classification (grouping data into only two classes), and multi-class classification (classifying the data into more than two classes). In this thesis, only binary classification is considered since any multi-class classification task can be divided into multiple binary classification tasks. This can be achieved by classifying data into one class against all other classes [17].

## 1.2 Research Motivation

Granular Computing (GrC) is a general computation theory that imitates human thinking and reasoning by dealing with information as a form of aggregates called information granules. An information granule is a collection or group of elements (usually data points) that are linked together due to the fact that they share some degree of similarity, functionality or indistinguishablity [18]. The process of forming information granules, information granulation, underlays most of the activities that involve information processing, from human thinking and problem solving to artificial intelligence and digital signal processing [18, 19]. Information granulation facilitates dealing with information by providing a theoretical framework to deal with information at different levels of granularity, that is different levels of specificity/abstraction [18].

A lot of research on GrC in data classification has been carried out in various settings and diverse applications. Due to its broad range of coverage in many different frameworks and its wide spectrum of application fields, GrC research comes in numerous approaches, objectives and points of view.

Addressing the problem of data classification, different GrC studies have been conducted in different frameworks such as: cluster analysis, set theory [20], fuzzy theory [20, 21], graph theory [15, 22, 23] and rough set [20, 21, 24, 25]. Concepts of GrC have been realised in various methodologies including data clustering, image segmentation [23], fuzzy lattice [26, 27], rough logic [20, 21, 24, 25], information tables [28] and graphs [15, 22, 23].

One of the common technique in the GrC research in data classification is forming larger information granules from smaller ones. This is achieved by different methods and under different names such as: dilation [26, 27], coarsening or merging [29]. These methods are usually based on finding some relation between granules depending on the framework and approach. Examples of relations between granules are: spatial relations [23], distance measures [29], rough inclusion and fuzzy inclusion.

On the other hand, dividing large granules into smaller ones is adopted by some studies [30].

GrC-based methods have been employed in achieving common tasks of data classification. Two of the vital tasks are the extraction of classification rules from data [15, 20, 25, 28, 30, 31] and the identification of classification systems [24].

Studies of GrC in data classification are aimed at various applications such as, medical data classification [25], text classification [30, 32, 33], image classification [15, 21, 23] and spam detection [21].

## 1.3 Aim and Objectives

The main aim of this thesis is to develop a GrC approach to the design of intelligent classification systems and investigate its efficiency. Different classification systems are considered to investigate the efficiency of the proposed approach. In particular, the following classification systems are considered: Fuzzy inference system (FIS), Neural Network (NN), neuro-fuzzy system and classifier ensemble.

Based on the review of concepts, methods and applications of GrC, and to achieve the aforementioned aim, the following objectives have been identified:

1. Develop a clustering algorithm to construct granules that represent the data for building classification systems.
2. Evaluate the effectiveness of the developed clustering algorithm in forming information granules at various levels of granularity for classification problem.
3. Investigate how the resulted granules can be used in building classifiers of different types.
4. Evaluate the performance of different classifiers built using the proposed method.

## 1.4 Thesis Organisation

This thesis consists of six chapters and is organised as follows:

- Chapter 2 introduces the fundamentals, principles, methods and applications of GrC. A summary of the common frameworks of GrC is provided in Section 2.2. Section 2.3 reviews the synergy between GrC and Computational Intelligence (CI) with emphasis being laid on NNs, fuzzy systems and evolutionary algorithms. Applications of GrC in data clustering and classification are reviewed in Section 2.4.
- In chapter 3, a method of clustering-based information granulation for the problem of data classification is presented. The proposed method is used for the identification of fuzzy classifiers. The proposed method is described in Section 3.3. The implementation of the proposed method and experimental setup and results are provided in Section 3.4. Section 3.5 is dedicated to the application of the proposed method in fuzzy system identification and the comparison of its performance to standard Fuzzy C-Mean (FCM) based fuzzy systems.
- A GrC approach for the design of NN classifiers is proposed in Chapter 4. The approach is applied to NNs and Adaptive-Network-based Fuzzy Inference Systems (ANFIS). The proposed approach is described in Section 4.2 and the results of its application on NNs are presented in Section 4.3. In Section 4.4, an experimental comparison of classifiers constructed using the proposed approach and other well-known classifiers is provided.
- In Chapter 5, a GrC approach to classifier ensemble is proposed. The proposed method is described in Section 5.2. Experimental results of the evaluation of the proposed approach are reported in Section 5.3. Section 5.4 investigates the use of a genetic algorithm for the weights assignment of the ensemble.
- Conclusions and future work are presented in Chapter 6.

## 1.5 Publications

- M. Al-Shammaa and M.-F. Abbod, "Automatic generation of fuzzy classification rules from data," in Proceedings of the International Conference on Neural Networks-Fuzzy Systems (NN-FS '14), pp. 74-80, Venice, Italy, 2014

- M. Al-Shammaa and M. F. Abbod, "Automatic generation of fuzzy classification rules using granulation-based adaptive clustering," Systems Conference (SysCon), 2015 9th Annual IEEE International, Vancouver, BC, 2015, pp. 653-659.
  (Best Student Paper Award - 2$^{nd}$ place)

- M. Al-Shammaa and M. F. Abbod, "Granular computing approach for the design of medical data classification systems," Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), 2015 IEEE Conference on, Niagara Falls, ON, 2015, pp. 1-7.
  (Best Student Paper Award)

CHAPTER 2

# Granular Computing

This chapter introduces the fundamentals, principles, methods and applications of Granular Computing (GrC). Section 2.1 provides a brief introduction to the fundamentals and principles of GrC. Section 2.2 summarises the common frameworks of GrC. Section 2.3 reviews the synergy between GrC and Computational Intelligence (CI) with emphasis being laid on Neural Networks (NNs), fuzzy systems and evolutionary algorithms. Applications of GrC in data clustering and classification are reviewed in Section 2.4. Conclusions of this chapter are given in Section 2.5.

## 2.1 Introduction

While the concept of GrC was first introduced by Zadeh in 1979 [34], the term GrC was first used in 1997 [35] and since then it drew the interest of many researchers and witnessed a rapid development and research growth. Although the term was recently brought to live, the concepts of GrC are believed to be applied throughout human history, as it is a natural methodology for humans to deal with daily life issues. The ideas and principles underlying GrC have been covered in many different fields under different names, such as divide and conquer, structured programming, interval analysis, cluster analysis, etc. [36].

GrC is a general computation theory that imitates human thinking and reasoning by dealing with information as a form of aggregates called *information granules*. A granule is defined in Merriam Webster's Dictionary as: "a small particle; especially: one of numerous particles forming a larger unit" [37]. An information granule is a collection or group of elements (usually data points) that are linked together due to the fact that they share some degree of similarity, functionality or indistinguishablity [18]. The process of forming information granules (namely, *information granulation*) underlays, in a way or another, most of the activities that involve information processing, from human thinking and problem solving to artificial intelligence and digital signal processing [18, 19]. For example, in image processing, an image can be processed as a group of some features (lines, regions, etc.) which in turn can be dealt with as sets of individual pixels. GrC aims to provide an abstract explainable description of the data by forming granules that represent, with respect to a certain viewpoint, the features or the underlying structure of corresponding subsets of the data.

In order to solve a problem, it may be desirable in some cases to split it into smaller more achievable modules, or, in other cases, one may need to make generalisations (abstraction) of the problem, away from the burden of its details, in order to get a better comprehension. Either way, information granulation facilitates dealing with information by providing a theoretical framework to deal with information at different *levels of granularity*, that is different levels of specificity/abstraction [18]. Granularity is represented by the size, number and distribution of information granules. The larger is a granule, the lower is its granularity and the higher is its abstraction (generality) [38].

The process of forming granules is based on the *relationships* that exist among them, namely: *interrelationships* and *intrarelationships*. Interrelationships reflect the extent to which the elements within a granule are related, while intrarelationships describe the relation among different granules (in the same level). Granules are formed either by combining granules of higher granularity (usually smaller

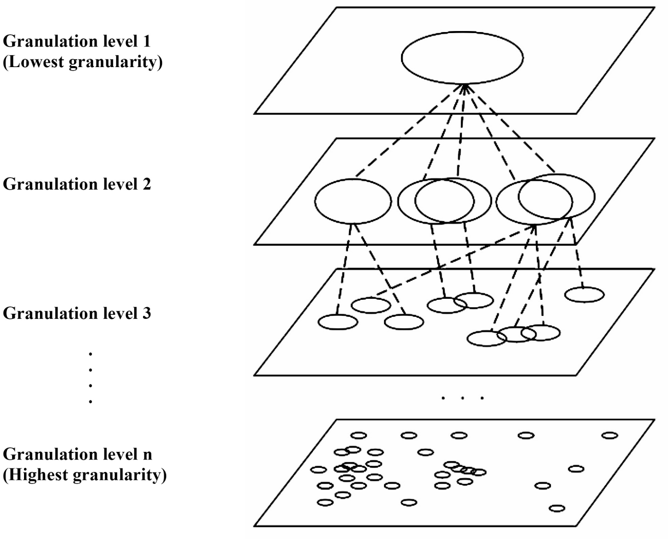

FIGURE 2.1. Granules and granulation levels [39].

in size) into granules of lower granularity (usually larger in size), or by dividing granules of lower granularity into granules of higher granularity. The former way of constructing granules can be represented by the intrarelationship of *coarsening*, i.e. producing coarser granules (granules with lower granularity), while the latter can be represented by the interrelationship of *refinement*, i.e. producing finer granules (granules with higher granularity) [19]. Figure 2.1 illustrates the relationships among granules and process of granulation.

GrC attempts to establish a unifying theory that integrates multiple theories, methodologies, techniques, and tools that adopt a granulation approach for solving complex problems [40]. Due to the application-dependent nature of the formulation and interpretation of information granules, different terminology may be used for description at different levels. GrC is a multidisciplinary study and can be viewed from different perspectives. For example, Y.Y. Yao [41] presented the "granular computing triangle" in which GrC can be studied from three different perspectives: philosophical perspective, methodological perspective and computational perspective. From the first perspective, GrC depicts a paradigm for the structuring and extraction of human thinking, while from the second perspective GrC deals with methods and techniques for structured problem solving. In the computational perspective, GrC represents an approach for information processing in the abstract level, in the brain, and in machines. The common factor of the three perspectives is the main feature of GrC: representing information at multiple levels of granularity.

While numeric computing is *data-oriented*, GrC is *knowledge-oriented*, thus, it imposes many challenges and open questions. Some of the questions that still need to be answered are: How to precisely define "Granular Computing"? How to formalise the framework of GrC and the process of constructing granules? How to measure the level of granularity and how to define the "size" of a granule? [18, 40]

## 2.2 Granular Computing Frameworks

The process of information granulation has been one of the key components of several models and frameworks that are already developed in various domains such as: system modelling, image processing, pattern recognition, and data compression. Some of these frameworks are: interval computation, rough sets, fuzzy sets and cluster analysis. This section highlights some of the attempts to information granulation within each framework.

Methods of each framework have different approaches and perspectives on information granulation. In fuzzy sets, for example, information granulation is performed by the means of fuzzification using fuzzy membership function. While, in another example, hierarchical clustering performs information granulation by clustering data based on their proximity or distance either by merging smaller clusters or by dividing larger ones. Although these examples are very different in their approaches, nature and perspectives, they are both considered methods of information granulation from the perspective of the unifying theory of GrC.

### 2.2.1 Interval Analysis

Interval analysis is an active field of research and application that addresses the problem of ambiguity that results from the uncertainty of numerical computation. In interval analysis, a variable is dealt with as an interval number, represented by an ordered pair of real numbers (endpoints), instead of a single value [42]. All elements of an interval can be represented by a single label that stands for the interval, hence they become completely indistinguishable [43]. This way of dealing with a set of values (interval) as an entity that embraces them all indistinguishably is one way of performing information granulation from the viewpoint of GrC.

Several studies have adopted the approach of interval analysis to information granulation. For example, a formal framework for information granulation is introduced in [44]. The proposed framework, which is named distributed intervals, is based on interval analysis theory and the criteria of coherency and proximity. The framework was derived by combining the concepts and characteristics of distributed entities and intervals. Benefiting from the two paradigms of distributed entities and intervals, the distributed interval frameworks results in a rich model for granulation.

A type of granular classifiers, named hyperbox-driven classifiers, is developed in [45]. The classifier relies on the interaction of interval analysis and fuzzy sets. The development of classifiers was achieved from the viewpoint of information granulation where a class of patterns represents an information granule in the feature space.

Other examples of studies addressing the approach of interval analysis to information granulation can be found in [46, 47]. In the former, the concept of knowledge granulation based on the maximal consistent block in interval valued information systems was proposed. While the latter developed multilevel and multi-view granular structures using various inclusion relations and operations on interval sets.

### 2.2.2 Fuzzy Sets

While interval theory addresses the the uncertainty that results from numerical computation, fuzzy set theory addresses the uncertainty that results from the transitional nature of entities [42], i.e. the gradual evaluation of the membership of different elements in a set. Fuzzy set theory is based on the vital concept of partial membership where the degree by which an object belongs to a set is represented by a membership function that takes a value between 0 and 1. Thus, from the viewpoint of GrC, fuzzy sets represent a naturally suited formalism to model information granulation [48, 49]. Figure 2.2 shows how granules are represented by fuzzy sets for a 2-dimensional illustrative data.

A hierarchy of granules is formed by an algorithm based on fuzzy concept lattices [50]. The concept of knowledge granularity presents in the hierarchy of fuzzy concept lattice. This hierarchy consists of multiple levels of nodes where each node represents a granule. Nodes at higher levels represent coarser granules while nodes at lower levels represent finer granules. The coarsest granule (top node) is divided into a number of children nodes (finer granules). Children nodes are further divided into finer granules until the finest granule (bottom node) is reached. Thus, a node includes all elements of its children nodes and the top node consists of all elements from the universe of the fuzzy concept lattice.

Another approach for the modelling of information granules is the formation of fuzzy descriptors as information granules [51]. Fuzzy descriptors are the main elements of fuzzy descriptive models that aim to provide with constructs that describe experimental data at a general level of relationships. An interactive approach is adopted for the formation of the information granules. In this interactive approach, a structure in a data set is visualised so that it can be examined by a designer who chooses from some visualized regions according to its level of data homogeneity. The visualization process is implemented using self-organizing maps.

Fuzzy relations are used alongside with spectral clustering to construct word granules that represent text in order to improve knowledge discovery [52]. As a result of this technique, topics in a text or a set of documents are represented by word granules containing words of sufficient significance. The technique captures the relationships that exist among related words by combining them into granules.

Information granules are formed as a representation of numeric membership functions of fuzzy sets [53]. As a result of the granulation of the membership functions, a set of finite number of information granules is produced to provide an abstract view at the membership concept instead of numeric membership values.

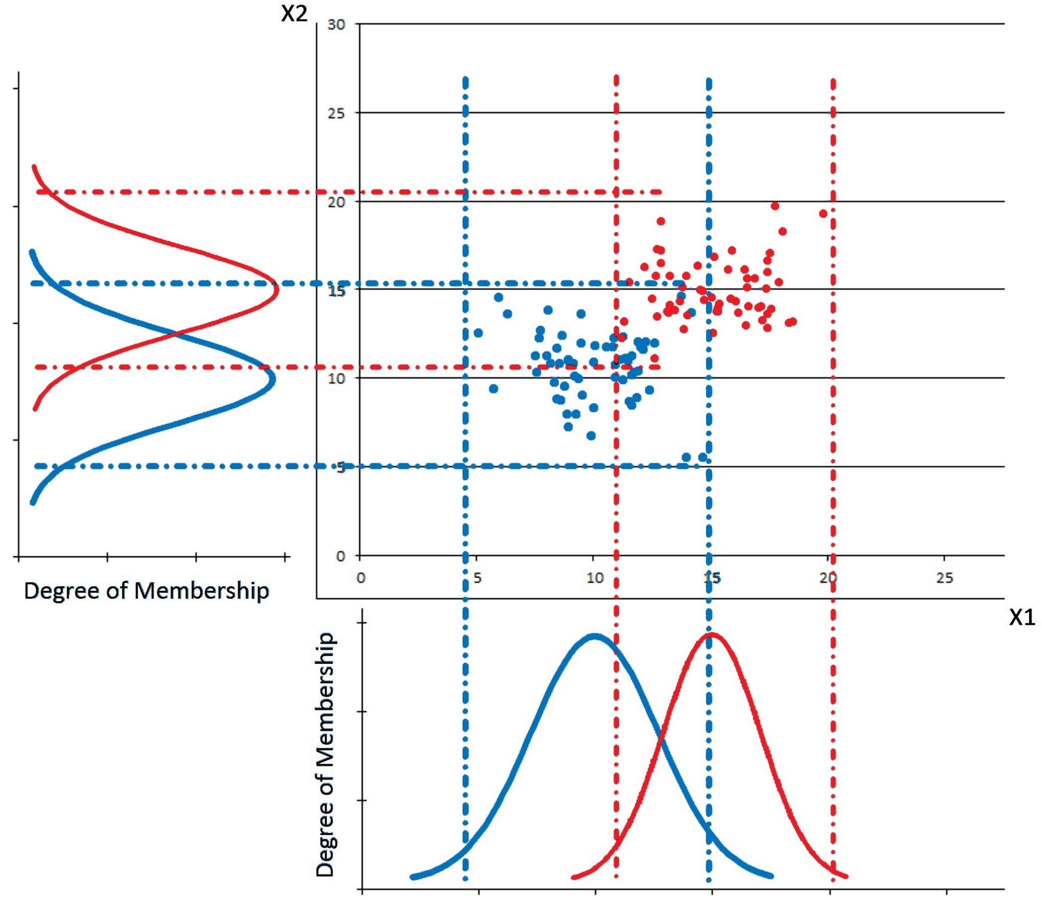

FIGURE 2.2. Granule representation by fuzzy sets.

### 2.2.3 Rough Sets

First described by Pawlak in 1982 [54], rough set theory is one of the main methods of GrC modelling. One research survey study showed that 39% of the surveyed articles (113 articles) belongs to the rough set framework [55].

Similar to interval analysis and fuzzy sets theory, rough set theory addresses vagueness and uncertainty. While fuzzy set theory uses fuzzy membership to address gradualness of knowledge, rough set theory addresses granularity of knowledge through the indiscernibility relation. The indiscernibility relation reflects the inablity to recognise some objects due to the lack of sufficient knowledge. Therefore, instead of dealing with single objects, reasoning is limited to dealing with granules of indiscernible objects which represent the amount of knowledge perceivable due to the indiscernibility relation [56]. Thus, objects with identical attributes (indiscernible objects) are aggregated into blocks (equivalence class) that represent elementary granules of knowledge [57].

Rough sets are defined by approximation or rough membership function. In rough set theory, a set S is characterized by three factors: lower approximation, upper approximation and boundary region, as shown in Figure 2.3. The lower approximation L of S is the union of all sets that are subsets of S. In other words, the lower approximation is the set of all objects that are certainly classified as S. The upper approximation U is the union of all sets that have a non-empty intersection with S, i.e. the set of all objects that are possibly classified as S. The boundary region of S is the set-theoretic difference of U and L, that is the set of all objects that are possibly but not certainly classified as S. A rough set is a set that has a non-empty boundary region. On the other hand, a set with empty boundary region is crisp [57]. Unlike interval analysis, rough set approximations are approximation of known sets in an approximation space defined by an underlying indiscernibility relation [58]. In contrast to membership function in the fuzzy set theory, rough set theory does not require prior knowledge about the system [59, 60].

Alternatively, rough sets can be defined by rough membership functions that express conditional probability that an object belongs to S given an indiscernibility relation (as illustrated in Figure 2.4) [61].

**Rough inclusion for information granulation**. Based on the concept of rough membership function, rough inclusion can be defined as the partial inclusion of a rough set in a rough set (i.e. partial inclusion of an information granule in an information granule) and reads "an object x is a part of an object y to degree at least r" [62, 63]. Therefore, several attempts have been made to model information granulation based on rough inclusion [63, 64, 65, 66].

**Multi-granulation rough sets**. In rough set theory, the equivalence classes produce a partition of the universe where an object belongs to a single class only. The inflexible nature of the binary equivalence relation imposes limitations on the

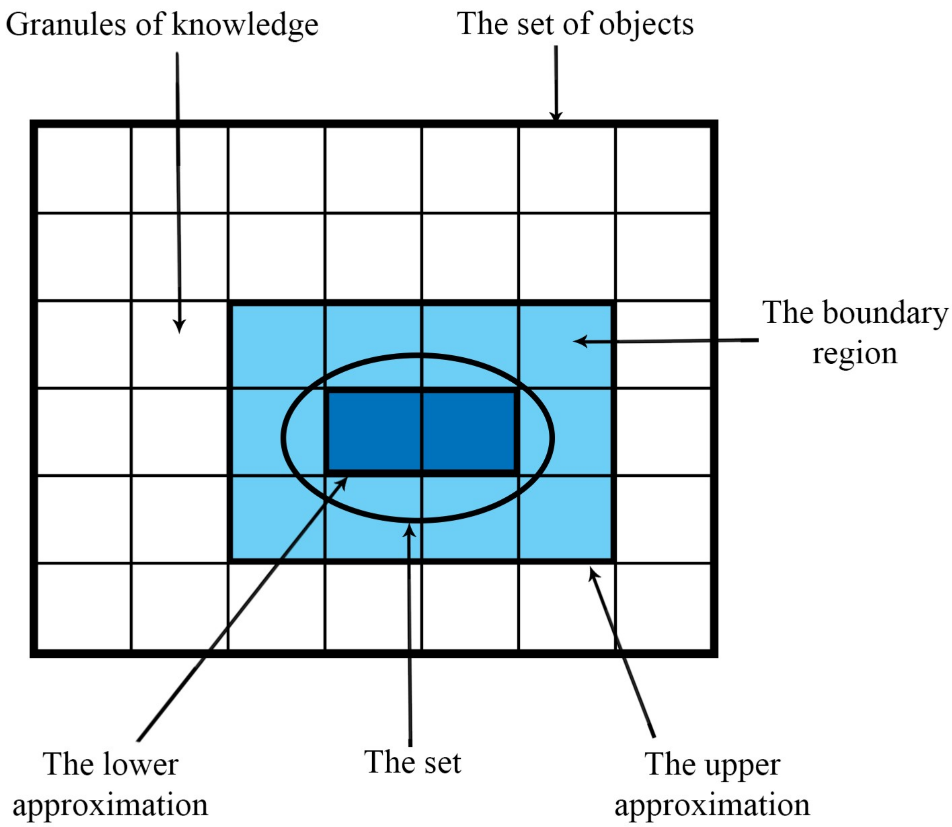

FIGURE 2.3. Rough set approximation [57].

development and application of rough sets as it results in single (fixed) granulation. To overcome these limitations, some extensions of rough set are proposed. One of the extensions of rough set is multi-granulation rough sets which employ multiple binary relations to describe a concept [67, 68].

**Variable precision rough sets**. Another extension of rough set is variable precision rough sets which uses partial classification to classify objects into equivalence classes. As a result, there is an admissible error, within a predefined boundary, in the inclusion relation. Based on variable precision rough sets, information multi-granulation can be achieved [69, 70, 71, 72].

**Covering-based rough sets**. Covering-based rough sets is another extension of rough set that is based on a covering instead of a partition of the universe. Like

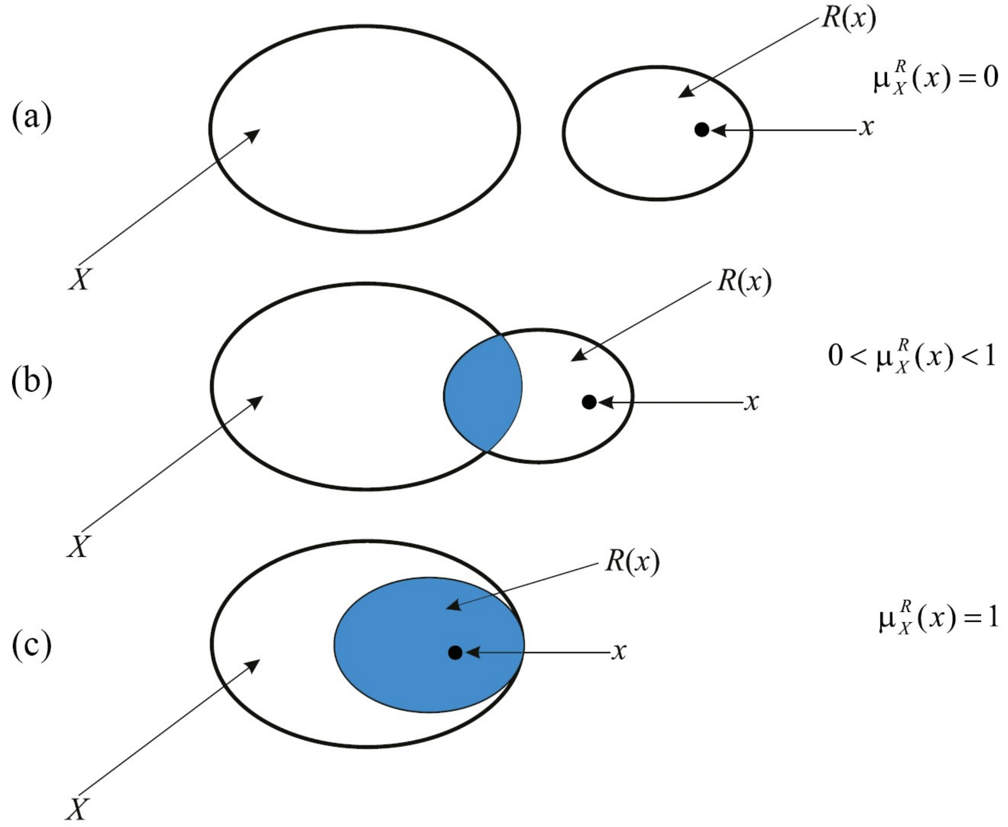

FIGURE 2.4. Rough membership function $\mu_X^R(x)$ of the element x to the set X given the equivalence class R [57].
(a) $\mu_X^R(x) = 0$. (b) $0 < \mu_X^R(x) < 1$. (c) $\mu_X^R(x) = 1$.

other rough set extensions, covering-based rough sets provide a framework for multi-granulation of information [73, 74, 75].

### 2.2.4 Cluster Analysis

Clustering is the process of grouping a set of objects into a number of groups (clusters) so that objects belonging to the same group have a higher level of similarity to each other than to those in other groups [76]. It plays a major role in some research fields, such as pattern recognition, data mining, machine learning, etc. Clustering is a useful tool for identifying the underlying relationship and structure of data. Clustering is a competent candidate for modelling information granulation due to the fact that it has a far less time and space complexity compared to other models of information granulation [77].

**Constructing information granules**. Various clustering algorithms are used to form information granules such as hierarchical clustering, K-means clustering and Fuzzy C-Mean (FCM) clustering. In [78], either hierarchical clustering or K-means clustering is used in the first phase of the proposed method to form information granules. In the second phase, a parametric refinement of information granules is performed using the principle of justifiable granularity. Thus, the proposed method combines the unsupervised (clustering) and supervised (justifiable granularity) approaches.

**Data-driven information granulation**. As an unsupervised approach, clustering allows the construction of information granules in a data-driven environment. Such environment is discussed in [79] for fuzzy clustering algorithms highlighting their advantage of producing fuzzy granules that can be described linguistically.

**Refinement of coarser information granules**. Clustering is also used in the refinement of coarser information granules. The process of forming finer granules from coarser ones is discussed in [80] where the refinement process is addressed as an optimisation task in which a general partition requirement has to be satisfied. In this study, the conditional FCM clustering algorithm is used.

**A multi-level view of information granulation**. Clustering is an efficient method for the extraction of information granules in a multi-level view of information granulation which plays an important role in problem solving. From the viewpoint of GrC, granularity level is determined by cluster size (or total number of clusters). Bigger clusters (fewer clusters) indicate coarser granularity level, while smaller clusters (more clusters) represent finer granularity level. Different granularity levels can be achieved by varying the similarity threshold used by the clustering method which results in varying the size and number of clusters.

One approach based on double clustering framework is proposed in [81] to address the extraction of fuzzy information granules from numerical data. The proposed approach, which is called multi-level double clustering, is a double-clustering approach where conditional FCM clustering algorithm is used in the multi-dimensional clustering and hierarchical clustering algorithm is used in the one-dimensional clustering.

**Dynamic data granulation**. Clustering is used to identify structures that present in data in a dynamic manner. In [82], an algorithm is proposed for the dynamic data granulation by performing conditional FCM clustering for each data snapshots. The constructed granules evolves dynamically from one data snapshot to another by adjusting the number of clusters depending on the structure and complexity of patterns in each data snapshot.

## 2.3 Granular Computing and Computational Intelligence

CI is a sub-field of artificial intelligence that is concerned with adaptive techniques that aim to enable or facilitate intelligent behaviour in complex and changing systems. The paradigms of CI are charactrised by their ability to learn, generalise, discover and adapt to new situations [83]. Some of CI paradigms are: artificial neural networks, evolutionary computation, swarm intelligence, artificial immune systems, and fuzzy systems.

One shortcoming of CI techniques is the lack of efficient knowledge acquisition, representation and retrieval structures. Thus, there is a demanding need to develop more efficient tools for the representation and retrieval of knowledge [84]. As a framework for knowledge acquisition and representation, GrC serves as a paradigm to overcome this shortcoming. There have been many studies that incorporate GrC with various CI techniques. In this section, the integration of GrC methods with fuzzy systems and NNs is discussed.

### 2.3.1 Fuzzy Systems

Generally, the integration of GrC methods in fuzzy system design aims to provide more interpretable fuzzy rule base. As a result, complexity of the fuzzy system is reduced. In such granular fuzzy systems, antecedents of the fuzzy rules are represented by granular extensions of the original fuzzy sets [85]. Particle swarm optimization is used to find the optimal distribution of information granulation in order to achieve a balance between the complexity and performance of the constructed granular fuzzy rule-based system.

GrC methods are used to describe data clusters automatically. A cluster is described by a fuzzy IF-THEN rule providing knowledge that is more understandable by human. In this case, each sample is given a fuzzy description then clustering is guided by selected exemplar fuzzy descriptions [86].

Using GrC approaches, the rules of rule-based fuzzy systems are automatically generated. Rough sets constructed by a genetic algorithm are used to generate the fuzzy rules. Fuzzy rules generated using rough sets have a reduced number of antecedent terms and a high coverage rate [87].

### 2.3.2 Neural Networks

NNs are a powerful tool for solving non-linear mapping problems due to their strong fault tolerance, self-organization and massive parallel processing. NNs are widely used in many fields of science and many areas of application [88]. However, due to the distributed style of NNs, they are regarded as black boxes, i.e. they are difficult to interpret. In addition, traditional NNs have some disadvantages like long training time and the high computational complexity when dealing with high dimensional problems. One way to address the problem of computational complexity is to decompose the problem into a set of simpler and easier to handle subtasks. GrC is a suitable framework to achieve this modular approach.

Due to the architecture of traditional NNs, they are primarily used to process numeric data. However, there are many cases where processing of non-numeric data (e.g. linguistic data) is desired. Incorporating concepts and methods of GrC allows NNs to deal with data of low granularity (i.e. non-numeric data) as information granules [89]. Based on the granularity of data and architecture of NNs, four classes of NNs [43]:

- Networks where both input data and structures (connections) are of high granularity. Standard numerically-driven NNs and their training methods belong to this class. Generalisation in networks of this class is represented by their ability to predict outputs of new numeric entries.

- Networks trained using numeric data (data of high granularity) but used with input data of low granularity (granulated input). Networks of this class have to be able to process non-numeric data (data of low granularity), usually through a preprocessing input layer. Outputs of theses networks tend to be of low granularity and their generalization includes the ability to deal with inputs of varying granularity.
- Networks with granular connections (architecture of low granularity ) but used with input data of high granularity. The outputs of such networks may be of low granularity as a result of their granular connections.
- Networks where both input data and structures (connections) are of low granularity.

The fusion of GrC and NNs results in Granular Neural Networks (GNNs) that belong to the last three classes of the above classes. The development of GNNs comprises of two key stages [43]:

- The first stage is information granulation where information granules are formed from the numeric data.
- Then, NNs are trained by the information granules resulted from the first stage. That is, the NN does not deal with the numeric data, instead, it deals with information granules of lower granularity.

One approach to construct a GNN is to build a NN based on fuzzy IF ... THEN rules [90]. Each rule is represented by a hidden layer node and the weights of connections are fuzzy sets of the rule instead of numerical values. Fuzzy sets used by the fuzzy rules represent knowledge that can be extracted from the NN architecture. An illustrative example is shown in Figure 2.5.

Fuzzy rule based GNNs are used in many applications like performing land use classification of satellite images [90, 91]. A fuzzy rule based evolving GNN is used for the modelling of evolving fuzzy system from fuzzy data streams of non-stationary environments [92]. In this system, granular fuzzy models are constructed using trapezoidal membership function representation.

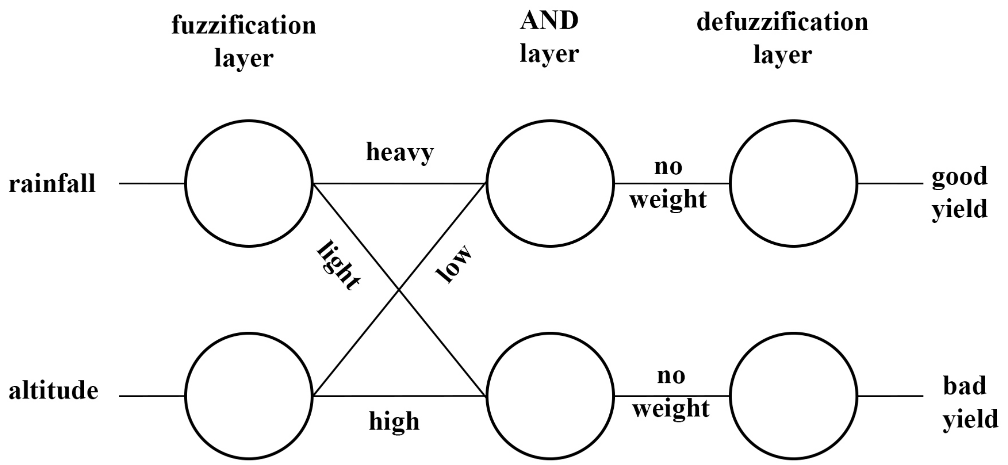

FIGURE 2.5. An illustrative example of a fuzzy NN built from two fuzzy rules [90]: IF **rainfall** is *heavy* AND **altitude** is *low* THEN **yield** is *good*, IF **rainfall** is *light* AND **altitude** is *high* THEN **yield** is *bad*

A NN whose connection weights are represented by fuzzy sets (linguistic terms) is used to determine the natural granularity of a dataset [93]. The GNN is based on multilayer perceptron architecture and the back-propagation learning algorithm with momentum. Linguistic arithmetic operations based on fuzzy sets are used in training the granular connection weights.

Another method to the realisations of GNNs is the interval-valued NN. In the approach of interval-valued NNs, the connections of standard (numeric) NNs are augmented by granular connections (intervals) formed from granulating a data set [89].

In addition to fuzzy and interval approaches, many studies that aim to develop GNNs adopt approaches based on rough sets. For example, a NN is developed from a set of rough rules extracted from data in a framework named rough rule granular extreme learning machine [94]. The rough rules are extracted by a process of data reduction using the algorithms of attributes reduction and attributes values reduction in rough set theory.

Hybrid methods that involve the combination of rough set and fuzzy set are employed in the development of GNNs. A fuzzy rough GNN that is based on multilayer perceptron and back-propagation algorithm is used for pattern classification [95, 96]. Initial weights of the GNN are derived from knowledge extracted from data using fuzzy rough set theoretic techniques. Input and target vectors are represented by fuzzy granules and membership values.

## 2.4 Granular Computing Applications

GrC has a wide spectrum of applications like decision making [97, 98, 99], pattern recognition [100, 101, 102], image segmentation [103, 104, 105] and data mining [106, 107, 108]. In this section, applications of GrC in data clustering and classification are presented.

### 2.4.1 Data Clustering

As discussed in Section 2.2.4, cluster analysis is one of many fields that provide frameworks for the concepts and approaches of GrC. Conventional clustering methods are unsupervised, i.e. there is no prior knowledge about the training data set nor the desired clustering output. However, some clustering methods, namely semi-supervised clustering methods, make use of information or constraints on the clusters or desired output for a subset of the training data set [109]. While both unsupervised and semi-supervised clustering methods aim to group data into clusters, GrC aims to use the resulted granules (clusters) as information entities that represent the data at a certain level of abstraction. On the other hand, the approach and concepts of GrC are widely adopted for the task of clustering. For example, GrC concepts in fuzzy sets and rough sets are employed to develop soft clustering methods.

Conventional clustering uses hard partitioning that assign each object into only one cluster. However, for many practical applications, there is incompleteness and uncertainty in data. Clustering based on GrC allows for soft partitioning. That is, allowing partial inclusion and overlap of clusters. In addition, the multi-level granularity approach of GrC facilitates processing high-dimensional data or providing abstract (general) view of finely-detailed data. Other advantages of granularity clustering include enabling the integration of different clustering methods and reducing processing time and storage [77]. Clustering based on two main frameworks of GrC, namely fuzzy granular clustering and rough granular clustering, are addressed here.

#### 2.4.1.1 Fuzzy Granular Clustering

Fuzzy clustering methods are frequently used for granular clustering. One of the widely used fuzzy clustering algorithms is FCM [110, 111, 112, 113]. For example, FCM is used to create meaningful profiles of a social network of phone users. The profiles are created by repeated applications of FCM to form granules that are described by fuzzy cluster memberships [112]. Clustering guided by domain knowledge is achieved by using FCM. The domain knowledge is represented by information granules that portray user's point of view at the data [113]. In this case, information granules represent the viewpoints that serve as prototypes for the clustering process.

A fuzzy clustering algorithm called multi-step maxmin and merging algorithm is used alongside with a clustering validity measure called granularity-dissimilarity to find prototypes with optimal granularity [114]. The validity measure uses fuzzy membership functions to calculate granularity. The algorithm involves merging the worst granule (the granule with largest granularity and least dissimilarity) into other granules until granulation criteria is reached.

Fuzzy granular gravitational clustering algorithm is based on Newton's law of universal gravitation and GrC [115]. In this algorithm, two user-set variables control the granularity of the final resulted clusters. Clusters with low gravitational density are merged into clusters with high gravitational density if the distance between them is less than the radius, i.e. the first user-set variable. The algorithm continues iteratively until no distance between two clusters is shorter than the user set radius. The radius is changed each iteration according to the second user set variable to control the number and size of clusters. The radius of influence of each cluster is represented by a fuzzy membership function.

A rapid fuzzy rule clustering algorithm based on granular computing is used to describe clusters [86]. The algorithm comprises of an unsupervised feature selection method and data granulation. Data granulation is based on exemplar fuzzy descriptions selected from descriptions created by the feature selection method for all samples. A single fuzzy rule is extracted from each granule to make it understandable for humans.

#### 2.4.1.2 Rough Granular Clustering

As a framework of GrC, rough sets are used in many studies in data clustering [116, 117, 118, 119]. The rough inclusion and rough membership function allow for an object to partially belong to more than one cluster.

One application of rough granular clustering is document clustering and its application in search engine technology and the problem of Web search result clustering [120]. The approach is based on tolerance rough set that is used to provide approximation of concepts in documents and to enhance the vector representation of text snippets. As a result, clusters of documents of similar concepts are formed followed by cluster labelling that is derived from tolerance classes.

Another application of rough granular clustering is the development of ordered ranking of objects [117]. Daily price patterns are grouped based on volatility using an ensemble of two rough ordered clustering scheme.

### 2.4.2 Classification

Classification is a supervised learning method that aims to group or cluster objects into predefined groups (classes) based on certain criteria. One important desirable characteristic of classification systems is their ability to interpret the data structures underlying the classification model (e.g. the decision rules). That is, the system should be able to extract knowledge related to the classification task regardless of data complexity [15]. In this regard, GrC is an appropriate approach that allow for the automatic extraction and representation of knowledge, hence, there has been many studies that have adopted this approach to address classification problem.

GrC approach to classification is adopted in various applications. For example, GrC helps in improving the classification of photographic images [15] and remote sensing images [90, 121]. GrC provides an effective framework for text classification and categorisation [32, 33] and pattern classification [122, 123]. In addition, relational data classification is addressed through GrC approach [31]. Another application of GrC in classification is medical and biomedical data classification [124, 125, 126]. Numerical data classification [127] and financial data classification [128] are also addressed through GrC.

In general, a classification system based on GrC is realised in two stages:

- First, information granulation is performed by partitioning, covering or other methods. The resulted granules provide a knowledge base that describes features, semantics and/or structures of data.

- Based on information granules, the next stage involves categorising data into predefined classes according to some measure of similarity.

The construction of information granules is achieved by any of granule modelling methods. Data clustering algorithms are widely used to construct information granules [22, 33, 127, 129]. Another method for the formation of granules that is used in image classification problems is image segmentation [15, 121]. Also, a method that relies on fuzzy inclusion relation to perform granulation is presented in [26]. Divide-and-conquer principle is used in [124] to form granules through partitioning.

The next stage involves a classification technique that is based on machine learning, computational intelligence, statistical analysis, etc. A set of rules that is induced from the granules constructed in the first stage, is used to perform the classification [28, 31, 121]. Alternatively, fuzzy sets are employed for the classification task [129]. SVMs are frequently used for classification in this stage too [124, 125, 126].

GrC helps solving the imbalanced data challenge where most of the data samples are labelled as one class and far few samples are labelled as the other class. With imbalanced data, traditional machine learning algorithms usually achieve high accuracy for the majority class but poor accuracy for the minority class. A method based on GrC is proposed in [130] to solve this problem. The method, called "knowledge acquisition via information granulation", uses information granulation to eliminate some unnecessary details by representing data by granules at a selected level of granularity. As a result, the minority examples in imbalanced learning tasks are better identified. The level of granularity is determined using two proposed indices: the homogeneity index (H-index) and the undistinguishable ratio (U-ratio).

## 2.5 Conclusions

Fundamentals, principles, methods and applications of GrC were introduced in this chapter. GrC is a general computation theory that imitates human thinking and reasoning by dealing with information as a form of aggregates (i.e. information granules). An information granule is a collection or group of elements (usually data points) that are linked together due to the fact that they share some degree of similarity, functionality or indistinguishablity. GrC facilitates dealing with information by providing a theoretical framework to deal with information at different levels of granularity, that is different levels of specificity/abstraction.

The process of information granulation has been one of the key components of several models and frameworks that are developed in various domains. Some of these frameworks are: interval computation, rough sets, fuzzy sets and cluster analysis. One of the main aims of interval analysis, fuzzy sets theory and rough set theory is to address vagueness and uncertainty. Interval theory addresses the the uncertainty that results from numerical computation by approximation (interval number). While fuzzy set theory uses fuzzy membership to address gradualness of knowledge, rough set theory addresses granularity of knowledge through rough approximation and rough membership (based on indiscernibility relation).

As a framework for knowledge acquisition and representation, GrC serves as a paradigm to overcome some shortcomings of CI, specifically, the lack of efficient knowledge acquisition of CI techniques. There have been many studies that incorporate GrC with various CI techniques in order to develop more efficient tools for the representation and retrieval of knowledge. GrC methods are employed in fuzzy system design to provide more interpretable fuzzy rule base and reduce its complexity.

GrC is a suitable framework to achieve a modular approach to address the problem of computational complexity of NNs by decomposing the problem into a set of simpler and easier to handle subtasks. Incorporating concepts and methods of

GrC allows NNs to deal with data of low granularity (i.e. non-numeric data) that exist in many applications.

GrC has a wide spectrum of applications like decision making, pattern recognition, image segmentation and data mining. Applications of GrC in data clustering and classification were presented in this chapter. Due to the intrinsic relationship between clustering and GrC, the approach and concepts of GrC are widely adopted for the task of clustering. From the viewpoint of GrC, granularity level is determined by cluster size (or total number of clusters). Different granularity levels can be achieved by varying the similarity threshold used by the clustering method which results in varying the size and number of clusters. Clustering based on frameworks of GrC, such as fuzzy sets and rough sets, allows for soft partitioning (partial inclusion and overlap of clusters). In addition, the multi-level granularity approach of GrC facilitates processing high-dimensional data or providing abstract (general) view of finely-detailed data.

GrC is an appropriate approach that allow for the automatic extraction and representation of knowledge, hence, there has been many studies that have adopted this approach to address classification problem. GrC provides classification systems with the ability to interpret the data structures underlying the classification model (e.g. the decision rules).

CHAPTER 3

# Clustering-Based Granulation for Data Classification

A METHOD of clustering-based information granulation for the problem of data classification is presented in this chapter. The proposed method is used for the identification of fuzzy classifiers. Section 3.1 provides a brief introduction followed by Section 3.2 that introduces the FCM method used in the proposed method. The proposed method is described in Section 3.3. The implementation of the proposed method and experimental setup and results are provided in Section 3.4. Section 3.5 is dedicated to the application of the proposed method in fuzzy system identification and the comparison of its performance to standard FCM-based fuzzy systems. Conclusions of this chapter are provided in Section 3.6.

## 3.1 Introduction

Data clustering is an unsupervised learning method that aims to group or cluster objects into groups (clusters) based on the similarities among the feature variables. Data clustering is one of the most widely studied in the data mining and machine learning paradigms. It has a wide range of applications such as handwriting recognition [131], document clustering [132], gene expression analysis [133, 134, 135] and content-based image retrieval [136, 137].

The basic problem of clustering may be stated as follows [138]

“Given a set of data points, partition them into a set of groups which are as similar as possible.”

Clustering provide a concise model of the data which can be interpreted in the sense of either a summary or a generative model. There is a wide spectrum of data clustering algorithms depending on data type and the nature of of the problem. Some of the widely used clustering algorithms are: Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [139], k-means algorithm [140] (centre-based algorithm), the global k-means algorithm [141] (search-based algorithm) and FCM algorithm [142, 143].

The task of data clustering is similar to data classification in the sense that it segments the input (feature) space into segments. However, in classification, segmentation is based on knowledge of the structure of the class groups that is extracted from the training data set in the training phase while in clustering the segmentation is based on the similarities among the feature variables without any knowledge of the structure of the class groups [14]. Data clustering serves as an intermediate step in many types of classification methods, such as rule-based clustering methods where the resulted clustering is used to construct classification rules.

In this chapter, a method of information granulation is proposed to produce information granules from training data set at multiple levels of granularity. The granulation process is performed using a modified FCM clustering method followed by a coarsening process, i.e. merging the clusters of higher granularity level into larger clusters of lower granularity level. The resulted granules are to be used to build binary classifiers of various types using various methods. In particular, the identification of fuzzy classifier systems using the granulated data is investigated in this chapter.

## 3.2 Fuzzy C-means Clustering Method

### 3.2.1 Soft Clustering

In terms of how clusters are formed, there are two main types of clustering algorithms: hierarchical and partitioning clustering. While in hierarchical clustering input data are clustered in a nested hierarchy way, partitioning clustering uses objective function to segment the input space into a number of clusters.

A clustering algorithm may assign an object to one and only one cluster. Such clustering algorithm is referred to as hard clustering algorithm where the partitioning is described by the matrix [144]:

$$U = \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ u_{21} & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ u_{c1} & u_{c2} & \cdots & u_{cn} \end{pmatrix} \tag{3.1}$$

where $n$ is the number of objects in the data set and $c$ is the number of clusters. In hard clustering, the partitioning matrix U has the following properties [144]:

- The association of an object to a cluster is either 1 or 0 (the object either belongs to the cluster or not):

$$u_{ji} \in \{0,1\}, \quad 1 \le j \le c, 1 \le i \le n \tag{3.2}$$

- An object belongs to only one cluster:

$$\sum_{j=1}^{c} u_{ji} = 1, \quad 1 \le i \le n \tag{3.3}$$

- A cluster is associated with at least one object:

$$\sum_{i=1}^{n} u_{ji} > 0, \quad 1 \le j \le c \tag{3.4}$$

In many cases, it is necessary for the clustering algorithm to be able to express uncertainty in the association of objects to clusters. That is, the clustering algorithm is required to allow for the overlapping of clusters. To meet this requirement, soft clustering is used. In soft clustering, an object can belong to more than one cluster with partial inclusion, namely, a fuzzy membership function as illustrated in Figure 3.1. In soft clustering, the partitioning matrix U has the following properties [144]:

- The association of an object to a cluster is a fuzzy membership function that takes a value between 0 and 1 (the object $i$ belongs to the cluster $j$ by a degree of $u_{ji}$):

$$u_{ji} \in [0,1], \quad 1 \leq j \leq c, 1 \leq i \leq n \tag{3.5}$$

- The sum of degree of association of an object to all the clusters is equal to 1:

$$\sum_{j=1}^{c} u_{ji} = 1, \quad 1 \leq i \leq n \tag{3.6}$$

- A cluster is associated with at least one object by a degree greater than 0:

$$\sum_{i=1}^{n} u_{ji} > 0, \quad 1 \leq j \leq c \tag{3.7}$$

### 3.2.2 Fuzzy C-means Algorithm

FCM is one of the most popular clustering methods that has been used in a broad spectrum of applications and has provided efficient solutions for several problems [145]. FCM was developed by Dunn [142] and was then improved by Bezdek [143]. FCM results in a fuzzy partitioning of the input space into C clusters depending on their means (centroid), hence the name.

The FCM algorithm attempts to find a solution for the minimisation of the objective function [146]:

$$J_m(U,V) = \sum_{i=1}^{n} \sum_{j=1}^{c} u_{ji}^{m} d^2(x_i, v_j) \tag{3.8}$$

where:

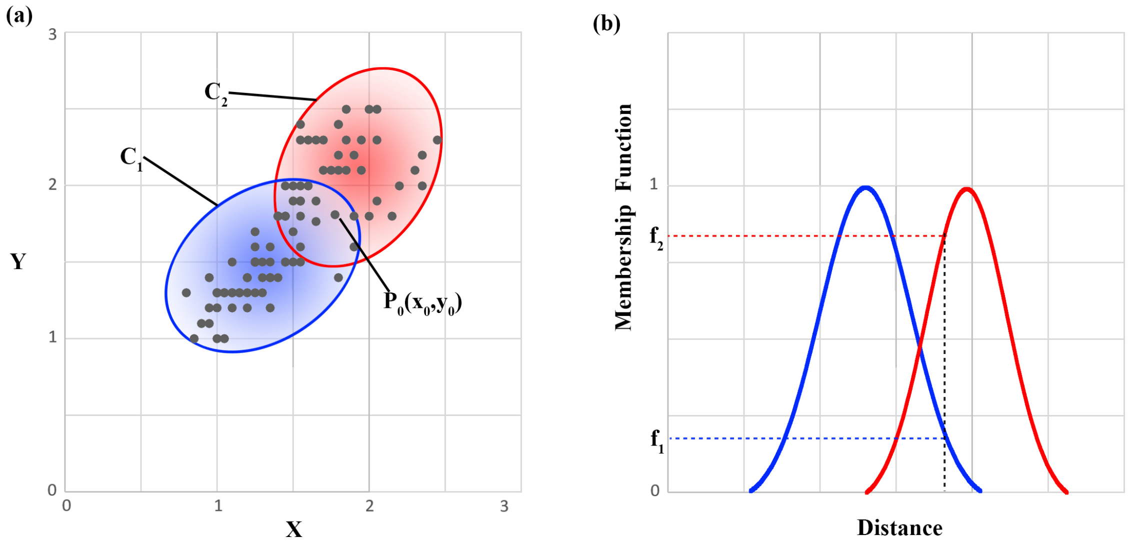

FIGURE 3.1. Illustrative example of fuzzy clustering. (a) Overlapping clusters. (b) Fuzzy membership function of a sample point $\mathrm{P}_0$

$X = \{x_1, x_2 \dots x_n\}$ is the data,

$c$ is the number of clusters; $2 \leq c < n$,

$m$ is a weighting exponent that controls the "fuzziness" of the resulting clusters; $1 \leq m < \infty$,

$U$ is the fuzzy c-partition of $X$,

$V = \{v_1, v_2 \dots v_c\}$ is the vector of clusters centres,

$d(.,.)$ is an inner product metric, e.g. Euclidean distance.

The FCM algorithm can be formalised by the following iterative procedure:

1. Starting with fixed $c$ and $m$, choose an initial (usually, random) matrix $U^0$.

2. For step $k, k = 1, 2, \dots Itr_{max}$, perform the following:

3. Compute new clusters centres $V^k, j = 1, 2, \dots c$ using the equation:

$$v_j = \frac{\sum_{i=1}^{n} u_{ji}^m x_i}{\sum_{i=1}^{n} u_{ji}^m} \tag{3.9}$$

4. Compute the updated fuzzy c-partition matrix $U^k$ using the equation:

$$u_{ji} = \frac{1}{\sum_{k=1}^{c} \left( \frac{d(x_i, v_j)}{d(x_i, v_k)} \right)^{\frac{2}{m-1}}} \tag{3.10}$$

5. If $k = 1$ return to step 2. Otherwise, compare $U^k$ to $U^{k-1}$. If $\|U^k - U^{k-1}\| < \epsilon$ stop. Otherwise, set $U^{k-1} = U^k$ and return to step 2.

The output of the algorithm is the fuzzy c-partition matrix $U^k$ and the clusters centres $V^k$ . An alternative termination criteria is to compare the value of the objective function $J_m^k$ to $J_m^k - 1$ [147], i.e. the last step of the above algorithm is replaced by:

5. Compute $J_m^k(U^k, V^k)$ using equation 3.8.
6. If $k = 1$ return to step 2. Otherwise, compare $J_m^k$ to $J_m^{k-1}$. If $|J_m^k - J_m^{k-1}| < \epsilon$ stop. Otherwise, set $U^{k-1} = U^k$, $J_m^{k-1} = J_m^k$, and return to step 2.

## 3.3 Granulation using FCM

Information granulation is the process of forming information granules at a certain level of granularity. The main aim of information granulation is to facilitate dealing with information by providing a theoretical framework to deal with information at different levels of granularity. One of the frameworks of forming information granules is data clustering. In this section, a method of granulation using a modified FCM algorithm, namely FCM-based Granulation (FCMGr), is proposed for the task of binary data classification. FCMGr uses an iterative sequence of a modified FCM data clustering followed by a set of cluster merging steps to generate granules for the lower granularity level in each iteration.

### 3.3.1 The proposed FCM clustering algorithm

To use the FCM algorithm for generating data granules for binary data classification, it is desirable to differentiate data points belonging to different classes. In general, there are two approaches to achieve this differentiation: clustering data belonging to each class separately or including class labels as an additional input which increases the distance between two data points belonging to different classes.

While the second approach increases the dimensionality of the input data (hence increases the computational time of the clustering process), it has the advantage of reducing the effect of outliers points on the size of the clusters, especially in regions of overlap of classes in the input space of high dimensional data at lower granularity levels. This is illustrated in Figure 3.2 where two-dimensional two-class data (Figure 3.2-a) are clustered by the two different approaches. Figure 3.2-c shows the clusters resulted from clustering data points belonging to each class separately while Figure 3.2-d shows the clusters resulted from clustering data with class labels included as an additional input as shown in Figure 3.2-b. The outlier point at (5,2) (blue circle) has a greater influence on clusters of the same class in the first approach than the second approach. This influence is manifested by the wider clusters in the first approach (Figure 3.2-c) resulting in more overlap between clusters of different classes which suggest less efficient classification. FCMGr uses the second approach. The data points and cluster centres combined with their corresponding class labels are denoted by $\hat{X}$ and $\hat{V}$ respectively while the data points and cluster centres without class labels are denoted by $X$ and $V$ respectively.

To use FCM in the proposed method of granulation, some modifications to the algorithm are made. Firstly, the order of computing $\hat{U}$ and $\hat{V}$ is reversed. That is, the modified FCM starts by choosing initial clusters centres $\hat{V}^0$ then computes the new fuzzy partition matrix $\hat{U}^k$ using equation 3.10 and the updated clusters centres $\hat{V}^{k+1}$ using equation 3.9. For the first granulation level (the level with the highest granularity), the initial clusters centres $\hat{V}^0$ are chosen to be the points of the input data after adding some small random noise ($\delta$) to avoid singularity problem of making $d(\hat{x}_i, \hat{v}_k) = 0$ in equation 3.10.

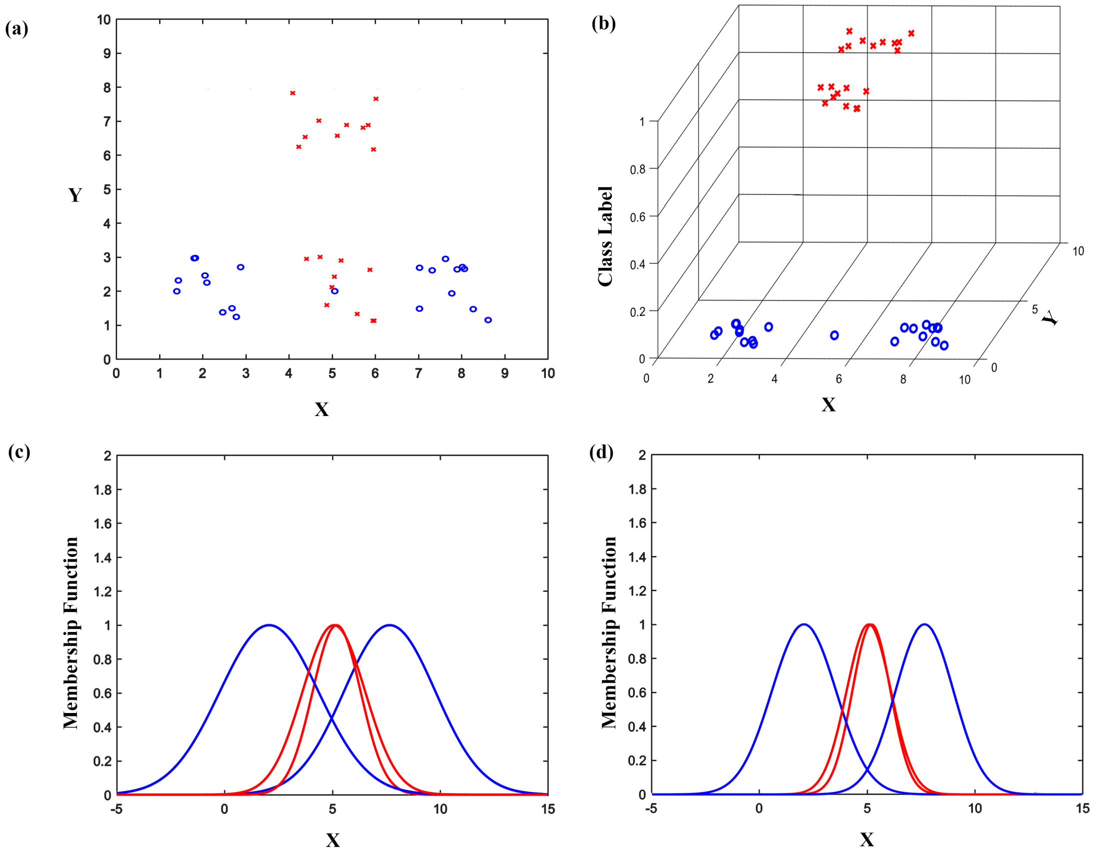

FIGURE 3.2. Illustrative example of labelled data fuzzy clustering. (a) Labelled data. (b) Labelled data with label as input. (c) Clustering each class separately. (d) Clustering labelled data with label as input.

The second modification is that the fuzziness factor $m$ is chosen dynamically for each granularity level rather than being fixed prior the algorithm start. To determine $m$, an objective function is proposed. For each granularity level, $m$ is varied and the value that minimises the objective function is used. The proposed objective function is given by:

$$F_g(m) = A \sum_{j=1}^{c} \sum_{\substack{i=1 \\ C_i = L_j}}^{n} d(x_i, v_j) u_{ji}^m - B \sum_{j=1}^{c} \sum_{\substack{i=1 \\ C_i \neq L_j}}^{n} d(x_i, v_j) u_{ji}^m \tag{3.11}$$

where $c$ is the number of clusters, $C_i$ and $L_j$ are the class labels of $x_i$ and $v_j$ respectively, and **A** and **B** are scaling factors. Initially, the number of clusters $c$ is the number of training samples. Then, for each granularity level, the merging process determines $c$.

The first term of the objective function $F_g$ is the sum of weighted distances of each data point to clusters of similar class label ($D_s$) while the second term is the sum of weighted distances of each data point to clusters of opposite class label ($D_o$). Minimising the objective function $F_g$ with respect to $m$ results in the best trade-off between minimising $D_s$ and maximising $D_o$, i.e. maximising the inclusion of data points to clusters of similar class label and the separation of data points from clusters of opposite class label.

For imbalanced data (where data samples belonging to a class are significantly more than the data samples belonging to the other class), the minimisation of $F_g$ usually leads to the domination of minimising the first term over maximising the second term resulting in wide clusters with high values of inclusion (high values of $u_{ji}$ for $x_i$ and $v_j$ of similar class labels) but with low separation values (high values of $u_{ji}$ for $x_i$ and $v_j$ of opposite class labels). To overcome this issue, the scaling factors $A$ and $B$ are introduced. Let $P_C$ equals the ratio of number of data points of the class with less data samples to the number of data points of the class with more data samples, then $A$ is chosen to equal $1-P_C$ and $B$ equals $P_C$.

It should be noticed that both $d(x_i, v_j)$ and $u_{ji}^m$ are computed for input data and cluster centres without class labels ($X$ and $V$ respectively).

### 3.3.2 Cluster Merging Algorithm

After generating granules at a granularity level using the modified FCM, initial clusters for the generation of granules at a lower granularity level are produced by merging clusters of similar class labels that have weighted distance less than $R_g$ between them. The value $R_g$ controls the granularity of level $g$, the higher its value the lower the granularity of the level. The weighted distance between two merging clusters of similar class is computed using the following proposed equation:

$$D_{jk}(v_j,v_k) = d(v_j,v_k)\left(1+\frac{\sum\limits_{\substack{i=1\\C_i\neq L_j}}^{n} u_{ji}u_{ki} - \sum\limits_{\substack{i=1\\C_i=L_j}}^{n} u_{ji}u_{ki}}{2\sum\limits_{\substack{i=1\\C_i=L_j}}^{n} u_{ji} * \sum\limits_{\substack{i=1\\C_i=L_j}}^{n} u_{ki}}\right) \tag{3.12}$$

where $C_i$ and $L_j$ are the class labels of the data point $x_i$ and cluster $v_j$ respectively.

Equation 3.12 means that the more points that belong to both $v_j$ and $v_k$ the less the weighted distance $D_{jk}$ is and the higher granularity level that they will be merged at. On the other hand, the more points of opposite class that are located between $v_j$ and $v_k$ the more the weighted distance $D_{jk}$ is and the lower granularity level that they will be merged at.

The merging of $v_j$ and $v_k$ is done by updating $v_j$ with the centre of the new cluster using the equation:

$$v_j = \tilde{v}_j + (v_k - \tilde{v}_j) * \frac{\sum\limits_{i=1}^{n} u_{ki}}{\sum\limits_{i=1}^{n} u_{ji}} \tag{3.13}$$

where $\tilde{v}_j$ is the old value of $v_j$.

After updating $v_j$, $\hat{v}_k$ and its corresponding $\hat{u}_{ki}$ values are deleted. The merging process is performed iteratively until no two clusters of the same class ($v_j$, $v_k$) have $D_{jk}(v_j,v_k) < R_g$ for a certain granularity level $g$.

### 3.3.3 Complete proposed algorithm

The complete FCMGr algorithm can be stated as follows:

1. Set the initial clusters centres $\hat{V}^0 = \hat{X} + \delta$ and the granularity level index $R_g = 0.01$.
2. For granularity level $g = 1,2,\ldots$, set the initial fuzziness factor $m = 1.1$ and do steps 3-11.

3. For fuzziness factor $m$, set $F_{MIN} = \infty$ and do steps 4-8.

4. For step $k, k = 1, \ldots Itr_{MAX}$, perform the following:

5. Compute new fuzzy c-partition matrix $\hat{U}^k_{g,m}$ using the equation:

$$\hat{u}_{ji} = \frac{1}{\sum\limits_{h=1}^{c} \left( \frac{d(\hat{x}_i, \hat{v}_j)}{d(\hat{x}_i, \hat{v}_h)} \right)^{\frac{2}{m-1}}} \tag{3.14}$$

6. Compute the updated clusters centres $\hat{V}^k_{g,m}$ using the equation:

$$\hat{v}_j = \frac{\sum\limits_{i=1}^{n} \hat{u}^m_{ji} \hat{x}_i}{\sum\limits_{i=1}^{n} \hat{u}^m_{ji}} \tag{3.15}$$

7. Compute $J^k_{g,m}(\hat{U}^k, \hat{V}^k)$ using the equation:

$$J^k_{g,m}(\hat{U}^k, \hat{V}^k) = \sum_{i=1}^{n} \sum_{j=1}^{c} \hat{u}^m_{ji} d^2(\hat{x}_i, \hat{v}_j) \tag{3.16}$$

8. If $k = 1$ return to step 4. Otherwise, compare $J^k_{g,m}$ to $J^{k-1}_{g,m}$. If $|J^k_{g,m} - J^{k-1}_{g,m}| < \epsilon$ continue to next step. Otherwise, set $\hat{U}^{k-1} = \hat{U}^k$, $J^{k-1}_{g,m} = J^k_{g,m}$, and return to step 4.

9. Compute $F_g(m)$ using equation 3.11.

10. If $F_g(m) < F_{min}$ set $F_{min} = F_g(m)$, $\hat{U}^{best}_g = \hat{U}^k_{g,m}, \hat{V}^{best}_g = \hat{V}^k_{g,m}, m_{best} = m$.

11. If $m < m_{max}$ increase $m$ by $\Delta m$ and return to step 3. Otherwise, continue to the next step.

12. Compute $U_g$ for input data and cluster centres without class labels ($X$ and $V$ respectively) using equation 3.10 after setting $m = m_{best}$.

13. For each cluster $v_j$ in $V^{best}_g$ do steps 14-16.

14. For each cluster $v_k$ in $V^{best}_g, k \neq j, L_k = L_j$ do steps 15-16.

15. Compute $D_{jk}(v_j, v_k)$ using equation 3.12.
16. If $D_{jk}(v_j, v_k) < R_g$ merge $v_j$ and $v_k$ by setting $\tilde{v}_j = v_j$ and using equation 3.13 then deleting $\hat{v}_k$ and its corresponding $\hat{u}_{ki}$ values.
17. If the number of clusters $N_C$ is more than two, increase $R_g$ by $\Delta R$ and return to step 2. Otherwise, stop.

The output of the algorithm is a set of granulation levels each of which consists of a set of clusters centres $\hat{V}_g^{best}$ and a fuzzy partition matrix $\hat{U}_g^{best}$. If no clusters are merged at a certain $R_g$ value, the resulted granulation level is deleted so that there are only distinct granulation levels in the output. Figure 3.3 shows the flowchart of the FCMGr algorithm.

## 3.4 Implementation

### 3.4.1 Data Sets

To test and evaluate the performance of FCMGr, benchmark data sets are used in the rest of this chapter and throughout the next chapters. All of the data sets except the last two are available from the UCI Machine Learning Repository [148]. The data sets of the UCI Machine Learning Repository are widely used in classification, regression and decision making research [149, 150, 151]. The selected data sets are all associated with the task of binary classification and have various dimensionality (number of attributes), size (number of samples) and class distribution. Table 3.1 summarises these properties of the selected data sets. To use these data sets in training and testing classifier performance, the data are partitioned into training and testing data, 80% and 20% respectively. Choosing different part of the data for testing results in different training/testing partitioning of the data. Since the testing data partition is 20% of the data, there are 5 training/testing partitionings (5-fold cross validation): P1, P2 ... P5. Following is the description of each of the selected data sets:

1. Pima Indian Diabetes Data Set: This data set contains records of Pima Indian patients from the United States tested for diabetes. All patients are females

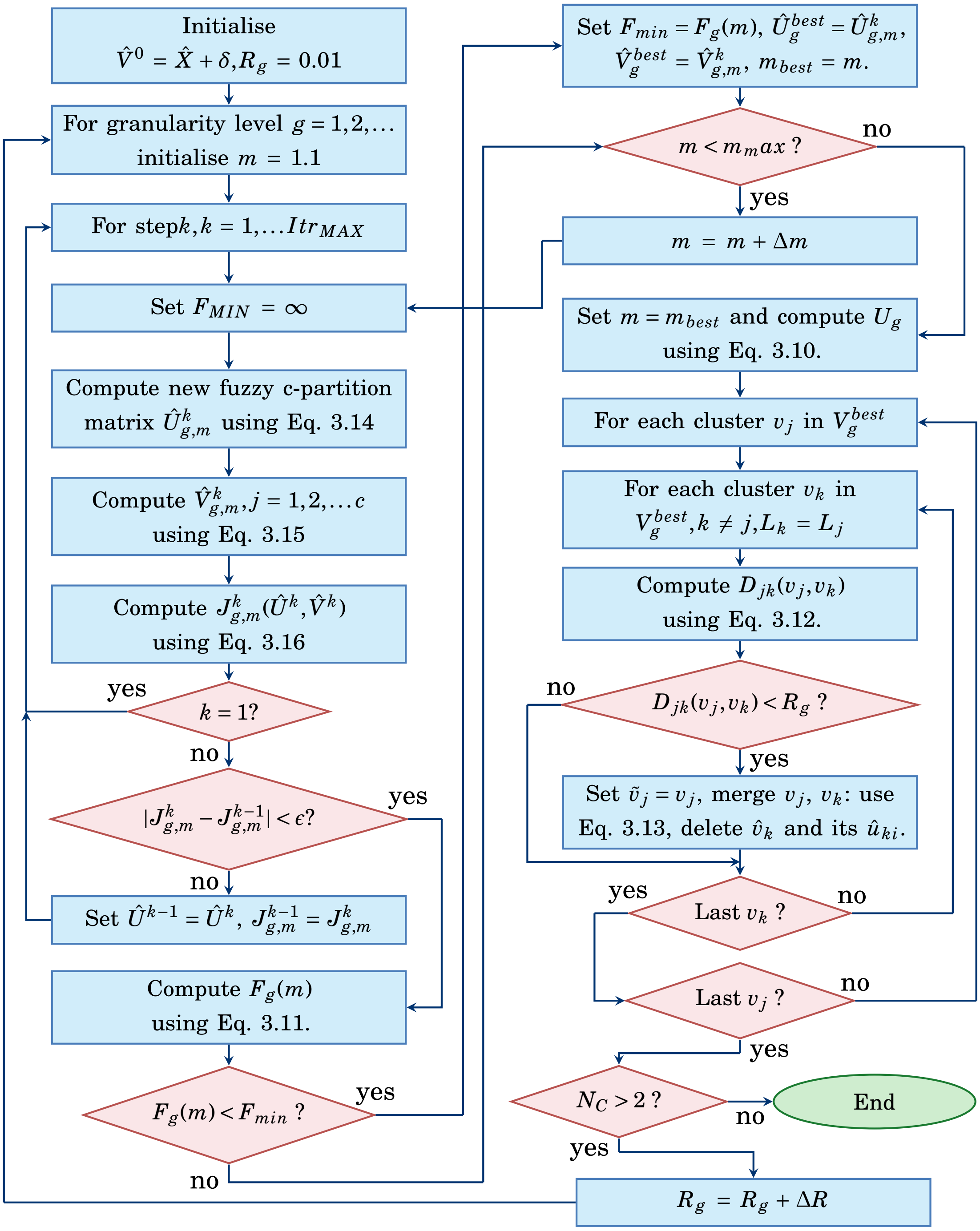
Initialise $\hat{V}^0 = \hat{X} + \delta, R_g = 0.01$
For granularity level $g = 1,2,\ldots$ initialise $m = 1.1$
For step$k, k = 1,\ldots Itr_{MAX}$
Set $F_{MIN} = \infty$
Compute new fuzzy c-partition matrix $\hat{U}^k_{g,m}$ using Eq. 3.14
Compute $\hat{V}^k_{g,m}, j = 1,2,\ldots c$ using Eq. 3.15
Compute $J^k_{g,m}(\hat{U}^k, \hat{V}^k)$ using Eq. 3.16
$k = 1?$
yes
no
$|J^k_{g,m} - J^{k-1}_{g,m}| < \epsilon?$
yes
no
Set $\hat{U}^{k-1} = \hat{U}^k$, $J^{k-1}_{g,m} = J^k_{g,m}$
Compute $F_g(m)$ using Eq. 3.11.
$F_g(m) < F_{min}$ ?
yes
no
Set $F_{min} = F_g(m)$, $\hat{U}^{best}_g = \hat{U}^k_{g,m}$, $\hat{V}^{best}_g = \hat{V}^k_{g,m}$, $m_{best} = m$.
$m < m_{max}$ ?
no
yes
$m = m + \Delta m$
Set $m = m_{best}$ and compute $U_g$ using Eq. 3.10.
For each cluster $v_j$ in $V^{best}_g$
For each cluster $v_k$ in $V^{best}_g, k \neq j, L_k = L_j$
Compute $D_{jk}(v_j, v_k)$ using Eq. 3.12.
$D_{jk}(v_j, v_k) < R_g$ ?
no
yes
Set $\tilde{v}_j = v_j$, merge $v_j$, $v_k$: use Eq. 3.13, delete $\hat{v}_k$ and its $\hat{u}_{ki}$.
Last $v_k$ ?
yes
no
Last $v_j$ ?
no
yes
$N_C > 2$ ?
no
End
yes
$R_g = R_g + \Delta R$

FIGURE 3.3. Flowchart of the FCMGr algorithm.

at least 21 years old. The attributes of the data set are: number of times pregnant, plasma glucose concentration (a 2 hours in an oral glucose tolerance test), diastolic blood pressure (mm Hg), triceps skin fold thickness (mm), 2-hour serum insulin (mu U/ml), body mass index (weight in kg / (height in m)$^2$), diabetes pedigree function, age (years), class variable (0 or 1).

2. BUPA Liver Disorders Data Set: This data set contains records of liver patients from USA. The first 5 attributes are all blood test results which are likely to be related to liver disorders that might result from excessive alcohol consumption. Each data sample constitutes the record of a single male individual. The attributes of the data set are: MCV (mean corpuscular volume), Alkphos (alkaline phosphotase), SGPT (serum glutamic-pyruvic transaminase, alamine aminotransferase), SGOT (serum glutamic-oxaloacetic transaminase, aspartate aminotransferase), GAMMAGT (gamma-glutamyl transpeptidase), drinks (number of half-pint equivalents of alcoholic beverages drunk per day), selector field (class labels).

3. ILPD (Indian Liver Patient Dataset): ILPD contains records of liver patient and non liver patient. This data set contains 441 male patient records and 142 female patient records collected from north east of Andhra Pradesh, India. Selector is a class label used to divide into groups(liver patient or not). The attributes of the data set are: age of the patient, gender of the patient, total Bilirubin, direct Bilirubin, Alkaline Phosphotase, SGPT (Alamine Aminotrans-

TABLE 3.1. Properties of the selected data sets [148].

| Data set | No. of samples | No. of attributes | Class distribution |
|---|---|---|---|
| Pima Indian | 768 | 9 | 268/500 (1/0) |
| BUPA liver disorders | 345 | 7 | 145/200 (0/1) |
| ILPD | 583 | 11 | 167/416 (0/1) |
| Wisconsin breast cancer | 699 | 8 | 241/458 (1/0) |
| Statlog heart disease | 270 | 14 | 120/150 (1/0) |
| Focality of prostate cancer | 500 | 10 | 91/409 (0/1) |
| Bladder cancer | 234 | 8 | 97/137 (1/0) |

ferase), SGOT (Aspartate Aminotransferase), total protiens, Albumin, Albumin and Globulin Ratio, selector field (class labels labelled by the experts).

4. Wisconsin Breast Cancer Data Set: This data set contains instances taken from fine needle aspirates of human breast tissue. The data set was obtained from the University of Wisconsin Hospitals, Madison, United States. There are 16 instances that contain a single missing attribute value. The first attribute (sample code number) is discarded. The attributes of the data set are: sample code number, clump thickness, uniformity of cell size, uniformity of cell shape, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli, mitoses, class labels (2 for benign, 4 for malignant).

5. Statlog Heart Disease Data Set: This database contains 13 (excluding the class labels) attributes extracted from a larger set of 75. The attributes of the data set are: age, sex, chest pain type (4 values), resting blood pressure, serum cholestoral in mg/dl, fasting blood sugar > 120 mg/dl, resting electrocardiographic results (values 0,1,2), maximum heart rate achieved, exercise induced angina, oldpeak (ST depression induced by exercise relative to rest), the slope of the peak exercise ST segment, number of major vessels (0-3) colored by flourosopy, thal (3 = normal; 6 = fixed defect; 7 = reversible defect), class labels (absence (1) or presence (2) of heart disease).

6. Focality of Prostate Cancer Data Set: The attributes of the data set are: PSA (Prostate-Specific Antigen), PSA density, Bx (Gleason score), Bilateral, total cores invaded, total cores, total length of tumour, total length, total length of benign tissue, class labels (multifocal, 1 or 0).

7. Bladder Cancer Data Set: Bladder cancer data set comprises of progression information of patients who had undergone surgical tumour removal for bladder cancer. The attributes of the data set are: age of patient, sex of patient, grade of tumour, stage of tumour, location of tumour (upper or lower half of urinary tract), ever smoked, methylation percentage, class labels (progression, 1 or 0).

### 3.4.2 Classification Measures

Several performance measures can be used to evaluate the performance of a classifier. When the output of a classifier is compared to the target class labels, there can be four types of outcome:

- True Positive (TP): when a positive class sample is correctly predicted.
- True Negative (TN): when a negative class sample is correctly predicted.
- False Positive (FP): when a negative class sample is incorrectly predicted as positive class.
- False Negative (FN): when a positive class sample is incorrectly predicted as negative class.

Where the positive and negative classes are chosen conventionally to be the minority and majority classes respectively. The number of samples that belong to each of the above outcomes can be summarised in a Confusion Matrix, as in Table 3.2. Several measures are calculated based on some or all of these numbers aiming to provide an accurate measure from a particular perspective. The simplest and widely used measure is the over all accuracy defined as the ratio of the number of correctly classified samples to the number of all samples:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{3.17}$$

Another measure is the Sensitivity (or True Positive Rate) which is the ratio of the number of (TP) samples to the number of all positive samples:

$$Sensitivity = \frac{TP}{TP + FN} \tag{3.18}$$

TABLE 3.2. Confusion Matrix.

| Actual Class | Predicted Positive | Predicted Negative |
|---|---|---|
| Actual Positive | $TP$ | $FN$ |
| Actual Negative | $FP$ | $TN$ |

Similarly, the ratio of the number of (TN) samples to the number of all negative samples is the Specificity (or True Negative Rate):

$$Specificity = \frac{TN}{TN + FP} \tag{3.19}$$

It can be noticed that for balanced data (number of positive samples ≈ number of positive samples) accuracy measure reflects the measures of specificity and sensitivity as well. However, it is not the case with imbalanced data where high accuracy can be achieved at the price of low specificity or sensitivity. For example, a classifier whose output is always 1 can achieve 90% accuracy for data with 90% samples of class 1, however, the sensitivity in this case is 0%. Another issue with calculating the accuracy is the choice of the threshold value that separate the output values into output class labels. The threshold value has a direct influence on accuracy and its data dependant.

A performance measure that aims to overcome the above issues is Receiver Operating Characteristic (ROC). ROC is a plot of the true positive rate (sensitivity) versus false positive rate (1- specificity) while varying the threshold value. Higher value of the sensitivity (y axis of the ROC) along with lower value of false positive rate (x axis of ROC) means better classification performance. Thus, the Area Under Curve (AUC) of the ROC is an indicator of the performance of the classifier in terms of sensitivity and specificity for the full range of threshold values. Larger values of AUC indicate better classification performance. Figure 3.4 shows an example ROC with the red circle marking the optimal operation point. The optimal operation point is chosen as the first intersection point of the ROC curve with a straight line with slope $S$ moving from from the upper-left corner of the ROC plot towards the bottom-right corner. The slope $S$ is given by:

$$S = \frac{N}{P} \tag{3.20}$$

where $N = TN + FP$ and $P = TP + FN$.

To evaluate classification performance of the classifiers in this chapter and the next chapters, the ROC is computed using training data to find the optimal threshold

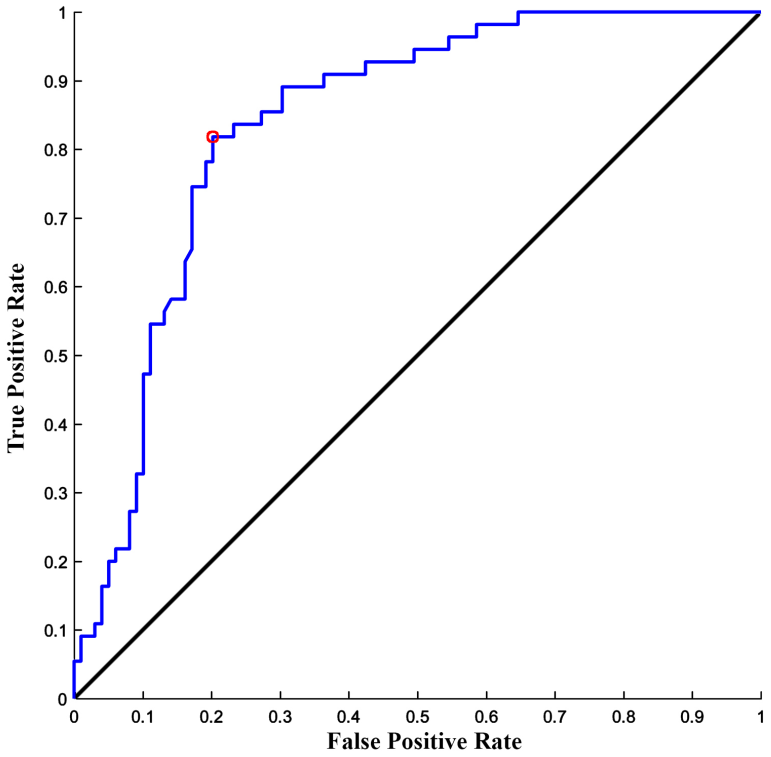

FIGURE 3.4. Receiver operating characteristic curve.

which is used to compute the output class labels of the test data. Then, both accuracy and AUC are computed to measure classification performance.

### 3.4.3 Granulation Results

Table 3.3 shows the results of granulation of the selected data sets using FCMGr. These results are for training/testing partition P1 of each data set. The number of granulation levels can be controlled by the variable $\Delta R$. Increasing $\Delta R$ increases the number of merged clusters, hence reduces the number of granulation levels, and vice versa. To produce granulation levels that have significant change in number of clusters, a suitable $\Delta R$ has to be chosen. For example, referring to Table 3.3, a suitable $\Delta R$ for Pima data set is found to be 0.05 while the suitable $\Delta R$ for Bladder cancer data set is found to be 0.1. The suitable value of $\Delta R$ can be found by varying it until a desired number of granularity levels is achieved.

In addition, it should be noticed that the number of granulation levels and the size and number of clusters in each granulation level depend on the data samples being granulated, that is using different data samples (e.g. different training/testing partitioning) may result in different number of granulation levels and/or different size and number of clusters in each granulation level. Table 3.4 shows the results of granulation of the selected data sets using training/testing partitioning P2. For instance, Pima data set has 11 granulation levels in Table 3.3 versus 12 granulation levels in Table 3.4 for the same value of $\Delta R$. The results shown in Tables 3.3 and 3.4 include the selected value of the fuzziness factor $m$ for each granulation level.

## 3.5 Fuzzy Classifier Design

To evaluate the proposed clustering algorithm, the resulted clusters at each granulation level are used to build a fuzzy classifier. The performance of the classifiers is evaluated using the selected data sets and two classification performance measures: accuracy and AUC. Following is a brief introduction to the fuzzy classification systems followed by the experimental results.

### 3.5.1 Fuzzy Inference System

FISs are widely used in many applications, from system modelling, simulation and control to classification and decision support. FIS is a computational technique that relies on fuzzy logic for performing input-output mapping. It inherits, from fuzzy logic, the ability to deal with vagueness and incompleteness of knowledge and the ability of describing non-linear relationships between input and output.

A FIS consists of three components (as shown in Figure 3.5): a set of fuzzy if-then rules, a database that defines the fuzzy membership functions used in these rules and a reasoning mechanism that performs the inference process. Inputs are transformed to fuzzy sets using the membership functions defined in the database of the FIS and the output is computed from the fuzzy rules using the reasoning mechanism. The consequent parts of the rules of the FIS can be fuzzy values (Mamdani model [152]), a crisp value (type II), or a function of linear combination of input variables

TABLE 3.3. Granulation results for the selected data sets (P1 partitioning).

Pima Indian diabetes data set

No of levels = 11, $\Delta R$ = 0.05

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $c$ | 614 | 351 | 211 | 131 | 79 | 47 | 27 | 14 | 7 | 4 | 2 |
| $m$ | 4.00 | 1.70 | 1.75 | 1.75 | 1.70 | 1.70 | 1.65 | 1.60 | 1.60 | 1.60 | 1.80 |

BUPA liver disorders data set

No of levels = 11, $\Delta R$ = 0.05

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $c$ | 273 | 146 | 86 | 54 | 37 | 25 | 15 | 9 | 6 | 4 | 2 |
| $m$ | 4.00 | 1.80 | 1.90 | 1.95 | 1.80 | 1.85 | 2.10 | 1.90 | 1.80 | 1.80 | 1.45 |

ILPD data set

No of levels = 11, $\Delta R$ = 0.1

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $c$ | 463 | 255 | 140 | 80 | 44 | 27 | 15 | 9 | 5 | 3 | 2 |
| $m$ | 4.00 | 1.70 | 1.65 | 1.60 | 1.55 | 1.40 | 1.40 | 1.40 | 1.50 | 1.50 | 1.30 |

Wisconsin breast cancer data set

No of levels = 13, $\Delta R$ = 0.1

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $c$ | 546 | 367 | 257 | 178 | 125 | 85 | 51 | 29 | 16 | 9 | 5 | 3 | 2 |
| $m$ | 4.00 | 4.00 | 1.70 | 1.70 | 1.65 | 1.65 | 1.60 | 1.60 | 1.55 | 1.75 | 2.15 | 2.20 | 2.30 |

Statlog heart disease data set

No of levels = 13, $\Delta R$ = 0.15

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $c$ | 216 | 191 | 162 | 130 | 105 | 77 | 58 | 37 | 21 | 11 | 6 | 4 | 2 |
| $m$ | 2.95 | 1.75 | 1.50 | 1.50 | 1.45 | 1.40 | 1.35 | 1.35 | 1.30 | 1.30 | 1.30 | 1.30 | 1.40 |

Focality of prostate cancer data set

No of levels = 12, $\Delta R$ = 0.1

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $c$ | 400 | 217 | 121 | 70 | 39 | 22 | 13 | 8 | 6 | 4 | 3 | 2 |
| $m$ | 4.00 | 2.55 | 2.25 | 1.95 | 1.75 | 1.50 | 1.45 | 1.40 | 1.45 | 1.30 | 2.20 | 4.00 |

Bladder cancer data set

No of levels = 13, $\Delta R$ = 0.1

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $c$ | 187 | 132 | 97 | 77 | 54 | 39 | 35 | 32 | 22 | 14 | 8 | 4 | 2 |
| $m$ | 1.30 | 2.10 | 1.95 | 1.85 | 1.75 | 1.70 | 1.55 | 1.60 | 1.45 | 1.35 | 1.55 | 1.40 | 1.75 |

TABLE 3.4. Granulation results for the selected data sets (P2 partitioning).

Pima Indian diabetes data set

No of levels = 12, $\Delta R$ = 0.05

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $c$ | 614 | 348 | 208 | 126 | 76 | 45 | 27 | 15 | 9 | 5 | 3 | 2 |
| $m$ | 4.00 | 1.70 | 1.70 | 1.70 | 1.70 | 1.60 | 1.60 | 1.55 | 1.55 | 1.50 | 1.50 | 1.75 |

BUPA liver disorders data set

No of levels = 12, $\Delta R$ = 0.05

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $c$ | 273 | 146 | 84 | 53 | 36 | 25 | 17 | 13 | 8 | 6 | 4 | 2 |
| $m$ | 4.00 | 1.65 | 1.65 | 1.70 | 1.60 | 1.60 | 1.45 | 1.45 | 1.50 | 1.45 | 1.45 | 1.35 |

ILPD data set

No of levels = 10, $\Delta R$ = 0.1

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $c$ | 463 | 254 | 138 | 76 | 39 | 22 | 11 | 6 | 3 | 2 |
| $m$ | 4.00 | 1.80 | 1.70 | 1.60 | 1.60 | 1.40 | 1.30 | 1.40 | 1.50 | 1.30 |

Wisconsin breast cancer data set

No of levels = 13, $\Delta R$ = 0.1

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $c$ | 546 | 370 | 269 | 188 | 131 | 89 | 52 | 29 | 17 | 10 | 5 | 3 | 2 |
| $m$ | 1.30 | 1.65 | 1.60 | 1.55 | 1.50 | 1.45 | 1.35 | 1.30 | 1.30 | 1.30 | 1.55 | 1.80 | 2.05 |

Statlog heart disease data set

No of levels = 13, $\Delta R$ = 0.15

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $c$ | 216 | 193 | 165 | 130 | 102 | 75 | 53 | 36 | 20 | 11 | 6 | 4 | 2 |
| $m$ | 2.95 | 1.75 | 1.55 | 1.55 | 1.40 | 1.45 | 1.30 | 1.35 | 1.30 | 1.30 | 1.30 | 1.30 | 1.40 |

Focality of prostate cancer data set

No of levels = 12, $\Delta R$ = 0.1

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $c$ | 400 | 222 | 125 | 73 | 44 | 28 | 17 | 10 | 7 | 4 | 3 | 2 |
| $m$ | 4.00 | 4.00 | 4.00 | 2.80 | 2.30 | 1.75 | 1.65 | 1.50 | 1.50 | 1.30 | 2.35 | 4.00 |

Bladder cancer data set

No of levels = 14, $\Delta R$ = 0.1

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $c$ | 187 | 128 | 94 | 75 | 54 | 39 | 32 | 31 | 22 | 15 | 9 | 5 | 3 | 2 |
| $m$ | 1.30 | 2.30 | 2.10 | 2.00 | 1.90 | 1.80 | 1.75 | 1.75 | 1.40 | 1.60 | 1.55 | 1.45 | 1.60 | 1.95 |

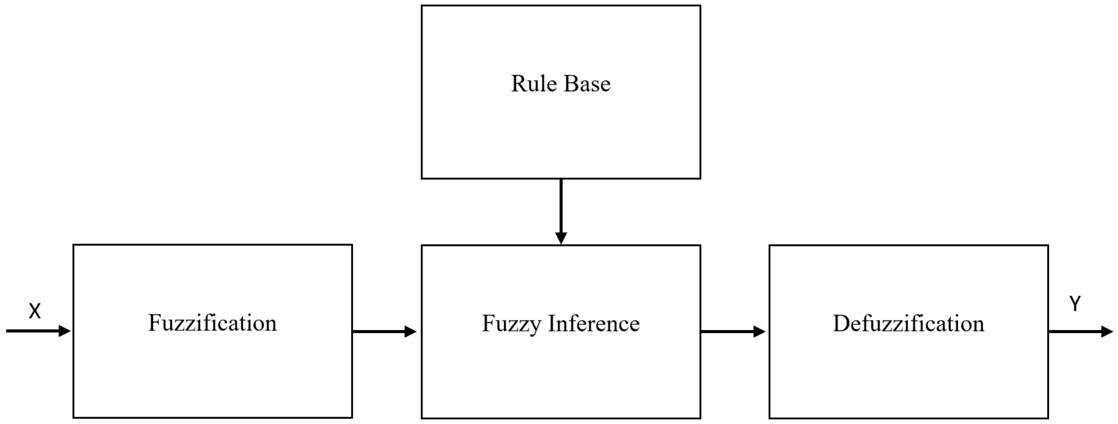

FIGURE 3.5. Block Diagram of a Fuzzy Inference System.

(TSK model [153]). In the case of Mamdani model, the process of defuzzification is required to produce the final crisp value of the output.

Information required to design FIS and construct its rules can be obtained from the knowledge of human experts. However, with increasing the complexity of the system, the number of inputs and the number of fuzzy variables for each input, it becomes more difficult for human experts to generate these rules. Instead, some computational methods (e.g. data clustering) can be used to generate fuzzy rules from data automatically.

The central problem of fuzzy modelling is the generation of fuzzy rules that fit the data to the highest possible extent. The task of generating and learning fuzzy rules from numerical data has been addressed by different approaches, some examples are: clustering methods [154], particle swarm optimisation [155], fuzzy genetic algorithm [156], rough sets model with genetic algorithms [157] and data mining methods [158].

Data clustering is one of the most widely used approaches for rule base generation. Fuzzy sets of an input variable can be obtained from data clusters by projecting each cluster to the dimension that corresponds to the input as shown in Figure 2.2.

### 3.5.2 Fuzzy System Identification

For each granulation level, a fuzzy system is constructed from the clusters centres $\hat{V}_g$, the fuzzy partition matrix $\hat{U}_g$ and the training data samples. A fuzzy rule is constructed from each cluster with the number of antecedent fuzzy variables equals to the dimensionality of the cluster (number of inputs) and the consequent fuzzy variable corresponds to the class label of the cluster. For an antecedent fuzzy variable at dimension $d$, a fuzzy membership function is defined for each cluster using the cluster centre at the corresponding dimension $\hat{v}_j(d)$, the corresponding values of the fuzzy partition matrix $\hat{u}_{ji}$ and and the training data samples at the corresponding dimension $\hat{X}(d)$. Gaussian membership functions are used to represent the fuzzy variables. A Gaussian function has a symmetric bell curve shape and is defined as follows:

$$f(x) = ae^{-x-\mu^2/2\sigma^2} \tag{3.21}$$

where the parameter $a$ determines the peak of the curve, $\mu$ is the centre of the curve and the parameter $\sigma$ controls the width of curve. So, a Gaussian function of peak of 1 is defined by the two parameters $\mu$ and $\sigma$. A Gaussian membership function is defined using the cluster centre $\hat{v}_j(d)$, the fuzzy c-partition matrix $\hat{u}_{ji}$ and and the training data samples $\hat{X}(d)$ by setting $\mu = \hat{v}_j(d)$ and computing $\sigma_{dj}$ as follows:

$$\sigma_{dj} = \frac{1}{n}\sum_{i=1}^{n}\sqrt{-\frac{(\hat{x}_i(d) - \hat{v}_j(d))^2}{2log(\hat{u}_{ji})}} \tag{3.22}$$

For data samples $\hat{X}$ of dimensionality $D$, $\sigma_{Dj}$ and $\hat{v}_j(D)$ define the Gaussian membership function of the consequent fuzzy variable.

### 3.5.3 Classification Results

The FCMGr algorithm is implemented using MATLAB R2016a. Using FCMGr, a FIS is built from each granulation level of the selected data sets and compared to standard FCM with various values of fuzziness factor $m$. Since standard FCM use random initial fuzzy c-partition $U$, each FCM classifier is trained and tested for

10 times and the mean result is calculated. Detailed results for 4 sample data sets are given in Tables 3.5-3.8 and a summary of the results for all the data sets is given in Table 3.9. Table 3.5 shows the test results for the bladder cancer data set at various granulation levels. The values of fuzziness factor $m$ are chosen from the range (1.3-2.8) in steps of 0.25. The table shows the accuracy (%) and AUC (between parentheses) with results in bold indicates the highest (best) result. The results shown in the table are for granulation levels with number of clusters less than 50. From the table, it can be noticed that fuzzy classifiers built from the granules of FCMGr have better results at most of the granulation levels. Similar results can be seen in Table 3.7 for the Wisconsin breast cancer data set, although the AUC of FCMGr is lower than the standard FCM at most of the granulation levels which suggests that the improvement in accuracy is a result of correctly classifying more samples of the majority class at the price of misclassifying more samples of the minority class.

However, the results of FCMGr are higher in only about half the granulation levels of the heart disease data set, as shown in Table 3.6, and less than half the granulation levels of the BUPA liver disorders data set, as shown in Table 3.8.

To summarise the results for all data sets, the results of the granulation level that has best combined accuracy and AUC over the selected range of granulation levels are compared to the results of standard FCM classifiers with number of clusters that results in best combined accuracy and AUC. These results are shown in Table 3.9. The results show that FCMGr achieved better classification performance in terms of accuracy and AUC. However, for each set of training data, best results are achieved from a different granulation level. Therefore, it is necessary to find a method of automatically selecting granulation levels that results in better classification performance. This task is equivalent to a crucial task in the identification of a FIS, that is, selecting the number of fuzzy rules of a FIS that results in better classification performance.

### 3.5.4 Automatic Selection of Granulation Level

To select the granulation level that results in highest classification quality, three measures are considered for each level:

TABLE 3.5. Accuracy (%) and AUC (%) for bladder cancer data set.

| $N_C$ | FCMGr | FCM<br>m=1.30 | FCM<br>m=1.55 | FCM<br>m=1.80 | FCM<br>m=2.05 | FCM<br>m=2.30 | FCM<br>m=2.55 | FCM<br>m=2.80 |
|---|---|---|---|---|---|---|---|---|
| 39 | **65.96**<br>(**72.07**) | 54.68<br>(50.79) | 55.53<br>(57.96) | 58.51<br>(58.90) | 54.68<br>(48.44) | 55.53<br>(46.95) | 54.89<br>(46.74) | 56.38<br>(45.93) |
| 35 | **65.96**<br>(**63.83**) | 53.62<br>(50.04) | 56.81<br>(59.81) | 58.51<br>(60.96) | 55.32<br>(48.03) | 54.47<br>(46.77) | 58.09<br>(48.10) | 54.47<br>(46.91) |
| 32 | **63.83**<br>(**64.84**) | 52.98<br>(55.56) | 58.72<br>(62.07) | 60.21<br>(62.09) | 54.47<br>(47.52) | 54.26<br>(46.13) | 55.96<br>(44.75) | 56.60<br>(48.37) |
| 22 | **61.70**<br>(**67.58**) | 58.51<br>(62.92) | 58.94<br>(64.84) | 60.21<br>(63.69) | 54.89<br>(47.11) | 55.96<br>(47.77) | 54.68<br>(46.44) | 55.74<br>(49.51) |
| 14 | 63.83<br>(**72.44**) | 62.34<br>(63.74) | **64.04**<br>(69.94) | 62.34<br>(64.54) | 55.96<br>(53.81) | 55.96<br>(48.64) | 55.96<br>(44.07) | 56.81<br>(44.01) |
| 8 | **68.09**<br>(**70.60**) | 63.19<br>(63.99) | 64.47<br>(67.39) | 65.74<br>(66.08) | 55.53<br>(52.01) | 53.83<br>(49.82) | 54.89<br>(47.29) | 57.87<br>(47.25) |
| 4 | 57.45<br>(61.63) | 58.51<br>(63.77) | 62.77<br>(62.10) | **63.83**<br>(**64.95**) | 54.89<br>(51.98) | 55.53<br>(52.05) | 54.68<br>(44.84) | 55.53<br>(47.02) |
| 2 | 61.70<br>(60.99) | 62.98<br>(62.25) | **65.96**<br>(**64.52**) | 62.77<br>(61.36) | 57.23<br>(58.58) | 58.51<br>(54.39) | 54.89<br>(41.38) | 57.45<br>(47.00) |

TABLE 3.6. Accuracy (%) and AUC (%) for Statlog heart disease data set.

| $N_C$ | FCMGr | FCM<br>m=1.30 | FCM<br>m=1.55 | FCM<br>m=1.80 | FCM<br>m=2.05 | FCM<br>m=2.30 | FCM<br>m=2.55 | FCM<br>m=2.80 |
|---|---|---|---|---|---|---|---|---|
| 37 | 57.41<br>(63.96) | **72.78**<br>(**77.64**) | 62.78<br>(68.05) | 66.48<br>(74.83) | 62.96<br>(69.90) | 61.48<br>(65.51) | 55.37<br>(59.42) | 57.41<br>(56.71) |
| 21 | **79.63**<br>(**81.32**) | 72.59<br>(79.40) | 62.96<br>(68.06) | 62.59<br>(69.21) | 62.41<br>(69.34) | 60.37<br>(64.03) | 58.33<br>(61.13) | 58.15<br>(55.09) |
| 11 | **77.78**<br>(**81.11**) | 70.56<br>(76.13) | 64.26<br>(74.12) | 59.26<br>(63.95) | 61.48<br>(68.90) | 60.56<br>(62.45) | 58.52<br>(64.36) | 53.70<br>(55.38) |
| 6 | 64.81<br>(77.85) | **68.89**<br>(**78.23**) | 65.74<br>(75.75) | 61.30<br>(68.56) | 63.52<br>(69.33) | 60.00<br>(64.10) | 60.37<br>(62.96) | 57.41<br>(55.15) |
| 4 | **72.22**<br>(**76.81**) | 66.67<br>(76.38) | 63.33<br>(71.78) | 63.52<br>(70.69) | 57.78<br>(64.28) | 61.11<br>(68.12) | 60.00<br>(61.01) | 56.48<br>(54.00) |
| 2 | **70.37**<br>(76.60) | **70.37**<br>(**76.67**) | 62.04<br>(72.13) | 60.56<br>(69.23) | 64.26<br>(71.06) | 61.30<br>(68.11) | 56.48<br>(57.26) | 53.89<br>(54.23) |

TABLE 3.7. Accuracy (%) and AUC (%) for Wisconsin breast cancer data set.

| $N_C$ | FCMGr | FCM<br>m=1.30 | FCM<br>m=1.55 | FCM<br>m=1.80 | FCM<br>m=2.05 | FCM<br>m=2.30 | FCM<br>m=2.55 | FCM<br>m=2.80 |
|---|---|---|---|---|---|---|---|---|
| 29 | **94.89**<br>(97.17) | 92.55<br>(97.51) | 92.85<br>(**97.58**) | 92.04<br>(97.42) | 93.72<br>(96.35) | 92.99<br>(95.56) | 92.99<br>(95.73) | 92.70<br>(95.29) |
| 16 | **96.35**<br>(**98.40**) | 91.82<br>(98.20) | 93.28<br>(97.74) | 92.48<br>(97.72) | 92.77<br>(96.10) | 92.85<br>(95.45) | 92.85<br>(95.60) | 92.85<br>(95.69) |
| 9 | 91.97<br>(98.00) | 91.09<br>(97.99) | **95.77**<br>(**98.31**) | 93.65<br>(97.58) | 93.43<br>(96.28) | 92.85<br>(95.51) | 92.77<br>(95.33) | 92.92<br>(95.74) |
| 5 | **95.62**<br>(96.89) | 92.63<br>(**97.81**) | 94.01<br>(97.70) | 93.87<br>(96.73) | 93.43<br>(96.47) | 92.92<br>(95.93) | 92.70<br>(95.62) | 92.70<br>(95.50) |
| 3 | **94.16**<br>(97.40) | 91.82<br>(96.30) | 91.24<br>(**97.43**) | 91.53<br>(97.16) | 93.43<br>(97.25) | 94.16<br>(97.30) | 94.16<br>(96.18) | 94.16<br>(96.20) |
| 2 | **93.43**<br>(97.01) | 91.24<br>(96.21) | 92.70<br>(**97.29**) | 90.51<br>(97.17) | 92.70<br>(96.99) | **93.43**<br>(97.01) | 92.70<br>(96.62) | 92.70<br>(95.85) |

TABLE 3.8. Accuracy (%) and AUC (%) for BUPA liver disorders data set.

| $N_C$ | FCMGr | FCM<br>m=1.30 | FCM<br>m=1.55 | FCM<br>m=1.80 | FCM<br>m=2.05 | FCM<br>m=2.30 | FCM<br>m=2.55 | FCM<br>m=2.80 |
|---|---|---|---|---|---|---|---|---|
| 37 | **67.65**<br>(**61.64**) | 56.76<br>(47.14) | 54.85<br>(55.54) | 56.47<br>(54.41) | 63.68<br>(56.15) | 57.79<br>(48.62) | 59.12<br>(49.24) | 60.74<br>(51.13) |
| 25 | **64.71**<br>(**58.45**) | 54.12<br>(44.64) | 58.38<br>(52.89) | 58.97<br>(55.93) | 63.38<br>(56.43) | 60.44<br>(49.24) | 59.71<br>(50.53) | 59.12<br>(49.51) |
| 15 | 58.82<br>(51.01) | 55.59<br>(48.46) | 57.79<br>(55.35) | 64.41<br>(55.43) | **65.15**<br>(**59.58**) | 57.79<br>(48.95) | 58.97<br>(49.30) | 60.88<br>(49.27) |
| 9 | **66.18**<br>(53.24) | 58.97<br>(**54.41**) | 65.15<br>(51.35) | 62.50<br>(52.23) | 61.32<br>(53.77) | 59.71<br>(46.42) | 54.12<br>(48.46) | 61.18<br>(49.78) |
| 6 | **64.71**<br>(53.29) | 61.18<br>(**53.82**) | 62.50<br>(52.91) | 58.24<br>(41.67) | 57.50<br>(45.71) | 59.12<br>(45.77) | 56.18<br>(47.89) | 57.94<br>(49.09) |
| 4 | 60.29<br>(38.50) | 59.26<br>(55.93) | 56.76<br>(**59.89**) | 60.44<br>(40.18) | 58.38<br>(42.66) | 57.06<br>(45.21) | 58.97<br>(48.09) | **62.06**<br>(49.71) |
| 2 | 61.76<br>(48.12) | 57.35<br>(48.99) | 60.29<br>(47.73) | 63.24<br>(49.61) | 64.71<br>(49.61) | 64.71<br>(50.70) | **66.18**<br>(52.32) | **66.18**<br>(**53.91**) |

TABLE 3.9. Maximum values of accuracy (%) and AUC (%) for all the data sets.

| Data set | FCMGr | FCM<br>m=1.30 | FCM<br>m=1.55 | FCM<br>m=1.80 | FCM<br>m=2.05 | FCM<br>m=2.30 | FCM<br>m=2.55 | FCM<br>m=2.80 |
|---|---|---|---|---|---|---|---|---|
| Pima | **79.22**<br>(**84.32**) | 74.68<br>(80.99) | 75.84<br>(82.00) | 75.00<br>(80.82) | 68.83<br>(79.58) | 68.64<br>(75.39) | 70.13<br>(77.92) | 69.61<br>(78.89) |
| BUPA | **67.65**<br>(**61.64**) | 61.62<br>(55.47) | 56.76<br>(59.89) | 59.26<br>(61.37) | 65.15<br>(59.58) | 64.71<br>(50.70) | 66.18<br>(52.32) | 66.18<br>(53.91) |
| ILPD | 68.10<br>(**66.91**) | **70.17**<br>(60.46) | 69.83<br>(58.68) | 66.38<br>(57.06) | 67.50<br>(56.73) | 68.45<br>(58.85) | 67.67<br>(60.96) | 69.05<br>(60.43) |
| Wisconsin | **96.35**<br>(**98.40**) | 92.63<br>(97.81) | 95.77<br>(98.31) | 93.65<br>(97.58) | 93.43<br>(97.25) | 94.16<br>(97.30) | 94.16<br>(96.18) | 94.16<br>(96.20) |
| Statlog | **79.63**<br>(**81.32**) | 72.59<br>(79.40) | 73.33<br>(81.23) | 67.78<br>(77.62) | 64.26<br>(73.44) | 63.33<br>(71.29) | 60.00<br>(66.10) | 57.41<br>(56.71) |
| Focality | 84.00<br>(70.12) | 85.00<br>(69.25) | **85.80**<br>(69.89) | 85.00<br>(67.91) | 83.90<br>(68.99) | 83.30<br>(72.70) | 85.00<br>(71.31) | 85.00<br>(**72.96**) |
| Bladder cancer | **68.09**<br>(**70.60**) | 63.19<br>(63.99) | 64.04<br>(69.94) | 65.74<br>(66.08) | 57.23<br>(58.58) | 58.51<br>(54.39) | 56.81<br>(49.58) | 56.60<br>(48.99) |

- The minimum value of $F_g(m)$ defined by Equation 3.11 that is obtained in generating level $g$.

- The minimum value of the objective function $J^k_{g,m}(\hat{U}^k, \hat{V}^k)$ defined by Equation 3.16 that is obtained in generating level $g$ at $m = m_{best}$.

- Number of granules $N_C$.

The three measures are combined in the following equation:

$$T_g = AF_g(m) + BJ^k_{g,m}(\hat{U}^k, \hat{V}^k) + CN_{C,m} \tag{3.23}$$

where $A$, $B$ and $C$ are scaling variables chosen so that the three measures are normalised. That is:

$$A = 1/max(F_g(m)) \tag{3.24}$$

$$B = 1/max(J^k_{g,m}(\hat{U}^k, \hat{V}^k)) \tag{3.25}$$

$$C = 1/max(N_{C,m}) \tag{3.26}$$

The granulation level that minimises $T_g$ is selected to build the fuzzy classifier. The classification results of the automatically selected granulation level for each selected data set are shown in Table 3.10. The results are compared to the highest results of FCM fuzzy classifiers with same values of fuzziness factor $m$ selected in Section 3.5.3. The highest results for each value of $m$ are chosen from 7 FCM fuzzy classifiers with number of clusters equal to 2, 3, 5, 10, 15, 25, 35. The number of clusters for the FCM classifiers are fixed because the experiment is conducted 5 times with different training/testing partitioning each time (P1-P5). Therefore, number of clusters of each granulation level of the FCMGr is different for each training/testing partitioning. The FCM fuzzy classifiers are trained and tested 10 times and the mean result is used.

It should be noticed that this approach of automatically selecting the granularity level uses only training data since the desired outputs of testing data are unknown in practical cases. As a result, the selected granularity level using the proposed approach may not be the optimal granularity level for test data.

Results in Table 3.10 show that the granulation levels selected using the above equation resulted in better accuracy than the selected FCM classifiers for all the data sets over a wide range of number of clusters ($N_C$ and fuzziness factor $m$. In addition, the AUC of FCMGr is higher for 4 out of 7 data sets.

The automatically selected granulation level is different for each training/testing partitioning. This can be seen in Figures 3.6 and 3.7 for Pima data set. Figure 3.6 shows the accuracy of 3 FCM classifiers with various number of clusters compared to the accuracy of FCMGr at various granulation levels for training/testing partitioning P1. Since the number of clusters in FCMGr is determined automatically, it is not necessarily equal to the fixed number of clusters of the FCM classifiers. The automatically selected granulation level, marked by a blue circle, has 7 clusters. Figure 3.7 shows the results for training/testing partitioning P2 of the same data set (Pima). In this case, the automatically selected granulation level has 9 clusters. Figure 3.8 shows the mean accuracy for the 5 training/testing partitioning (P1-P5). In this case, the accuracy of FCMGr is the mean of accuracies of 5 granulation levels with different number of clusters.

TABLE 3.10. Accuracy (%) and AUC (%) of automatic selection of granulation level.

| Data set | FCMGr | FCM<br>m=1.30 | FCM<br>m=1.55 | FCM<br>m=1.80 | FCM<br>m=2.05 | FCM<br>m=2.30 | FCM<br>m=2.55 | FCM<br>m=2.80 |
|---|---|---|---|---|---|---|---|---|
| Pima | 73.16<br>(78.17) | 74.68<br>(80.99) | **75.84**<br>(**82.00**) | 75.00<br>(80.82) | 68.83<br>(79.58) | 68.64<br>(75.39) | 70.13<br>(77.92) | 69.61<br>(78.89) |
| BUPA | 64.22<br>(**63.84**) | 61.62<br>(55.47) | 56.76<br>(59.89) | 59.26<br>(61.37) | 65.15<br>(59.58) | 64.71<br>(50.70) | **66.18**<br>(52.32) | **66.18**<br>(53.91) |
| ILPD | **72.03**<br>(**64.03**) | 70.17<br>(60.46) | 69.83<br>(58.68) | 66.38<br>(57.06) | 67.50<br>(56.73) | 68.45<br>(58.85) | 67.67<br>(60.96) | 69.05<br>(60.43) |
| Wisconsin | **95.91**<br>(**98.43**) | 92.63<br>(97.81) | 95.77<br>(98.31) | 93.65<br>(97.58) | 93.43<br>(97.25) | 94.16<br>(97.30) | 94.16<br>(96.18) | 94.16<br>(96.20) |
| Statlog | **78.15**<br>(**85.82**) | 72.59<br>(79.40) | 73.33<br>(81.23) | 67.78<br>(77.62) | 64.26<br>(73.44) | 63.33<br>(71.29) | 60.00<br>(66.10) | 57.41<br>(56.71) |
| Focality | 81.60<br>(59.87) | 85.00<br>(69.25) | **85.80**<br>(69.89) | 85.00<br>(67.91) | 83.90<br>(68.99) | 83.30<br>(72.70) | 85.00<br>(71.31) | 85.00<br>(**72.96**) |
| Bladder cancer | **71.01**<br>(68.11) | 63.19<br>(63.99) | 64.04<br>(**69.94**) | 65.74<br>(66.08) | 57.23<br>(58.58) | 58.51<br>(54.39) | 56.81<br>(49.58) | 56.60<br>(48.99) |

### 3.5.5 Multi-Level FCMGr

Instead of using one granulation level to construct a FIS, multi-levels of the FCMGr can be used in the same way to produce a FIS. The set of all clusters of the selected

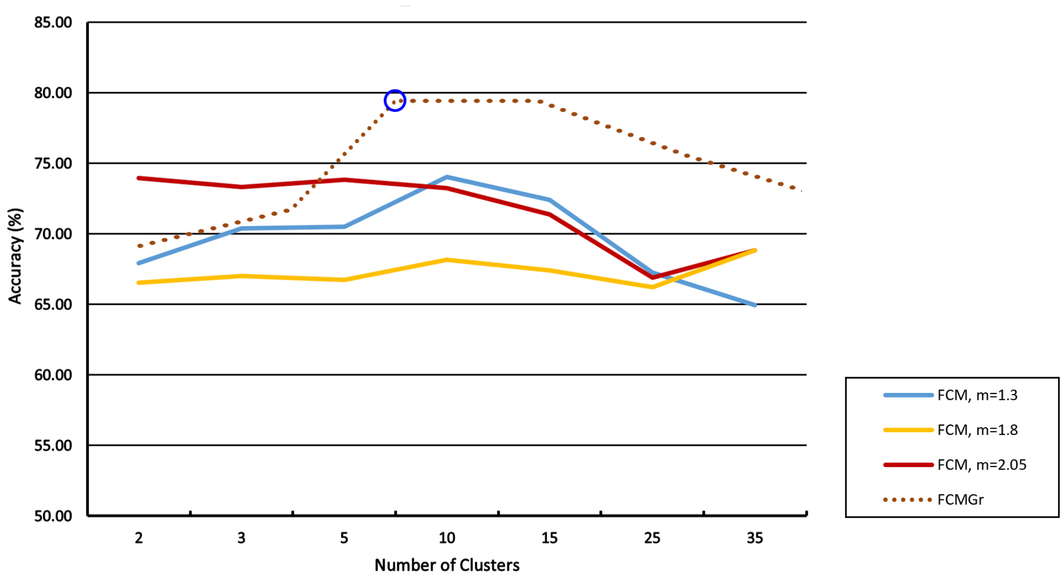

FIGURE 3.6. Accuracy of automatic selection of best granulation levels for Pima cancer data set (P1)

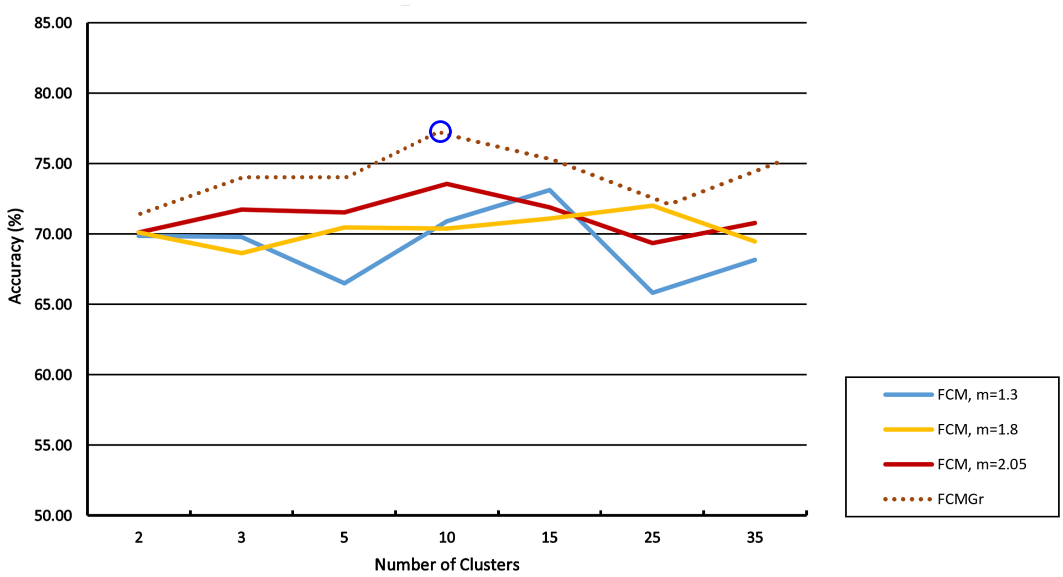

FIGURE 3.7. Accuracy of automatic selection of best granulation levels for Pima cancer data set (P2).

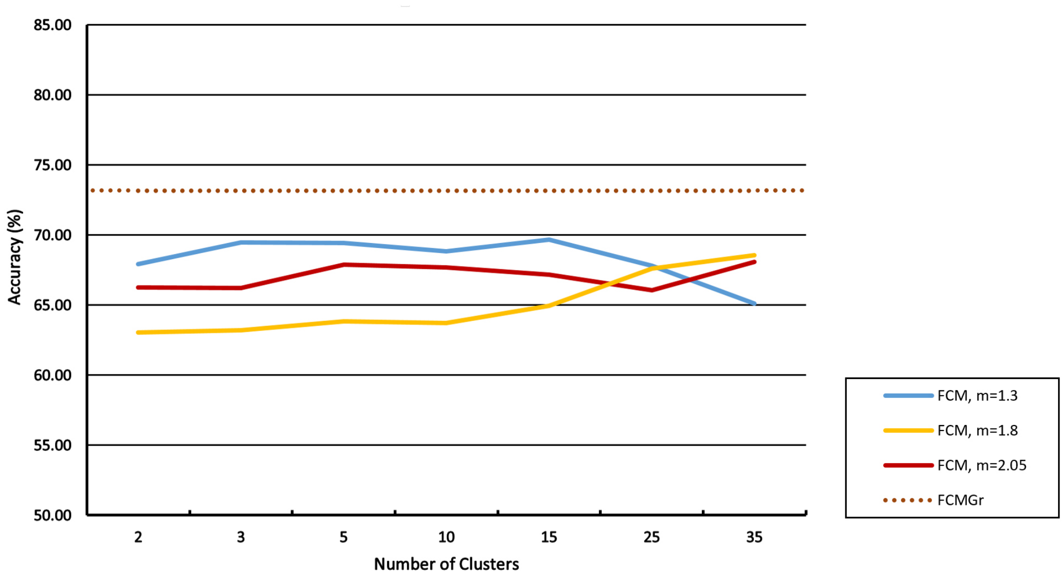

FIGURE 3.8. Mean accuracy of automatic selection of best granulation levels for Pima cancer data set (P1-P5).

levels and the concatenation of all the fuzzy c-partition matrices of the selected levels are used to construct the FIS. The constructed FIS consists of rules of different levels of abstraction/specificity, thus it may provide better results than FIS generated using single-level FCMGr for data sets where useful knowledge can be gained from different levels of abstraction/specificity.

Table 3.11 shows the test accuracy and AUC results of fuzzy classifiers constructed using multi-level FCMGr (ML FCMGr) compared to fuzzy classifiers constructed using single-level FCMGr (SL FCMGr) and 3 FCM-based fuzzy classifiers. Multi-level FCMGr has better accuracy than single-level FCMGr for three data sets, which suggests that, for these data sets, useful knowledge can be gained from different levels of abstraction/specificity. In addition, Table 3.11 shows the granulation levels selected for each data set. It should be noticed that the granulation levels are selected experimentally by evaluating various levels and selecting the level with highest results.

TABLE 3.11. Accuracy (%) and AUC (%) for multi-level FCMGr, single-level FCMGr and FCM with different values of $m$.

| Data set | FCM<br>m=1.30 | FCM<br>m=1.55 | FCM<br>m=2.80 | SL FCMGr | ML<br>FCMGr | Levels |
|---|---|---|---|---|---|---|
| Pima | 74.68<br>(80.99) | 75.84<br>(82.00) | 69.61<br>(78.89) | **79.22**<br>(**84.32**) | 74.99<br>(77.94) | 7,9 |
| BUPA | 61.62<br>(55.47) | 56.76<br>(59.89) | 66.18<br>(53.91) | 67.65<br>(61.64) | **68.33**<br>(**65.53**) | 6,7,10 |
| ILPD | 70.17<br>(60.46) | 69.83<br>(58.68) | 69.05<br>(60.43) | 68.10<br>(66.91) | **72.20**<br>(**67.92**) | 8,9 |
| Wisconsin | 92.63<br>(97.81) | 95.77<br>(98.31) | 94.16<br>(96.20) | **96.35**<br>(**98.40**) | 96.05<br>(97.53) | 8,11 |
| Statlog | 72.59<br>(79.40) | 73.33<br>(81.23) | 57.41<br>(56.71) | **79.63**<br>(81.32) | 79.26<br>(**84.49**) | 8,13,10,9 |
| Focality | 85.00<br>(69.25) | **85.80**<br>(69.89) | 85.00<br>(**72.96**) | 84.00<br>(70.12) | 81.80<br>(60.69) | 11,8 |
| Bladder cancer | 63.19<br>(63.99) | 64.04<br>(69.94) | 56.60<br>(48.99) | 68.09<br>(**70.60**) | **69.71**<br>(65.15) | 4,6,11 |

### 3.5.6 Discussion

Three experiments were conducted in this chapter. Firstly, classification performance of FCMGr in terms of accuracy and AUC was compared to 7 FCM-based fuzzy classifiers with different values of fuzziness factor $m$ and a number of clusters that results in best accuracy and AUC. The granularity level that results in best accuracy and AUC was chosen for FCMGr. In this experiment, FCMGr outperformed the FCM-based fuzzy classifiers for 5 out of 7 data sets with margin of 3.7%-7% for accuracy (and 6.6%-0.59% for AUC) approximately.

While the granularity level was chosen experimentally in the first experiment, it was selected automatically in the second experiment. Although the automatic selection did not produce optimal results for most of the data sets, it resulted in better classification performance than the FCM-based fuzzy classifiers for 4 data sets with margin of 5.27%-0.14% for accuracy and 4.59%-0.12% approximately. In addition, the automatic selection resulted in better accuracy than experimental selection for 2 data sets (and better AUC for 3 data sets) due to the fact that the proposed selection method is dynamic. That is, it selects the best granularity level for each training partition independently rather than using fixed granularity level for all training partitions as with the experimental method.

The third experiment involved evaluating multi-level FCMGr-based fuzzy classifiers. Compared to single level FCMGr, multi-level FCMGr resulted in better accuracy in only 2 data sets (and better AUC in 4 data sets).

Results of the three experiments of this chapter show that performance of FCMGr-based fuzzy classifiers depends on the data used. However, based on these results, it can be concluded that FCMGr is effective for the purpose of building fuzzy classifiers for most of the data sets used in this chapter.

## 3.6 Conclusions

In this chapter, a method of information granulation (FCMGr) was proposed to produce information granules from data at multiple levels of granularity. The resulted granules were used in the identification of fuzzy classifier systems using the granulated data. The granulation process was performed using a modified FCM clustering method followed by a coarsening process, i.e. merging higher granularity clusters into lower granularity ones.

FCM is soft clustering method in which an object can belong to more than one cluster with partial inclusion, namely, a fuzzy membership function. The modified FCM starts by choosing initial clusters centres then computes the new fuzzy partition matrix and the updated clusters centres. In addition, the fuzziness factor $m$ is chosen dynamically for each granularity level rather than being fixed prior the algorithm start. Initial clusters for generating granules at a lower granularity level are produced by merging clusters of similar class labels that have weighted distance less than $R_g$ between them.

To evaluate the proposed clustering algorithm, the resulted clusters were used to build a fuzzy classifier for each granulation level. A fuzzy rule was constructed from each cluster where Gaussian membership functions were used to represent the fuzzy variables.

The performance of the classifiers was evaluated using the selected data sets and measured using two classification performance measures: accuracy and AUC. Various data sets were used to train and test the performance of FCMGr. Data were partitioned into training and testing data, 80% and 20% respectively, in 5 training/testing partitionings (P1, P2 ... P5). To evaluate classification performance of the classifiers, both accuracy and AUC were used. Results show that fuzzy systems constructed using FCMGr achieved better accuracy in 5 data sets and better AUC in 6 data sets.

Furthermore, a method of automatically selecting the granulation level that results in better classification performance was proposed. Results show that the automatically selected granulation levels resulted in better accuracy and AUC than the selected FCM classifiers for 4 data sets over a range of number of clusters ($N_C$ and fuzziness factor $m$.

In addition to using one granulation level for the construction of a FIS, multi-levels of the FCMGr were used. The set of all clusters of the selected levels and the concatenation of all the fuzzy c-partition matrices of the selected levels were used to construct the FIS. Results show that multi-level FCMGr has better accuracy than single-level FCMGr for three data sets.

CHAPTER 4

# Granular Computing Approach for Neural Network Classifier Design

In this chapter, a GrC approach for the design of NN classifiers is proposed. The approach is applied to NNs and Adaptive-Network-based Fuzzy Inference Systems (ANFIS). An introduction to NNs is provided in Section 4.1. The proposed approach is described in Section 4.2 and the results of its application on NNs are presented in Section 4.3. In Section 4.4, an experimental comparison of classifiers constructed using the proposed approach and other well-known classifiers is provided. A summary of this chapter is provided in Section 4.5.

## 4.1 Introduction

Neural Networks (NNs) are machine learning models that have the ability to learn complex relationships between input and output. NNs are inspired by the biological structure of human brain and the process of human learning. NNs mimics how brain works by processing data in a network of computational units, called neurons, that perform non-linear transformations on input data.

A neuron consists of a transfer function whose input is the sum of weighted inputs of the neuron. The output $z$ of such a neuron is given by:

$$z = f\left(\sum_{i=1}^{n} w_i x_i\right) \tag{4.1}$$

where:

$n$ is the number of inputs,

$x_i$ is the $i$th input,

and $w_i$ is the weight associated with the $i$th input $x_i$

Neurons are connected together by weighted connections and are arranged into layers. Neurons that are connected directly to the inputs belong to the input layer, while neurons connected directly to the outputs belong to the output layer. Layers located between input and output layers are hidden layers. A NN consists of one input layer, one output layer, and one or more hidden layers.

In terms of how neurons are connected, there are two main types of NNs: feed-forward NNs and recurrent NNs. In feed-forward NNs connections of neurons of a layer are always directed towards a next layer, starting from input layer to output layer. That is, there is no feedback connection in feed-forward NNs. On the other hand, a recurrent NN contains at least one feedback connection connecting a neuron in a layer to a neuron in previous layer. Recurrent NNs are widely used in time series prediction problems. Feed-forward NNs are considered in this chapter.

Figure 4.1 shows the structure of a 4-5-3-2 feed-forward NN, i.e. a NN with 4 neurons in the input layer, 5 neurons in the first hidden layer, 3 neurons in the second hidden layer and 2 neurons in the output layer. In this network, every neuron in a layer is connected to every neuron in the next layer. It should be noticed that the connections in Figure 4.1 are weighted connection, i.e. they involve weight variables that multiply the output of the connected neuron.

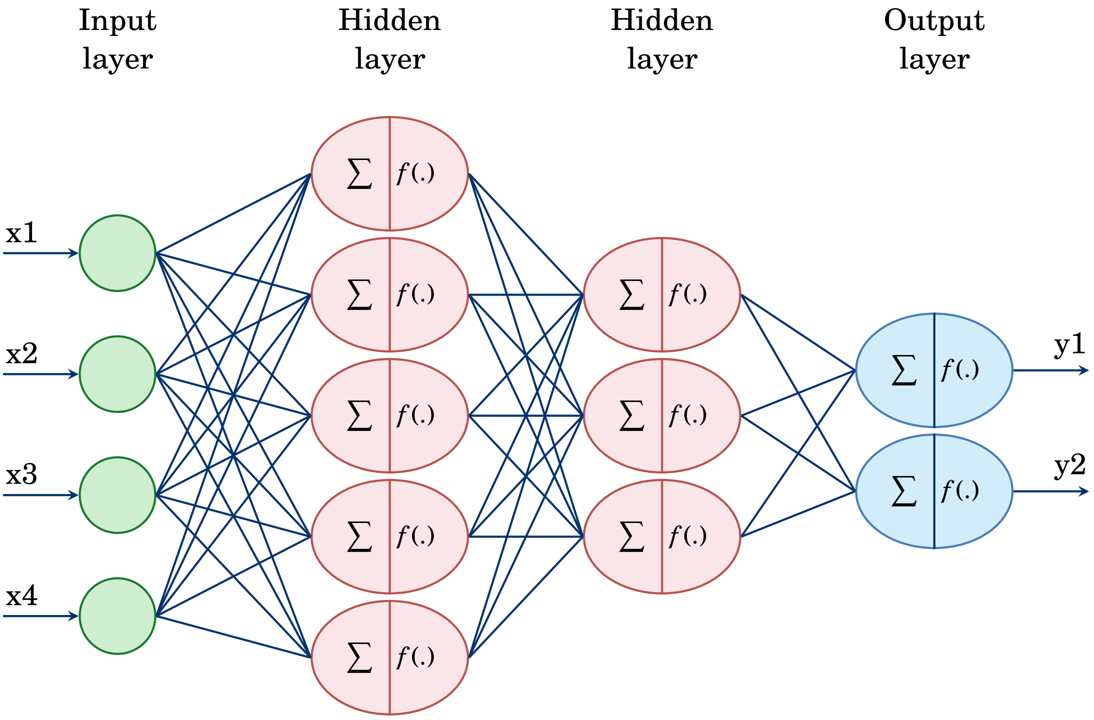

FIGURE 4.1. Multilayer feedforward neural network.

### 4.1.1 Neural Network Learning

NNs have the ability to adapt to its environment by learning, i.e. gaining knowledge from data. This is achieved by adjusting the free parameters (usually the weight variables) of the NN based on stimulation by the environment, e.g. the error of comparing the output of the NN to the desired output. A learning algorithm defines a set of rules that aims to guide this learning process to achieve the desired adaptation.

NNs can be trained by either supervised or unsupervised learning algorithms. In supervised training, desired output is presented to the NN and is used to determine the desired response to a given stimulus. Unlike the supervised-training method, the unsupervised method does not require the desired output. Instead, the learning process is controlled by a set of adaptation rules. One of the popular unsupervised-learning techniques is the self-organizing map (SOM) which is widely used in data visualising.

There are several types of learning algorithms like [159]: error-correction learning and memory-based learning (supervised learning), Hebbian learning and competitive learning (unsupervised learning), and Boltzmann learning (recurrent NNs). In this chapter, an error-correction algorithm, namely back-propagation algorithm, is considered.

In back-propagation algorithm, input is propagated forward through the network to produce the output of the network. Then, the error is calculated by comparing the output of the network with the desired output. The weights are then adjusted by back-propagating the error through the network in order to decrease the error each training cycle [159].

To ensure that the weight adjustments result in decreasing the error, back-propagation algorithm uses the gradient descent method where the adjustment of weights is made in the direction of decrease of the error function. In the gradient descent method, the derivative of the squared error function with respect to the weights of the network is calculated to determine the amount and direction of change of the weights.

The squared error function $E$ is given by:

$$E = \frac{1}{2}\sum_{j=1}^{N_O}\left(t_j - y_j\right)^2 \tag{4.2}$$

where $t_j$ is the target value of output neuron $j$, $y_j$ is the output of output neuron $j$ and $N_O$ is the number of output neurons. Using the chain rule, the derivative of the error function with respect to the weights of the network is given by:

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial o_j}\frac{\partial o_j}{\partial net_j}\frac{\partial net_j}{\partial w_{ij}} = \delta_j o_i \tag{4.3}$$

where $w_{ij}$ is the weight of connection going from neuron $i$ to neuron $j$, $o_j$ is the output of neuron $j$, $net_j$ is the weighted sum of all inputs of neuron $j$ and:

$$\delta_j = \frac{\partial E}{\partial o_j}\frac{\partial o_j}{\partial net_j} = \begin{cases} \left(o_j - t_j\right) f'\left(net_j\right) & \text{if } j \text{ is an output neuron} \\ \left(\sum_{l\in L}\delta_l w_{jl}\right) f'\left(net_j\right) & \text{if } j \text{ is an inner neuron} \end{cases} \tag{4.4}$$

where $f'(.)$ is the derivative of the transfer function of the neuron and $L$ is the set of neurons that receive input from neuron $j$. So, the weights are updated by the following:

$$\Delta w_{ij} = -\eta \frac{\partial E}{\partial w_{ij}} = \begin{cases} -\eta o_i \left(o_j - t_j\right) f'\left(net_j\right) & \text{if } j \text{ is an output neuron} \\ -\eta o_i \left(\sum_{l \in L} \delta_l w_{jl}\right) f'\left(net_j\right) & \text{if } j \text{ is an inner neuron} \end{cases} \tag{4.5}$$

where $\eta$ is the learning rate which is a scale factor that determines the step size of updating the weights. Increasing the value of $\eta$ increases the learning speed but may result in missing the optimal solution (i.e. the local minimum).

Two widely used transfer functions are the logistic function (sigmoid function) and hyperbolic tangent function given by (respectively):

$$sig\left(net_j\right) = \frac{1}{1+e^{-net_j}} \tag{4.6}$$

$$tanh\left(net_j\right) = \frac{e^{net_j} - e^{-net_j}}{e^{net_j} + e^{-net_j}} \tag{4.7}$$

The derivatives of these two functions are given by (respectively):

$$sig'\left(net_j\right) = sig\left(net_j\right)\left(1 - sig\left(net_j\right)\right) = o_j\left(1 - o_j\right) \tag{4.8}$$

$$tanh'\left(net_j\right) = \left(1 - tanh^2\left(net_j\right)\right) = \left(1 - o_j^2\right) \tag{4.9}$$

Equation 4.5 clearly shows the back-propagation process: to update the weight $w_{ij}$ of a layer other than the output layer, $\delta_l$ has to be computed for all the neurons receiving input from neuron $i$, i.e. neurons in the next layer. Thus, starting from the output layer, $\delta_j$ is computed using equation 4.4 then $\delta_j$ of neurons in previous layers can be computed, and so on.

Weights can be updated in two different ways: incremental (or online) and batch learning. In incremental learning, weights are updated after each training sample is presented to the network while in batch learning the weights are updated after all training samples have been presented.

An epoch of training is completed when all training sample are used to update the weights of the NN in either online or batch mode. Then, the training process is repeated until maximum number of epochs is reached or minimum error is achieved.

The gradient descent method is easy to implement but it has some drawbacks such as slow learning and getting stuck in local minima. Thus, there have been several methods to provide more efficient solutions to the problem of error minimization. In Newton's method, the second derivative of the objective function is calculated providing a more efficient solution with faster convergence [160]. In this method, the weights adjustment is given by:

$$\Delta w_{ij} = -H^{-1}\frac{\partial E}{\partial w_{ij}} \tag{4.10}$$

where $H$ is the Hessian matrix of the second-order partial derivatives of the error function $E$. However, computing $H$ in every iteration is computationally expensive as the computational complexity and memory requirements are proportional to the square of the number of weights. Therefore, some algorithms, namely Quasi-Newton algorithms, propose different ways to approximate $H$ by less computationally expensive methods. In Levenberg-Marquardt method, which is often used for regression applications, the performance function is assumed to have a quadratic form in a region around the current search point, called trusted region. Thus, $H$ is approximated by the Jacobian matrix $J$ that contains all first-order partial derivatives of the error function with respect to the weights and the weight updates are given by [161]:

$$\Delta w_{ij} = -\left[J^T J + \lambda I\right]^{-1} J^T e \tag{4.11}$$

where $e$ is a vector of network errors, and $\lambda$ is a constant that is increased when the performance function increases and decreased when the performance function decreases.

Another alternative to the gradient descent method is the conjugate gradient method where weights are updated in the conjugate directions, which provides more efficient convergence without the need to compute the Hessian matrix. The new search direction $p_k$ is determined by combining the new gradient descent direction with the previous search direction $p_{k-1}$ [161]:

$$p_k = -\frac{\partial E}{\partial w_{ij} + \beta p_{k-1}} \tag{4.12}$$

Scaled conjugate gradient method combines the trust region approach (similar to Levenberg-Marquardt method) with the conjugate gradient algorithm.

### 4.1.2 Generalisation and Overfitting

The performance of a NN is measured by its ability to generalise, i.e. how well it perform on test data that is not used during training. Thus, the aim of training is to find the training parameters that result in best generalisation. However, overtraining the NN may results in memorising the training data rather than generalising for new data. Therefore, training data is usually divided into two sets: training data set that is used to train the NN and validation data set to monitor the generalisation performance of the NN. Using one of the learning methods described in 4.1.1, the error of the NN decreases monotonically during training. However, its generalisation ability stops to increase at some point of training, i.e. when the NN starts to be overtrained. Validation data set is used to stop the learning at this point by monitoring the performance of the NN with the validation data. Once the validation error starts to increase, training is stopped as illustrated in Figure 4.2.

Thus, a given data set is divided into three sets: training set, validation set, and test set.

### 4.1.3 Number of Layers and Neurons

Although there are no formal rules for determining the number of neurons in the hidden layers of a NN, there are some rules of thumb that are drawn from practice. Some of these rules of thumb are [162]:

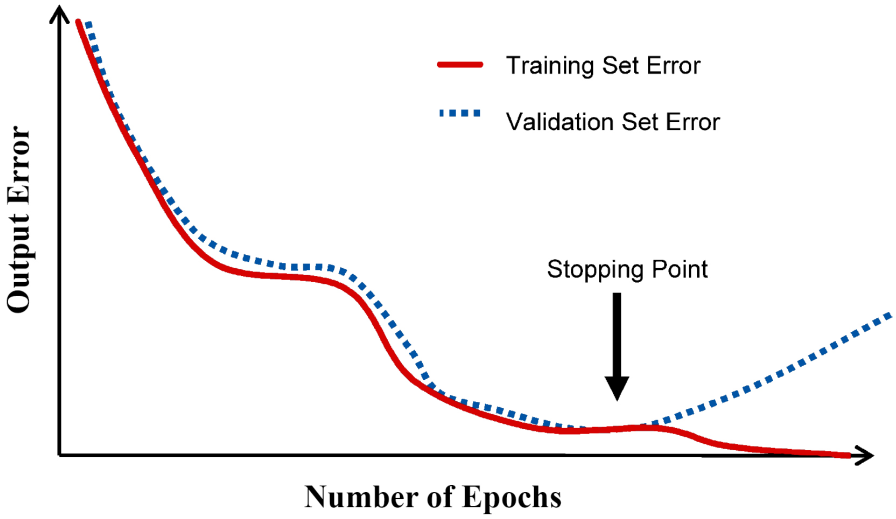

FIGURE 4.2. Training set and validation set error as a function of epoch [160].

- The number of hidden neurons should be between the number of input variables and the number of outputs.

- The number of hidden neurons should be 2/3 the number of input variables plus the number of outputs.

- The number of hidden neurons should be less than twice the number of input variables.

The first and second rules mean that, for single output NNs, the number of hidden neurons is less than the number of input variables. In general, the number of hidden neurons should be minimised so that the number of learning variables (weights) is minimised and, as a result, training time and computational cost are minimised. Therefore, it is suggested that the ratio of hidden neurons to number of inputs is decreased for number of inputs larger than 5 [160].

In addition, it is suggested that a single hidden layer is sufficient for most applications [160, 163].

## 4.2 Proposed Granular Computing Approach

In this section, a GrC approach for the design of NN classifiers is presented. The proposed approach, named GrC approach for NN Classifier design (GrCNNC), is based on two principles of GrC:

1. In GrC paradigm, a given task can be split into smaller and easier subtasks to achieve a more efficient solution.
2. Different levels of granularity provide different levels of information specificity/abstraction.

The first principle suggests that training a NN for each granule (cluster) may results in better performance than training a single NN for all data samples. Since the aim of training is to achieve best generalisation rather than memorising, using training data at a a certain level of abstraction may results in better performance than using high-granularity training data, according to the second principle.

To employ these principles, the following approach is proposed. Firstly, training data is granulated using the granulation method presented in Section 3.3, namely FCMGr. The result of the granulation is a set of granulation levels each consists of a fuzzy partitioning comprised of number of granules (clusters). The fuzzy partitioning is represented by the cluster centres vector $\hat{V}$ and the fuzzy c-partition matrix $\hat{U}$. Secondly, a granulation level is selected and a NN is constructed and trained for each granule. Thus, the overall classification task is divided into smaller subtasks of predicting the inclusion of a data sample to different granules. The granulation level that results in best classification performance is the level that provides the best abstraction level and it depends on the data.

Training data at a given level is represented by the distance $d(.,.)$ of each data sample to each of the granules at that level. So, the classification problem is transformed from mapping input space to class space to mapping distance space to inclusion space ($\hat{U}$). The final output is computed from the inclusion score results and the class of each granule. Target values for training the NNs are the fuzzy c-partition matrix values $\hat{u}_{ij}$ corresponding to the training samples and the granules.

It should be noticed that all the NNs share the same inputs, so, to reduce complexity and computational cost, they can be combined in a single NN with number of outputs equal to the number of granules.

To classify new data (test data), the distances of the data sample to each of the granules are computed first. Then inclusion scores are computed using the NN and used with granules class labels to compute the final output. Thus, the cluster centres vector $\hat{V}$ of training data is part of the classifier since it is used to compute the distance and final output while the fuzzy c-partition matrix $\hat{U}$ is used in training phase only.

Following is the algorithm of the GrCNNC training phase:

1. Perform granulation using FCMGr with the training and validation data sets.
2. Select a granulation level.
3. For each training sample $x_i^t, i = 1,2,\dots n$ and granule $v_j, j = 1,2,\dots c$, compute training distance matrix $D^t$ where $d_{ij}^t = d(x_i^t, v_j)$
4. For each validation sample $x_i^l, i = 1,2,\dots m$ and granule $v_j, j = 1,2,\dots c$, compute validation distance matrix $D^l$ where $d_{ij}^l = d(x_i^l, v_j)$
5. Construct a NN with $c$ inputs and $c$ outputs and train it with $D^t$ as input and $\hat{U}$ as the target output, where $x_i \in$ training set.
6. Monitor the performance of the NN with $D^l$ as input and $\hat{U}$ as target output, where $x_i \in$ validation set.

For the testing phase, the following steps are performed:

1. For the selected granulation level, compute the distance vector $D^s$ where $d_{ij}^s = d(x_i^s, v_j)$ for the test sample$x_i^s$ and granule $v_j, j = 1,2,\dots c$.
2. Using the trained NNs evaluate the inclusion scores vector $\hat{U}_i^s$ using $D^s$ as input.

3. Compute the final output $Z_i$ (class label) using the equation:

$$Z_i(\hat{U}_i^s, \hat{V}) = \frac{1}{c}\sum_{j=1}^{c} \hat{u}_{ij}^s \hat{v}_j(dim) \tag{4.13}$$

   where $dim$ is the dimension of $\hat{V}$, i.e. $\hat{v}_j(dim)$ is the class label of the cluster $\hat{v}_j$.

Block diagrams of the GrCNNC for training phase, validation and test phase are shown in Figures 4.3-4.5 where $E_v$ is the error calculated for the validation data set using Equation 4.2.

Since both the number of inputs and the number of outputs of the NN equal to the number of granules in the selected granulation level, the complexity and computational cost of GrCNNC depends on the number of granules. Choosing a granulation level with number of granules more than the number of input variables means that GrCNNC results in more complex and more computationally expensive system than standard NN. On the other hand, if the number of granules is less than the number of input variables significantly, the GrCNNC will result in less complex and less computationally expensive system than standard NN.

So far, the GrCNNC algorithm assumes using a single granulation level in training, validating and testing the NN. However, multiple levels can be used in the same algorithm. When more than one granulation level are used, cluster centres vector $\hat{V}$ and fuzzy c-partition matrix $\hat{U}$ used in the algorithm comprise of the corresponding vectors and matrices of all the selected levels.

### 4.2.1 Interpretability of GrCNNC

One desired property of a classifier is the ability to provide knowledge about the classification process and its reasoning in order to provide better understanding of the problem and allow for more straightforward generalisation based on the provided knowledge. However, it is difficult to extract this knowledge from NNs because of its distributive nature where a problem is modelled by large number of variables (weights and transfer functions). Therefore, a NN is considered to be a black box.

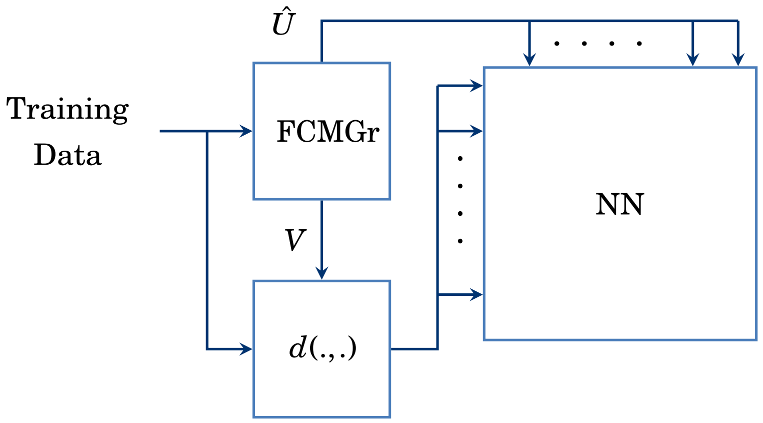

FIGURE 4.3. Block diagram of the GrCNNC for training phase.

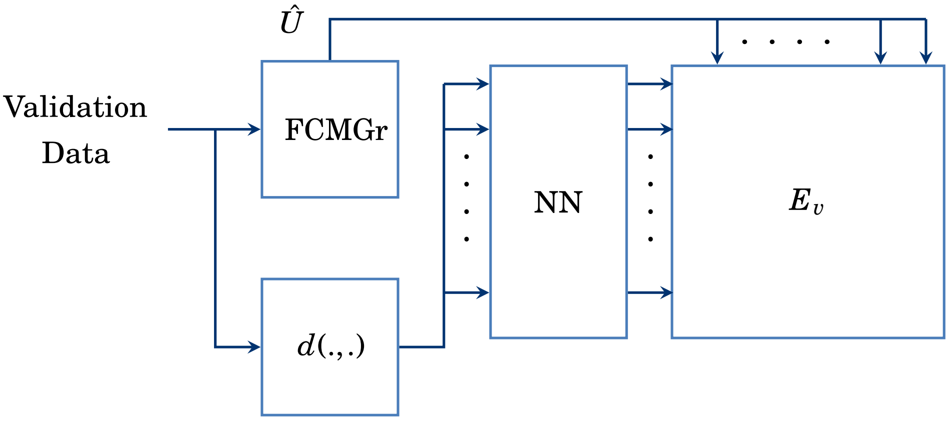

FIGURE 4.4. Block diagram of the GrCNNC for validation.

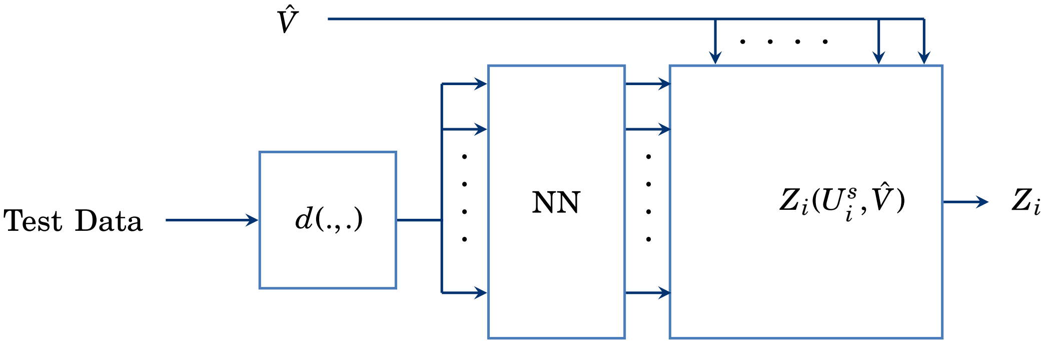

FIGURE 4.5. Block diagram of the GrCNNC for test phase.

There have been several attempts to address the issue of the interpretability of NNs. One of these attempts is to extract rules form NNs [164, 165, 166].

Although GrCNNC dose not change the structure of NN, GrCNNC increases the interpretability of the NN by training it using information granules rather than raw data. Information granules can be viewed as classification rules, thus, training the NN using information granules can be viewed as training the NN to learn these rules. In other words, in GrCNNC, the classification problem is broken down into the modelling of classification rules represented by the information granules. The number of granules in the selected granulation level(s) represent the number of the rules, and the centre of each granule determine the rule while a fuzzy c-partition value represent the degree to which a rule is applicable to a data sample.

### 4.2.2 Application of GrCNNC in ANFIS

Adaptive-Network-based Fuzzy Inference Systems (ANFIS) is a structure that combines the learning ability of NNs and the soft reasoning of FISs. In ANFIS, the parameters of both the antecedent and consequent parts of the fuzzy rules are adjusted or tuned using adaptive network learning [167]. ANFIS is restricted to the Sugeno-type FIS where the parameters of the membership functions in the antecedent part and the crisp functions of the consequent part are adjusted using back-propagation algorithm or a combination of back-propagation and least squares estimate algorithms.

GrCNNC can be applied in training the ANFISin a way similar to that of the NN. However, ANFIS computes the final output (class label) directly, therefore, the last step of the GrCNNC algorithm is not performed. Thus, ANFIS is trained using the distance matrix $D^t$ and the target class labels instead of the inclusion scores. As with NN, the number of ANFIS inputs using GrCNNC is the number of granules of the selected levels. It should be noticed that the number of fuzzy rules of ANFIS is independent of the selected granulation level (or number of granules) allowing for less complex system even with higher granularity levels. As with most applications, there are no rules for determining the number of fuzzy rules of ANFIS, instead, it is determined experimentally by choosing the number that results in best results for a specific data.

## 4.3 Experimental Results

### 4.3.1 Single Level GrCNNC

The GrCNNC is implemented using MATLAB R2016a. Since the outputs of the NN of the GrCNNC are scores (between 0-1) rather than class labels, the learning algorithm used is the Levenberg-Marquardt (discussed in 4.1.1) because its widely use in regression applications. The transfer function of the neurons is hyperbolic tangent function given by Equation 4.7. The granulation level that results in best performance is empirically selected for each data set.

The data sets presented in 3.4.1 are used to evaluate the GrCNNC. Each data set is divided into 3 parts: training (60%), validation (20%) and test (20%). 5-fold cross validation is used, therefore, each classifier is tested 5 times with different test partition each time. Each of the 5 fold tests is run 10 times and the mean and Standard Deviation (STD) of the accuracy and mean of AUC are calculated. The test partition is fixed for all the 10 runs, however, training/validation sampling is changed randomly.

The classification results of the data sets using GrCNNC compared to those of standard NN are shown in Table 4.1. By referring to the rules of thumb in 4.1.3, and for the range of number of inputs of the data sets used, single hidden layer with 8 neurons is found to result in best performance for both the standard NNs and the GrCNNC. Referring to Table 3.1, the range of number of inputs for the data sets is 6 to 13, thus, for 8 hidden neurons, the range of the ratio of hidden neurons to number of inputs is approximately $\frac{4}{3}$ to $\frac{2}{3}$.

The results of Table 4.1 (where highest results are in bold) show that GrCNNC has higher accuracy for all the data sets. The improvement in accuracy is accompanied by higher AUC for all datasets except ILPD, which indicates that the improvement in accuracy does not come at the cost of lower specificity or sensitivity. However, significant improvements (i.e. improvements that are higher than the STD values of the accuracy of the NNs) achieved in only four data sets, namely: BUPA, ILPD, Statlog and Bladder cancer.

In addition, the results show that GrCNNC has lower STD for all data sets except one (ILPD) which indicates that GrCNNC is more invariant to the change in initial weights and training samples.

Similarly, training accuracy and AUC are obtained, as shown in Table 4.2. From the table, GrCNNC has better training accuracy for 5 data sets and higher AUC for 4 data sets. As with test results, STD of training accuracy is lower for GrCNNC than standard NN.

TABLE 4.1. Test Accuracy (%), STD and AUC (%) of NN and GrCNNC.

| Data sets | Neural Network | | | GrCNNC | | | | Improvement | |
|---|---|---|---|---|---|---|---|---|---|
| | Acc. | STD | AUC | Acc. | STD | AUC | Level | Acc. | AUC |
| Pima | 75.42 | 0.76 | 82.97 | **76.18** | 0.62 | **83.43** | 8 | 0.76 | 0.45 |
| BUPA | 64.70 | 2.15 | 67.27 | **69.28** | 1.36 | **72.26** | 8 | 4.58 | 4.99 |
| ILPD | 69.85 | 0.63 | **66.80** | **71.90** | 0.88 | 63.43 | 10 | 2.05 | -3.37 |
| Wisconsin | 95.98 | 0.62 | 99.32 | **96.11** | 0.25 | **99.36** | 9 | 0.13 | 0.04 |
| Statlog | 80.89 | 2.22 | 88.83 | **83.37** | 0.41 | **89.01** | 13 | 2.48 | 0.18 |
| Focality | 80.96 | 0.51 | 59.15 | **81.12** | 0.33 | **59.80** | 10 | 0.16 | 0.65 |
| Bladder Cancer | 59.87 | 2.16 | 62.85 | **63.70** | 1.66 | **66.73** | 11 | 3.83 | 3.88 |

TABLE 4.2. Training Accuracy (%), STD and AUC (%) of NN and GrCNNC.

| Data sets | Neural Network | | | GrCNNC | | | | Improvement | |
|---|---|---|---|---|---|---|---|---|---|
| | Acc. | STD | AUC | Acc. | STD | AUC | Level | Acc. | AUC |
| Pima | 78.83 | 0.32 | 84.58 | **79.63** | 0.23 | **85.78** | 8 | 0.80 | 1.20 |
| BUPA | 73.89 | 1.96 | 76.47 | **75.98** | 0.70 | **79.90** | 8 | 2.09 | 3.43 |
| ILPD | **73.40** | 0.35 | **74.44** | 72.67 | 0.35 | 68.58 | 10 | -0.73 | -5.86 |
| Wisconsin | 97.95 | 0.23 | **99.65** | **97.99** | 0.09 | 99.46 | 9 | 0.03 | -0.19 |
| Statlog | **86.59** | 0.79 | **92.48** | 84.82 | 0.27 | 89.81 | 13 | -1.77 | -2.67 |
| Focality | 81.97 | 0.09 | 61.83 | **82.05** | 0.15 | **63.98** | 10 | 0.08 | 2.14 |
| Bladder Cancer | 72.51 | 1.39 | 76.83 | **72.67** | 0.64 | **77.98** | 11 | 0.16 | 1.15 |

The selected granulation level for each data set is shown in the table where higher levels have lower granularity (less number of granules). The number and centres of granules used in the GrCNNC for a data set can be known from the selected granulation level, hence, the rules that describe the classification process can be determined as discussed in 4.2.1. For instance, Table 4.1 shows that the selected granulation level for BUPA data set is the 8th level. The number of granules in level 8 for BUPA data set (P1) is 9 (Table 3.3), thus, the classification process can be described by 9 rules. The centre of first granule is:

$$\hat{v}_1 = (0.64, 0.34, 0.12, 0.22, 0.07, 0.07, 1.00)$$

Therefore, the first rule that describe the classification process for BUPA data set is (refer to 3.4.1 for the description of the attributes of BUPA data set):

$$\text{IF } MCV = 0.64 \text{ AND } Alkphos = 0.34 \text{ AND } SGPT = 0.12 \text{ AND } SGOT = 0.22$$
$$\text{AND } GAMMAGT = 0.07 \text{ AND } drinks = 0.07 \text{ THEN } class = 1$$

### 4.3.2 Multi-level GrCNNC

In this section, the results of using more than one granulation level is reported. Table 4.3 shows the classification results for the multi-level GrCNNC compared to those of NN with highest results in bold. As with single level GrCNNC, multi-level GrCNNC has better accuracy than NN for all data sets with higher AUC for all data sets except Wisconsin data set. Significant improvement is obtained for BUPA and bladder cancer data sets. However, compared to the accuracy improvement of single level GrCNNC, multi-level GrCNNC has better improvement in only five data sets, as shown in Figure 4.6.

In addition, Table 4.4 shows the training accuracy and AUC where multi-level GrCNNC has better classification results for most of the data sets.

As compared to the single level GrCNNC, the multi-level GrCNNC has higher AUC improvement over NN for all the data sets except Wisconsin, as shown in Figure 4.7.

Figures 4.8-4.14 show the ROC curves of the single level GrCNNC, multi-level GrCNNC and NN for all the data sets. The curves are drawn for a single run of one of the 5 test partitions. From the figures, noticeable improvement of the ROC curves of both single level GrCNNC and multi-level GrCNNC can be seen, especially for the BUPA, Statlog and bladder cancer data sets.

Table 4.3. Test Accuracy (%), STD and AUC (%) of NN and multi-level GrCNNC.

| Data sets | Neural Network | | | Multi-GrCNNC | | | | Improvement | |
|---|---|---|---|---|---|---|---|---|---|
| | Acc. | STD | AUC | Acc. | STD | AUC | Level | Acc. | AUC |
| Pima | 75.42 | 0.75 | 82.97 | **76.58** | 0.71 | **83.60** | 8,9 | 1.16 | 0.63 |
| BUPA | 64.70 | 2.15 | 67.27 | **70.51** | 1.59 | **73.53** | 7,8 | 5.81 | 6.26 |
| ILPD | 69.85 | 0.63 | 66.80 | **70.00** | 1.01 | **67.60** | 8,10 | 0.15 | 0.8 |
| Wisconsin | 95.98 | 0.62 | **99.32** | **96.51** | 0.51 | 99.16 | 9,10 | 0.53 | -0.16 |
| Statlog | 80.89 | 2.22 | 88.83 | **82.59** | 0.99 | **89.80** | 11,13 | 1.70 | 0.97 |
| Focality | 80.96 | 0.51 | 59.15 | **81.34** | 0.47 | **60.45** | 10,11 | 0.38 | 1.30 |
| Bladder Cancer | 59.87 | 2.16 | 62.85 | **64.93** | 1.75 | **67.04** | 11,12,13 | 5.06 | 4.19 |

Table 4.4. Training Accuracy (%), STD and AUC (%) of NN and multi-level GrCNNC.

| Data sets | Neural Network | | | Multi-GrCNNC | | | | Improvement | |
|---|---|---|---|---|---|---|---|---|---|
| | Acc. | STD | AUC | Acc. | STD | AUC | Level | Acc. | AUC |
| Pima | 78.83 | 0.32 | 84.58 | **79.99** | 0.24 | **86.26** | 8,9 | 1.16 | 1.68 |
| BUPA | 73.89 | 1.96 | 76.47 | **76.94** | 0.71 | **80.96** | 7,8 | 3.06 | 4.50 |
| ILPD | 73.40 | 0.35 | 74.44 | **73.80** | 0.37 | **74.65** | 8,10 | 0.40 | 0.21 |
| Wisconsin | 97.95 | 0.23 | **99.65** | **98.09** | 0.07 | 99.47 | 9,10 | 0.13 | -0.18 |
| Statlog | **86.59** | 0.79 | **92.48** | 85.31 | 0.32 | 91.47 | 11,13 | -1.28 | -1.01 |
| Focality | 81.97 | 0.09 | 61.83 | **82.01** | 0.14 | **64.20** | 10,11 | 0.04 | 2.36 |
| Bladder Cancer | 72.51 | 1.39 | 76.83 | **72.92** | 0.56 | **78.33** | 11,12,13 | 0.40 | 1.49 |

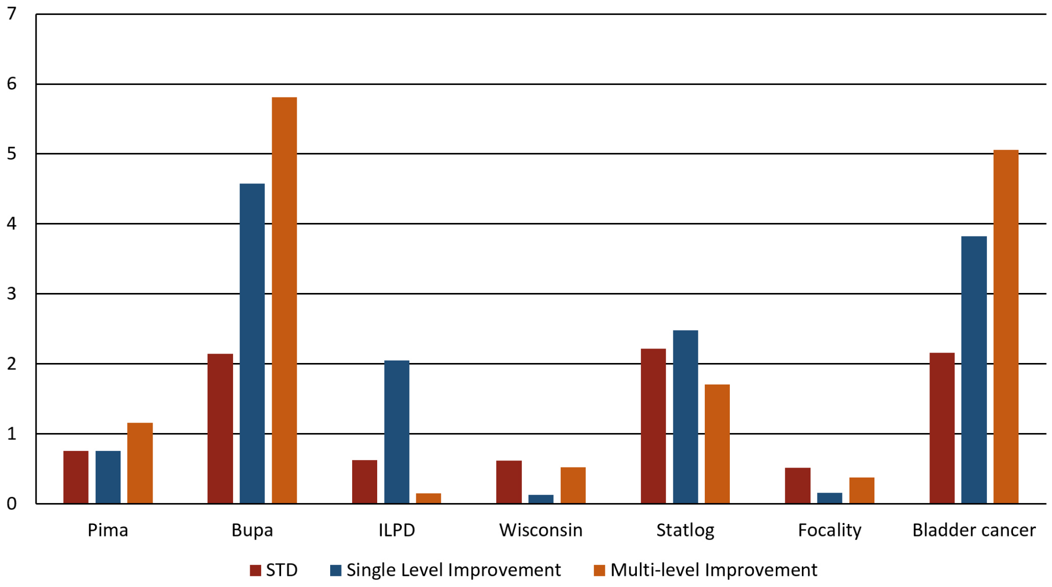

FIGURE 4.6. Accuracy improvement of single level GrCNNC vs. accuracy improvement of multi-level GrCNNC vs. standard deviation of NN accuracy.

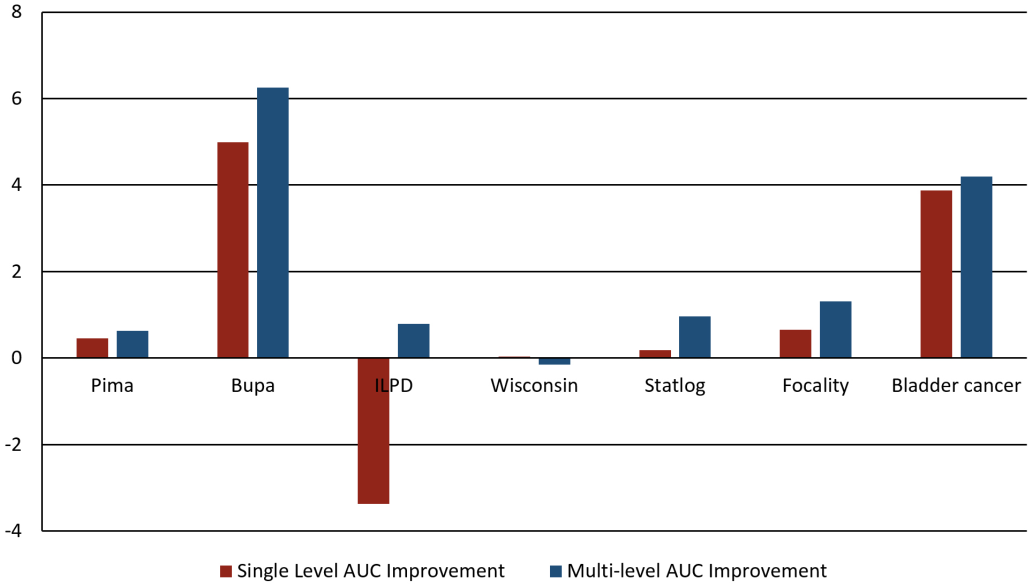

FIGURE 4.7. AUC improvement of single level GrCNNC vs. AUC improvement of multi-level GrCNNC.

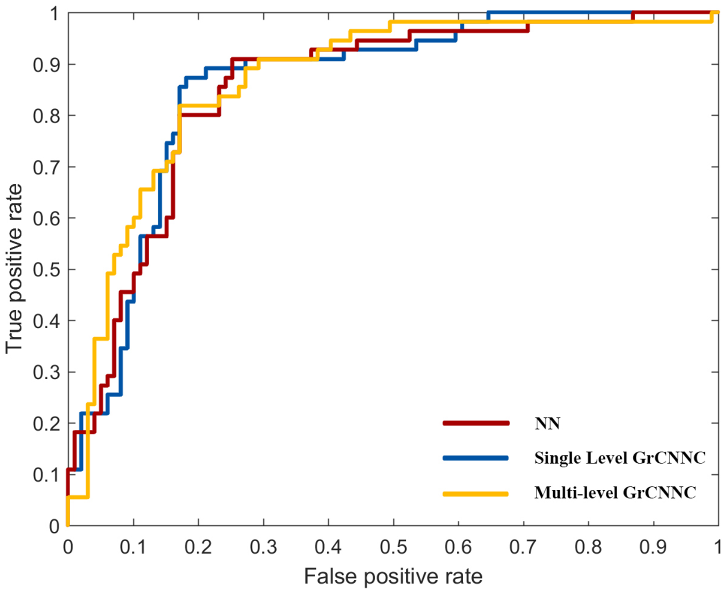

FIGURE 4.8. ROC curve of NN, single level GrCNNC and multi-level GrCNNC for Pima data set.

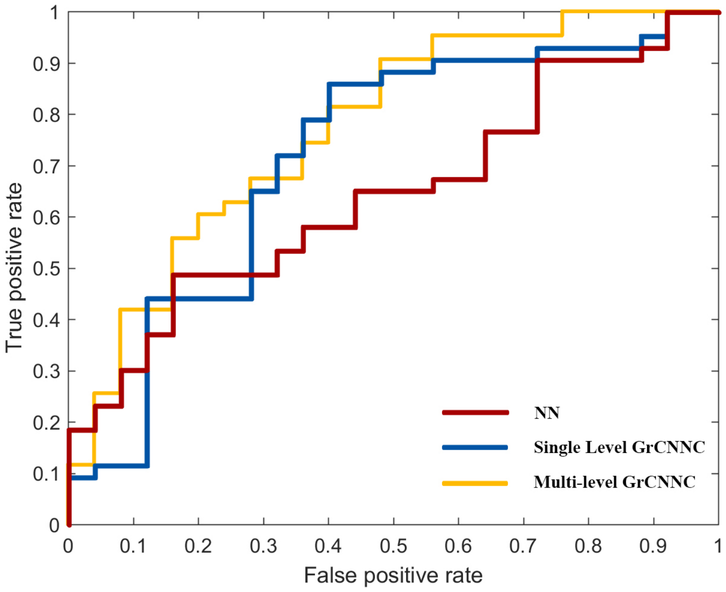

FIGURE 4.9. ROC curve of NN, single level GrCNNC and multi-level GrCNNC for BUPA data set.

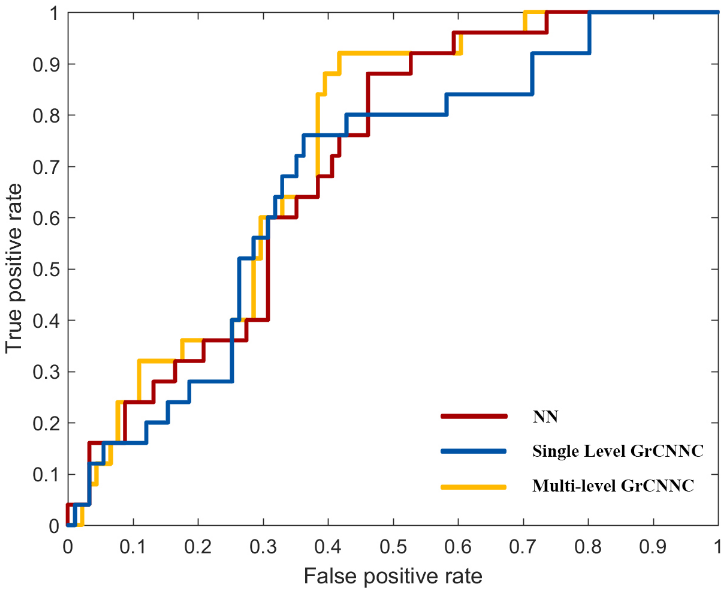

FIGURE 4.10. ROC curve of NN, single level GrCNNC and multi-level GrCNNC for ILPD data set.

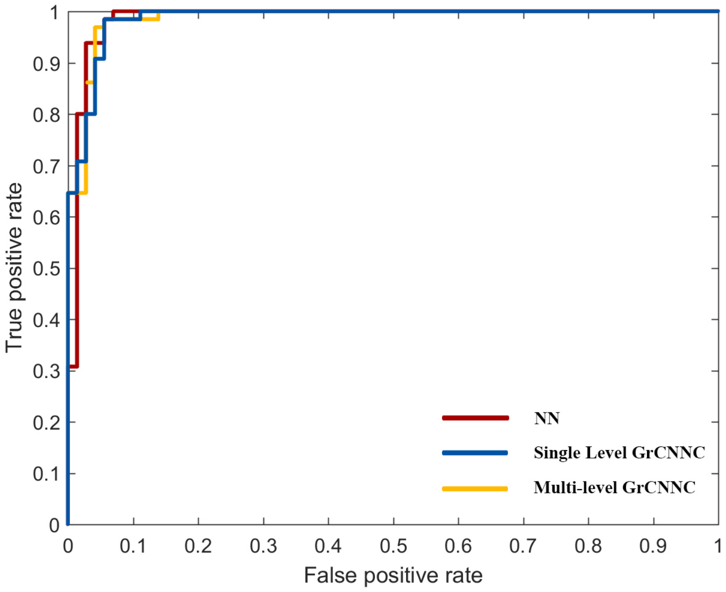

FIGURE 4.11. ROC curve of NN, single level GrCNNC and multi-level GrCNNC for Wisconsin data set.

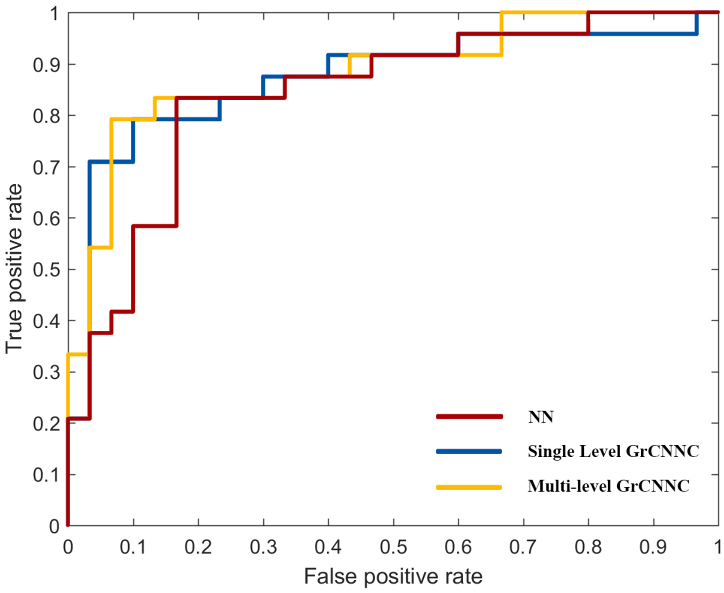

FIGURE 4.12. ROC curve of NN, single level GrCNNC and multi-level GrCNNC for Statlog data set.

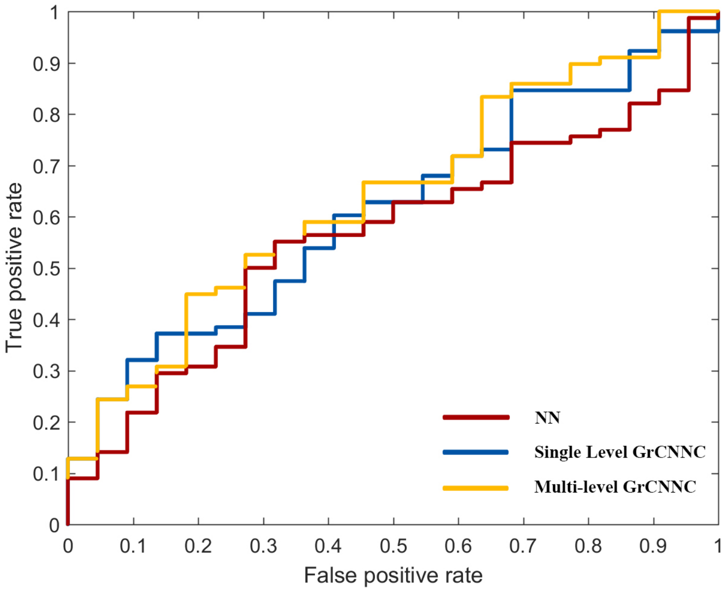

FIGURE 4.13. ROC curve of NN, single level GrCNNC and multi-level GrCNNC for focality data set.

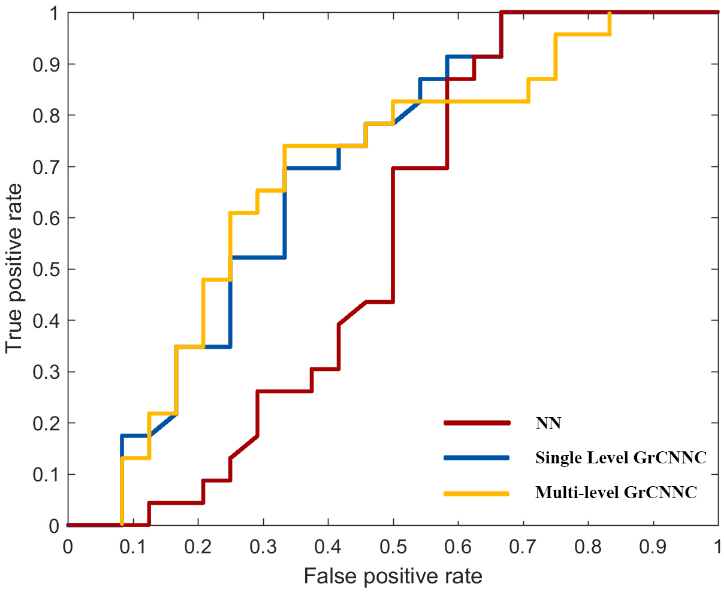

FIGURE 4.14. ROC curve of NN, single level GrCNNC and multi-level GrCNNC for bladder cancer data set.

### 4.3.3 ANFIS with GrCNNC

Test results of using multi-level GrCNNC with ANFIS are shown in Table 4.5. The results are compared to standard ANFIS. Accuracy and AUC of training and test are shown in the table with best results in bold. As mentioned in Section 4.2.2, the number of fuzzy rules of an ANFIS system is independent of the selected number of granules. So, both models are based on FISs which are generated using FCM with 2 clusters to decrease complexity, as there are no rules that specify the number of fuzzy rules. The GrCNNC model is trained using the granulated data, while the standard ANFIS is trained using the original training data. As with the previous experiments, each data set is divided into 3 parts: training (60%), validation (20%) and test (20%). Each classifier is tested 5 times with different test partition each time (5-fold cross validation).

As shown in the table, GrCNNC has improved the test accuracy of ANFIS for 5 out of 7 data sets with improved AUC with the highest improvement for bladder cancer data set. In addition, test AUC is improved for 6 data sets with significant improvement for focality and bladder cancer data sets.

Table 4.6 shows the training results. High improvements in training results are obtained for BUPA, focality and bladder cancer data sets.

TABLE 4.5. Test Accuracy (%) and AUC (%) of ANFIS and multi-level GrCNNC ANFIS.

| Data sets | ANFIS | | ML GrCNNC ANFIS | | | Improvement | |
|---|---|---|---|---|---|---|---|
| | Acc. | AUC | Acc. | AUC | Levels | Acc. | AUC |
| Pima | 75.01 | 81.96 | **77.21** | **82.92** | 1,3 | 2.2 | 0.96 |
| BUPA | 71.25 | 75.63 | **72.05** | **75.73** | 4,5 | 0.8 | 0.10 |
| ILPD | 70.00 | 70.58 | **70.64** | **71.48** | 1,2 | 0.64 | 0.90 |
| Wisconsin | 95.05 | 99.31 | **96.20** | **99.59** | 1,2,5 | 1.15 | 0.28 |
| Statlog | **82.96** | **89.64** | 81.11 | 88.33 | 1,2 | -1.85 | -1.31 |
| Focality | **81.86** | 50.23 | 81.00 | **59.49** | 1,2 | -0.86 | 9.26 |
| Bladder Cancer | 57.33 | 58.76 | **63.50** | **63.25** | 4,10 | 6.17 | 4.49 |

TABLE 4.6. Training Accuracy (%) and AUC (%) of ANFIS and multi-level GrCNNC ANFIS.

| Data sets | ANFIS | | ML GrCNNC ANFIS | | | Improvement | |
|---|---|---|---|---|---|---|---|
| | Acc. | AUC | Acc. | AUC | Levels | Acc. | AUC |
| Pima | 79.99 | 85.26 | **80.03** | **85.46** | 1,3 | 0.04 | 0.20 |
| BUPA | 77.32 | 82.24 | **82.51** | **87.23** | 4,5 | 5.19 | 4.99 |
| ILPD | **75.07** | **77.76** | 73.63 | 75.21 | 1,2 | -1.43 | -2.55 |
| Wisconsin | 97.76 | 99.62 | **98.83** | **99.89** | 1,2,5 | 1.07 | 0.27 |
| Statlog | **88.89** | **95.26** | 85.31 | 91.89 | 1,2 | -3.58 | -3.37 |
| Focality | 81.93 | 51.16 | **82.00** | **67.02** | 1,2 | 0.07 | 15.86 |
| Bladder Cancer | 77.35 | 81.76 | **93.09** | **98.01** | 4,10 | 15.74 | 16.25 |

### 4.3.4 Results Summary

So far in this chapter, three experiments were conducted. The first two experiments investigated the performance of single level GrCNNC and multi-level GrCNNC respectively. Single level GrCNNC resulted in better accuracy and AUC than NN for 6 data sets with accuracy improvement not less than STD for 4 of them. Improvement in training performance is less than that in test performance which suggests that the improvement in test performance is due to increased generalisation of single level GrCNNC. In addition, the results of the first experiment are used to demonstrate the interpretability of the GrCNNC.

Results of the second experiment show that using multi-level GrCNNC had improved AUC and accuracy over single level GrCNNC for 5 and 6 data sets respectively. The third experiment was conducted to evaluate the use of multi-level GrCNNC with ANFIS. Multi-level GrCNNC resulted in improved accuracy and AUC of ANFIS for 5 data sets in both test and training.

To sum up, based on the results of these experiments, it can be concluded that GrCNNC has performed relatively well compared to standard NNs.

## 4.4 Experimental Comparison

To compare the performance of GrCNNC-based NN and ANFIS to other classifiers, the classification results obtained in previous section are compared to those of 9 classifiers. The classifiers are implemented using MATLAB R2016a and the same data sets and experimental set-up (5-fold cross validation) of the previous section are used. Validation partition is used wherever validation is applicable (i.e. NN and ANFIS). In addition to the standard NN and ANFIS used in previous section, 7 classifiers are used: Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), Naive Bayes Classifier (NBC), Decision Tree (Dtree), SVM with linear kernel function (SVM Linear), SVM with Gaussian Radial Basis Function (RBF) kernel (SVM RBF) and SVM with polynomial kernel (SVM polynomial).

Test accuracy results of the selected classifiers are show in Table 4.7 with highest results in bold. The last 3 classifiers are the single-level and multi-level GrCNNC-based NN (SL GrCNNC and ML GrCNNC) classifier and multi-level GrCNNC-based ANFIS (GrCNNC ANFIS) classifier. In 5 out of 7 data sets, one of the 3 GrCNNC-based classifiers has the best results. GrCNNC ANFIS has the best results in 3 data sets while SL GrCNNC has the best results in 2 data sets. SVM Linear has the best result for bladder cancer data set while standard ANFIS has the highest result for Focality data set. However, in terms of AUC, GrCNNC ANFIS and ML GrCNNC have the best results in only 5 data sets while SVM linear has the best results for the remaining 2 data sets, as shown in Table 4.8.

To visualise the relative performance difference between the classifiers, Figures 4.15 and 4.16 show the rank of each classifier for each data set in terms of accuracy and AUC respectively, with rank 1 being the best classifier and rank 12 is the worst. The size of the bubbles reflect the relative normalised difference in performance compared to the worst performance for each data set. For example, the 3 GrCNNC-based classifiers have ranks 1 to 3 for Pima data set with high relative difference compared to the 4th classifier (LDA). On the other hand, SL GrCNNC has rank 1 for Statlog data set but with small relative difference to the 2nd classifier (SVM Linear).

Table 4.7: Test accuracy of the compared classifiers.

| Dataset | Pima | BUPA | ILPD | Wisconsin | Statlog | Focality | Bladder |
|---|---|---|---|---|---|---|---|
| NN | 75.42 | 64.70 | 69.85 | 95.98 | 80.89 | 80.96 | 59.87 |
| ANFIS | 75.01 | 71.25 | 70.00 | 95.05 | 82.96 | **81.86** | 57.33 |
| LDA | 75.92 | 68.61 | 71.51 | 95.91 | 82.59 | 81.40 | 67.98 |
| QDA | 74.74 | 62.48 | 57.70 | 94.89 | 79.26 | 80.80 | 66.69 |
| NBC | 74.61 | 55.72 | 57.19 | 95.91 | 82.59 | 81.80 | 69.71 |
| Dtree | 69.78 | 63.96 | 65.12 | 95.47 | 75.93 | 69.60 | 58.13 |
| SVM Linear | 73.57 | 59.80 | 69.44 | 96.20 | 83.33 | 66.40 | **70.98** |
| SVM RBF | 75.01 | 63.29 | 59.44 | 95.62 | 80.74 | 67.00 | 62.80 |
| SVM Polynomial | 72.93 | 52.80 | 57.54 | 95.32 | 65.19 | 77.40 | 64.99 |
| SL GrCNNC | 76.18 | 69.28 | **71.90** | 96.11 | **83.37** | 81.12 | 63.70 |
| ML GrCNNC | 76.58 | 70.51 | 70.00 | **96.51** | 82.59 | 81.34 | 64.93 |
| GrCNNC ANFIS | **77.21** | **72.05** | 70.64 | 96.20 | 81.11 | 81.00 | 63.50 |

Table 4.8: Test AUC of the compared classifiers.

| Dataset | Pima | BUPA | ILPD | Wisconsin | Statlog | Focality | Bladder |
|---|---|---|---|---|---|---|---|
| NN | 82.98 | 67.27 | 66.80 | 99.32 | 88.83 | 59.15 | 62.85 |
| ANFIS | 81.96 | 75.63 | 70.58 | 99.31 | 89.64 | 50.23 | 58.76 |
| LDA | 71.31 | 66.17 | 52.96 | 95.06 | 82.04 | 49.76 | 63.04 |
| QDA | 70.91 | 63.07 | 67.15 | 95.42 | 78.87 | 53.82 | 62.33 |
| NBC | 70.95 | 56.43 | 67.55 | 96.22 | 82.31 | 56.42 | 66.44 |
| Dtree | 67.68 | 62.65 | 57.71 | 94.77 | 75.64 | 49.69 | 53.31 |
| SVM Linear | 82.67 | 68.20 | 69.90 | 99.53 | **90.58** | 47.76 | **73.35** |
| SVM RBF | 83.49 | 73.34 | 62.42 | 98.92 | 87.51 | 58.93 | 58.23 |
| SVM Polynomial | 83.35 | 74.44 | 65.45 | 98.81 | 83.77 | 55.26 | 65.11 |
| SL GrCNNC | 83.43 | 72.26 | 63.43 | 99.36 | 89.01 | 59.80 | 66.73 |
| ML GrCNNC | **83.60** | 73.53 | 67.60 | 99.16 | 89.80 | **60.45** | 67.04 |
| GrCNNC ANFIS | 82.92 | **75.73** | **71.48** | **99.59** | 88.33 | 59.49 | 63.25 |

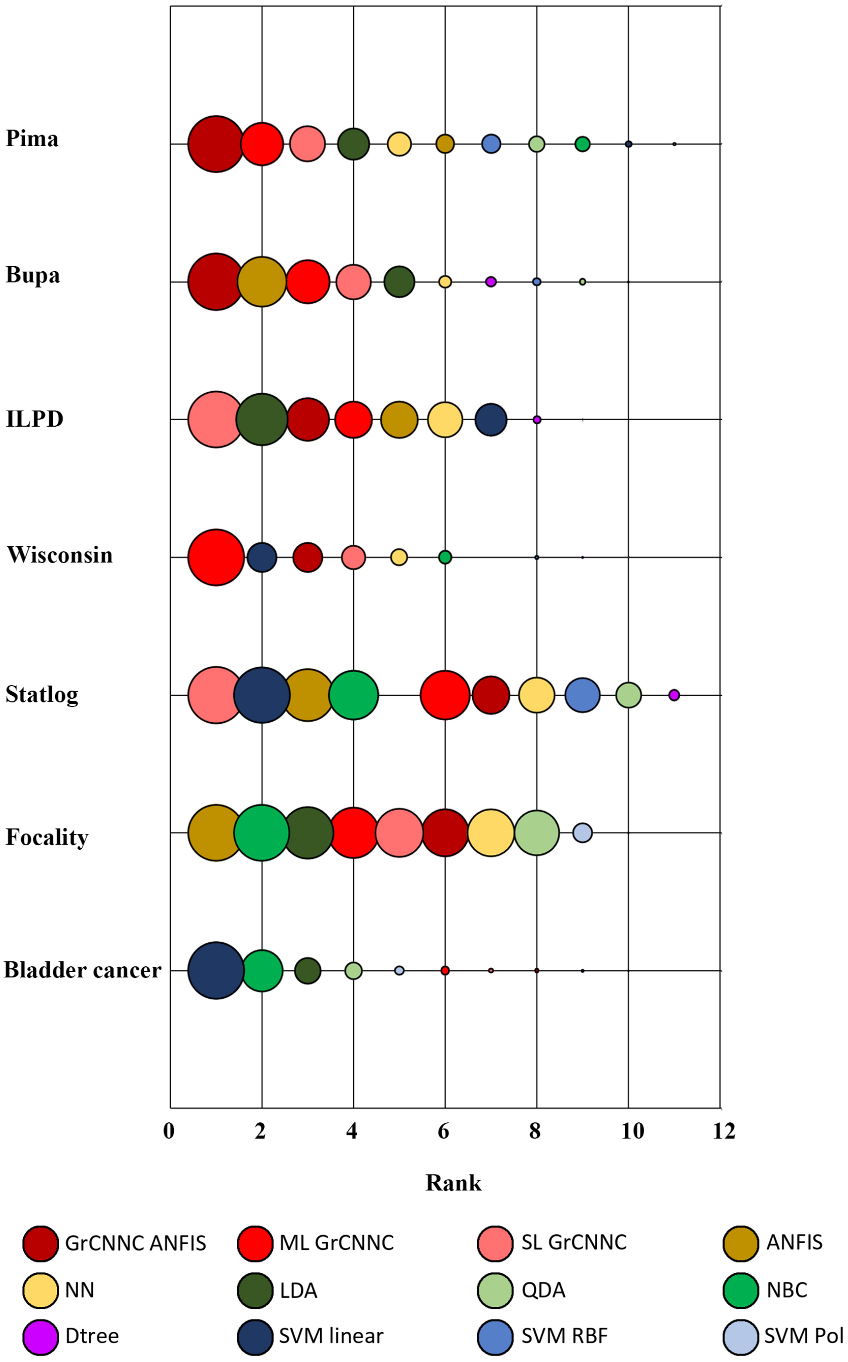

FIGURE 4.15. Relative improvement of accuracy and rank of classifiers in terms of accuracy for each data set.

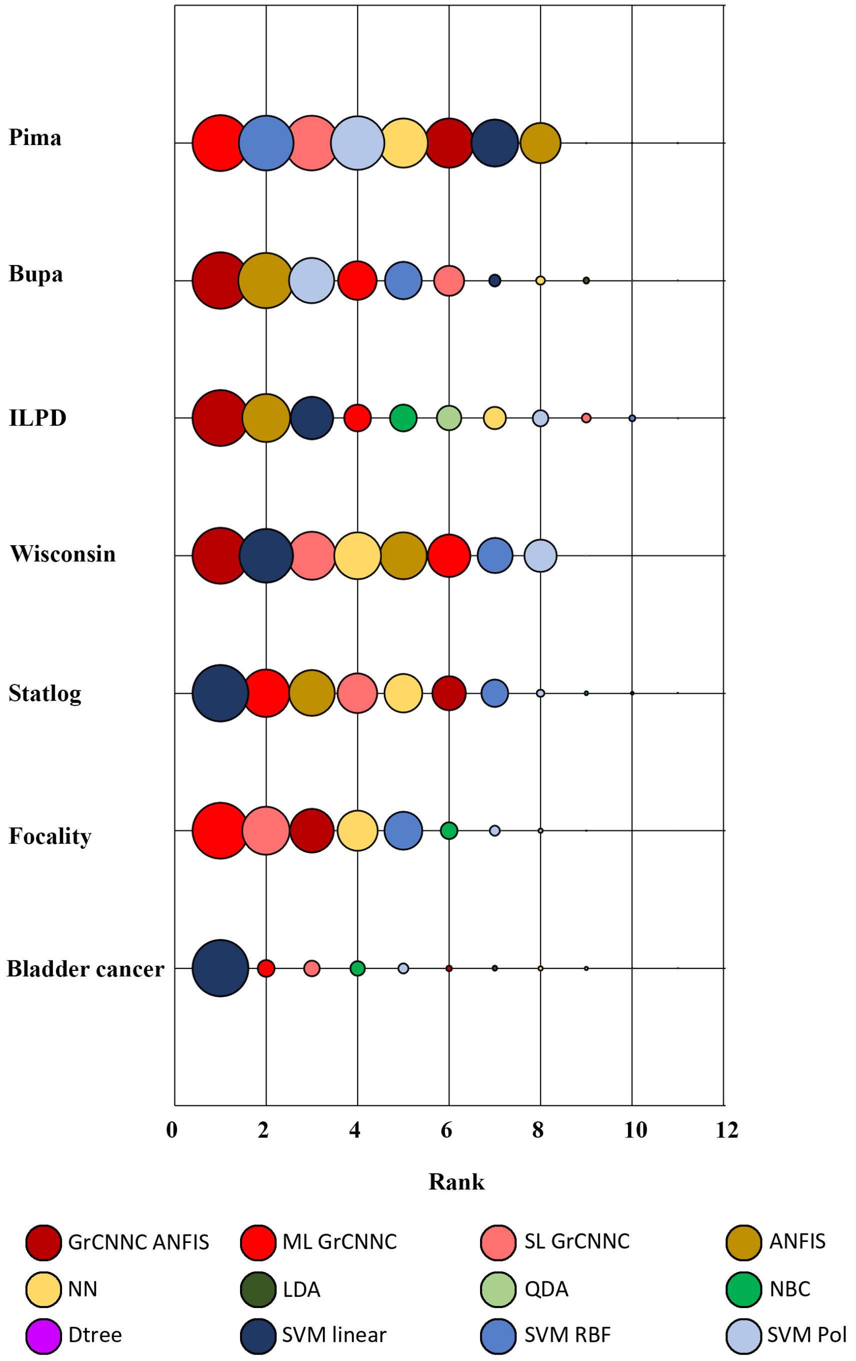

FIGURE 4.16. Relative improvement of AUC and rank of classifiers in terms of AUC for each data set.

## 4.5 Conclusions

A GrC approach for the design of NN classifiers was proposed in this chapter. In the proposed approach, GrCNNC, the overall classification task is divided into smaller subtasks of predicting the inclusion of a data sample to different granules. The GrCNNC uses the FCMGr granulation method to train the NN using information granules. Training data at a given granulation level is represented by the distance of each data sample to each of the granules at that level with the fuzzy c-partition matrix as the target output. The final output is computed from the inclusion score results and the class of each granule. The GrCNNC can use one or more granulation level to train the NN (single level and multi-level GrCNNC).

Systems based on NNs trained using GrCNNC are more interpretable than standard NNs due to the fact that GrCNNC uses information granules to train the NN rather than raw data. Information granules can be viewed as classification rules, thus, training the NN using information granules can be viewed as training the NN to learn these rules. Hence, the classification problem is broken down into the modelling of classification rules represented by the information granules. The size (complexity) of the NN trained using GrCNNC depends on the number of granules rather than data dimensionality, which allow dealing with high-dimensional data using systems with less complexity.

The GrCNNC was implemented and its performance is compared to that of a NN. Results show that GrCNNC has higher accuracy for all the data sets used. The improvement in accuracy is accompanied by higher AUC for 6 out of 7 datasets, which indicates that the improvement in accuracy does not come at the cost of lower specificity or sensitivity. In addition, the results show that GrCNNC has lower STD for all data sets except one which indicates that GrCNNC is more invariant to the change in initial weights and training samples.

In addition, results show that multi-level GrCNNC multi-level GrCNNC has higher accuracy improvement in 5 data sets and higher AUC improvement over NN for 6 out of 7 data sets. Also, this chapter contained an experimental comparison of GrCNNC-based NNs and ANFIS to 9 other classifiers.

CHAPTER 5

# Granular Computing Approach for Classifier Ensemble

A GRANULAR computing approach to classifier ensemble is proposed in this chapter. Section 5.1 provides a brief introduction to ensemble learning. The proposed method is described in Section 5.2. Experimental results of the evaluation of the proposed approach are reported in Section 5.3. Section 5.4 investigates the use of a genetic algorithm for automatic weights assignment of the ensemble combiner. Summary of this chapter is given in Section 5.5.

## 5.1 Introduction

Aiming at increasing the classification performance, ensemble learning adopts the idea of combining several classifiers to provide improved generalisation ability. This improvement, however, comes at the price of increased complexity and computational cost. Ensemble learning was first introduced in the late 1970's [168, 169], and since then, it has gained the interest of many researchers and has witnessed great improvements. Nowadays, ensemble learning has been widely used in a wide range of classification and pattern recognition applications [170, 171, 172, 173].

An ensemble system consists of two main components: an ensemble of base classifiers and an output combination mechanism to combines the outputs of the base classifiers. The idea of ensemble learning is based on building a system of diverse classifiers, since combining classifiers with similar outputs is not expected to result in any improvement [174].

There are two main types of ensemble output combiners: algebraic combiners and voting-based combiners. Algebraic combiners use continuous outputs (scores) of base classifiers to produce the overall ensemble output. On the other hand, voting-based combiners use class labels resulted from the base classifiers to produce the overall ensemble output. In addition, some ensemble learning algorithms use different methods to combine the outputs, such as a second layer classifier. Some algebraic and voting-based combiners are fixed combiners, i.e. they do not involve learning. However, other combiners require some training. Examples of fixed combiners are mean combiner (algebraic) and majority vote combiner (voting-based). Weighted sum (algebraic) and weighted vote (voting-based) are examples of trainable combiners [14].

The construction of base classifiers can be done in two different methods: dependent and independent. In dependent method, the construction (or training) of a base classifier is influenced by the output of one or more of the other classifiers in the ensemble. On the other hand, in independent method, base classifiers are constructed separately.

The diversity of the base classifiers is achieved by two main approaches: using different partition of the training data or using different types or parameters of base classifiers. The first approach can be achieved by choosing different training samples or different sets of input features (input variables) for each base classifier.

Four of well-known ensemble algorithms are:

1. Bagging

   Bagging (short for Bootstrap Aggregating) algorithm is one of the earliest and simplest ensemble-based algorithms. In bagging, $C$ base classifiers are trained using $C$ different replica's of the training data set using bootstrap sampling, i.e. sampling with replacement [175]. With bootstrap, some training instances may be used more than once in some of the training sets while some instances may not be used at all. Thus, the diversity of the base classifiers is achieved by the variation of the training data sets of the classifiers. Also, using this approach, the base classifiers can be independently trained in parallel which is one of the main advantages of bagging algorithm since it reduces the training time.

2. Boosting: Boosting is similar to bagging considering that both it relies on using different set of training data to produce diversity. However, instead of randomly sampling the training data, boosting assign weights to training samples to indicate usefulness of a training sample. Each time a base classifier is trained, the weights the training data are updated such that weights of correctly classified samples are decreased while weights of incorrectly classified samples are increased. In this way, the training of the next classifier is is focused on misclassified data [176]. The final ensemble output is computed using majority voting method. Unlike bagging, base classifiers in boosting algorithm can not be trained independently because the training of a classifier depends on the classification results if the previous base classifier.

3. Stacking: In stacking (or stacked generalisation), the ensemble consists of two levels of classifiers. The first level is a set of base classifiers trained using bootstrap method. The outputs of first-level classifiers, along with the actual class labels, are used to train a meta-classifier in the second level [177]. Stacking is based on the idea that the meta-classifier is trained so that it can learn, from the training data, where first-level classifiers fail and overcome their failure. To classify a new test sample, classification is first performed by first-level classifiers. Then, output scores of these classifiers are used by the meta-classifier to produce the final output. Stacking can be used to combine classifiers trained using a single data set with different learning methods (heterogeneous ensemble) [14].

4. Mixture of experts: In mixture of experts, a complex classification task is divided into a set of subtasks and base classifiers (experts) are trained for

different subtasks. The output of the expert is combined by a gating network. Input training samples are used to train both the experts and the gating network. Due to this approach, diversity is already achieved by using different subtasks for training [178].

In this chapter, a GrC approach for classifier ensemble is proposed. The proposed approach employs GrCNNC for the generation and training of the base classifiers. Weighted sum combiner is used in the proposed approach and a Genetic Algorithm (GA) is used to assign the weights automatically.

## 5.2 Proposed Granular Computing Approach

In this section, a GrC approach for classifier ensemble is proposed. The main aim of the proposed method is to produce diversity in the constructed base classifiers. Since the result of FCMGr is a set of granules at different granulation levels, it is expected that different granularity levels produce different classifiers. Therefore, like bagging and boosting, the proposed approach relies on varying the training data to produce diversity. However, instead of sampling with replacement that results in using a subset of the original training data, all the training data are used to train all base classifiers, each at different level (or levels) of granularity.

Thus, the construction of a classifier ensemble using the proposed approach consists of three main stages:

- Information granulation stage: In this stage, FCMGr is used to construct information granules at different levels of granularity.
- Classifier training stage: For each base classifier, a granularity level (or set of levels) is selected and its granules are used for training using GrCNNC approach.
- Ensemble combination stage: A combiner is designed to combine the outputs of the base classifiers.

To classify new data, the output of each base classifier is first computed using GrCNNC. Then, the final output is produced by the combiner. The combiner used in the proposed approach is the weighted sum combiner whose output is given by:

$$\mu_j(x) = \sum_{t=1}^{T} w_t d_{t,j}(x) \tag{5.1}$$

where:

$\mu_j(x)$ is the final ensemble score of class $j$ for data sample $x$,

$T$ is the number of base classifiers,

$w_t$ is a weight associated with base classifier $t$,

and $d_{t,j}(x)$ is the score of class $j$ for data sample $x$ using classifier $t$.

Since GrCNNC is used, the complexity of a base classifier (number of inputs and outputs) is determined by the granularity level or levels used to train it. Thus, using different granularity levels results in variation in classifier parameters which contributes to the diversity of the ensemble.

To focus on investigating the effectiveness of GrCNNC in providing diversity for the construction of classifier ensembles, the problem of determining combiner weights automatically is left to Section 5.4. Instead, the weights are determined experimentally by searching for the weights that produce best classification performance.

## 5.3 Experimental Results

The data sets presented in 3.4.1 are used to evaluate the proposed approach. Each data set is divided into 3 parts: training (60%), validation (20%) and test (20%). 5-fold cross validation is used, therefore, each classifier is tested 5 times with different test partition each time and the mean accuracy and AUC are computed.

The proposed approach is used to build classifier ensembles for each data set. Each ensemble consists of five classifiers trained with granules at five different granularity levels. Granularity levels are selected of relatively low granularity to avoid higher complexity and increase generalisation. It should be noticed that granularity levels used in this experiment are the resulted levels from experiment in 3.4.3 where the number of granularity levels is between 11-13 for all data sets. Therefore, constructing larger ensembles may require granulation with more granularity levels. This can be achieved by decreasing the granulation variable $R_g$ in the FCMGr algorithm in section 3.3.3.

To test the diversity of the base classifiers, training outputs of the base classifiers are compared and the percentage of samples that are classified differently by each two classifiers is computed. Table 5.1 shows classification disagreement percentage results for Pima data set. Each value in the table represent the disagreement percentage between the corresponding classifiers. When there is a large number of granularity levels, this measure can be used to select the most diverse levels.

TABLE 5.1. Classification disagreement (%) of the five base classifiers for Pima data set.

| Classifier | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 0.00 | | | | |
| 2 | 15.80 | 0.00 | | | |
| 3 | 13.36 | 7.98 | 0.00 | | |
| 4 | 15.15 | 9.45 | 5.37 | 0.00 | |
| 5 | 15.80 | 9.77 | 7.33 | 5.54 | 0.00 |

The proposed approach is implemented using MATLAB R2016a. Test accuracy and AUC of the proposed ensemble approach compared to those of GrCNNC are shown in Table 5.2. The ensemble has higher accuracy for 5 out of 7 data sets with high improvement for BUPA and bladder cancer data sets. AUC of the proposed approach is higher for 6 data sets with high improvement for BUPA, ILPD and bladder cancer data sets.

Similarly, training accuracy and AUC are shown in Table 5.3. The proposed approach has higher accuracy in 6 data sets and higher AUC for all the data sets with improvement margin of 0.16%-4.19%.

ROC curves of the proposed ensemble approach and single level GrCNNC for all the data sets are shown in Figures 5.1-5.7. These curves are drawn for 1 of the 5 test partitions. From the figures, noticeable improvement of the ROC curves of the proposed ensemble approach can be noticed, especially for the BUPA, ILPD and bladder cancer data sets. While these curves provide a way to compare the performance of the classifier at hand, it should be noticed that they reflect the performance for only one fold of the 5-fold cross validation used. The overall mean AUC may provide more accurate robust measure.

TABLE 5.2. Test Accuracy (%) and AUC (%) of SL GrCNNC and SL ensemble.

| Data sets | GrCNNC | | Ensemble | | Improvement | |
|---|---|---|---|---|---|---|
| | Acc. | AUC | Acc. | AUC | Acc. | AUC |
| Pima | 76.18 | **83.43** | **76.62** | 83.32 | 0.44 | -0.12 |
| BUPA | 69.28 | 72.26 | **71.01** | **74.52** | 1.73 | 2.26 |
| ILPD | **71.90** | 63.43 | 71.25 | **67.43** | -0.65 | 4.00 |
| Wisconsin | 96.11 | 99.36 | **96.51** | **99.43** | 0.40 | 0.07 |
| Statlog | **83.37** | 89.01 | 83.30 | **89.80** | -0.07 | 0.78 |
| Focality | 81.12 | 59.80 | **81.52** | **60.29** | 0.40 | 0.50 |
| Bladder Cancer | 63.70 | 66.73 | **65.45** | **69.09** | 1.75 | 2.37 |

TABLE 5.3. Training Accuracy (%) and AUC (%) of SL GrCNNC and SL ensemble.

| Data sets | GrCNNC | | Ensemble | | Improvement | |
|---|---|---|---|---|---|---|
| | Acc. | AUC | Acc. | AUC | Acc. | AUC |
| Pima | 79.63 | 85.78 | **80.24** | **86.36** | 0.61 | 0.58 |
| BUPA | 75.98 | 79.90 | **77.13** | **81.02** | 1.15 | 1.12 |
| ILPD | 72.67 | 68.58 | **73.14** | **72.77** | 0.47 | 4.19 |
| Wisconsin | 97.99 | 99.46 | **98.13** | **99.62** | 0.14 | 0.16 |
| Statlog | 84.82 | 89.81 | **85.25** | **91.63** | 0.43 | 1.82 |
| Focality | **82.05** | 63.98 | 81.93 | **64.58** | -0.12 | 0.60 |
| Bladder Cancer | 72.67 | 77.98 | **73.23** | **78.84** | 0.56 | 0.86 |

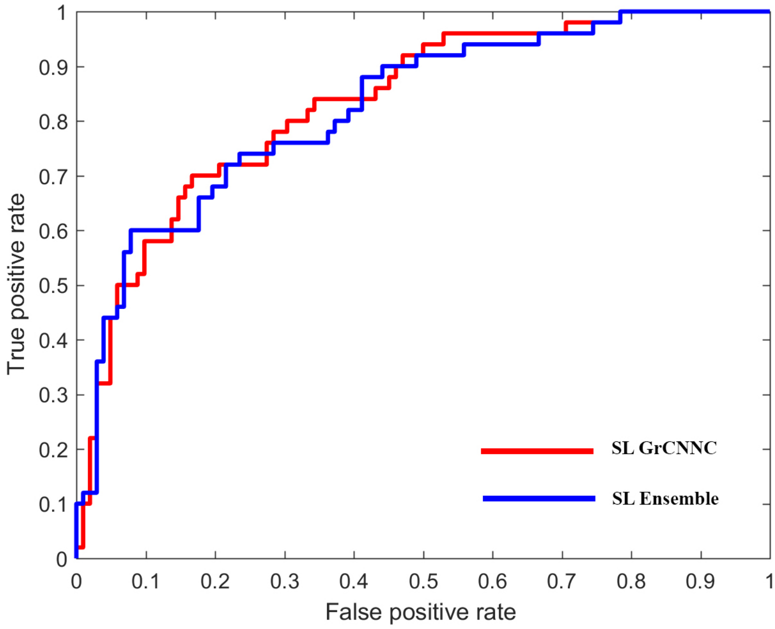

FIGURE 5.1. ROC curve of SL GrCNNC and SL Ensemble for Pima data set.

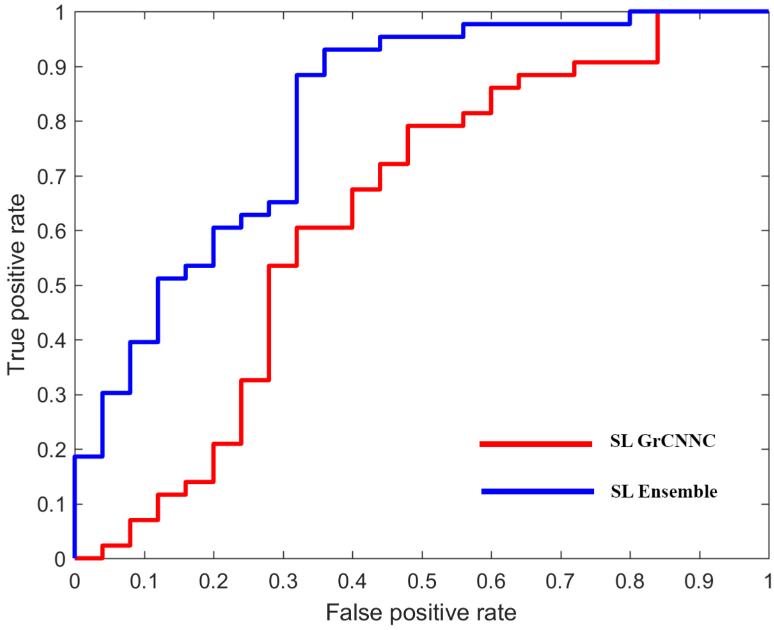

FIGURE 5.2. ROC curve of SL GrCNNC and SL Ensemble for BUPA data set.

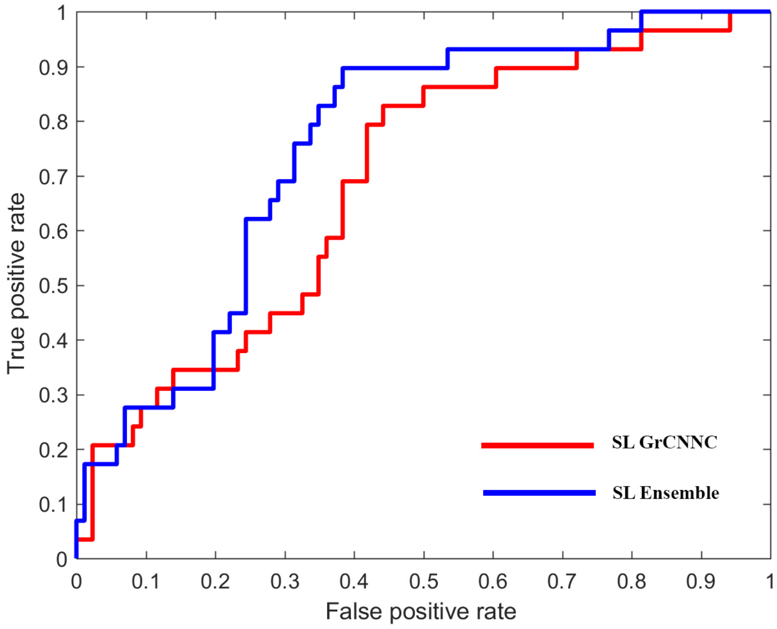

FIGURE 5.3. ROC curve of SL GrCNNC and SL Ensemble for ILPD data set.

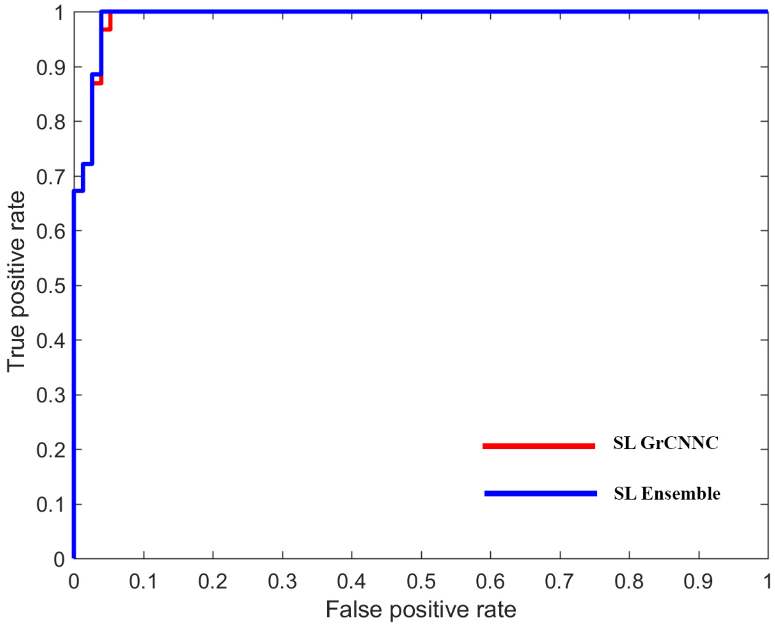

FIGURE 5.4. ROC curve of SL GrCNNC and SL Ensemble for Wisconsin data set.

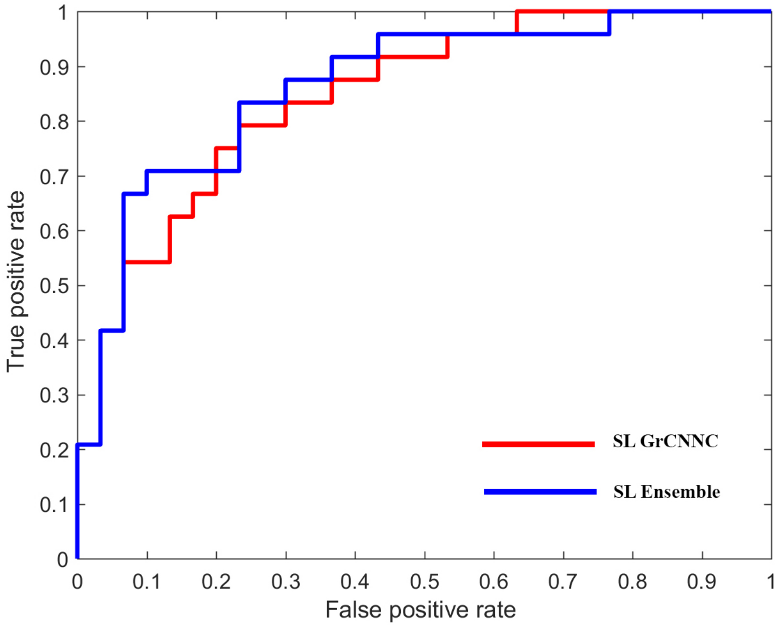

FIGURE 5.5. ROC curve of SL GrCNNC and SL Ensemble for Statlog data set.

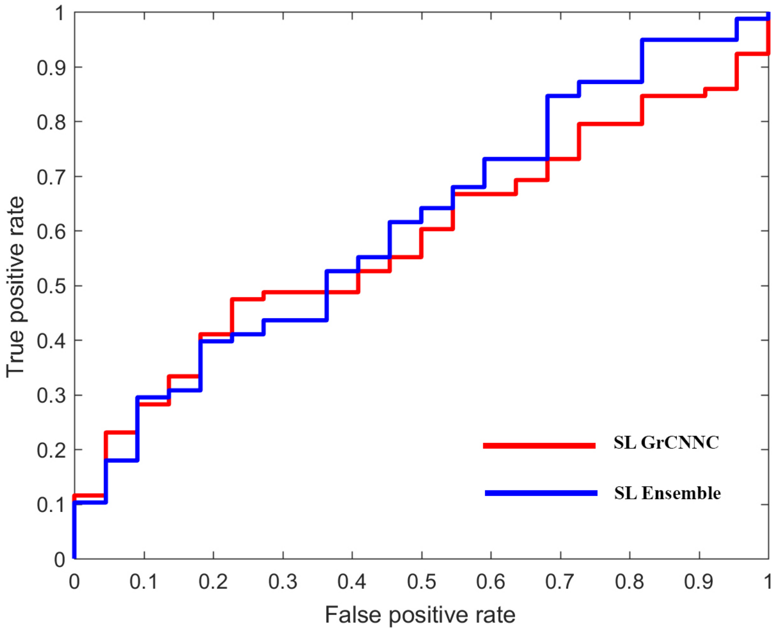

FIGURE 5.6. ROC curve of SL GrCNNC and SL Ensemble for Focality data set.

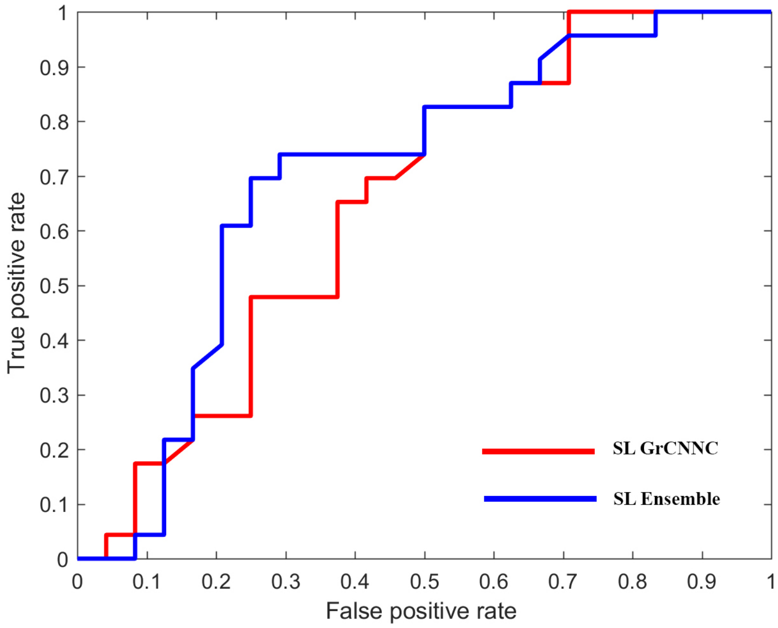

FIGURE 5.7. ROC curve of SL GrCNNC and SL Ensemble for bladder cancer data set.

### 5.3.1 Ensemble with Multi-level GrCNNC

One or more multi-level GrCNNC classifiers can be included in the ensemble using the proposed method. Test and training results of such ensembles are shown in Tables 5.4 and 5.5. Compared to GrCNNC, higher accuracy is obtained for all the data sets in test and training. Improvement in accuracy is accompanied by improvement of AUC for 4 data sets in test and training. However, compared to single level GrCNNC ensembles, multi-level GrCNNC ensemble (ML ensemble) results in higher test accuracy and AUC for only 4 data sets with highest improvement for BUPA data set as shown in Figures 5.8 and 5.9.

Similarly, improvement in training accuracy and AUC are compared in Figures 5.10 and 5.11.

Improvement values in these figures are computed relative to the performance of single level GrCNNC. As shown in the figures, training accuracy and AUC of ML ensemble are higher than SL ensemble for 6 data sets.

TABLE 5.4. Test Accuracy (%) and AUC (%) of ML GrCNNC and ML ensemble.

| Data sets | GrCNNC | | Ensemble | | Improvement | |
|---|---|---|---|---|---|---|
| | Acc. | AUC | Acc. | AUC | Acc. | AUC |
| Pima | 76.58 | **83.60** | **76.80** | 83.36 | 0.22 | -0.25 |
| BUPA | 70.51 | 73.53 | **71.53** | **74.85** | 1.02 | 1.32 |
| ILPD | 70.00 | 67.60 | **70.42** | **68.25** | 0.42 | 0.65 |
| Wisconsin | 96.51 | 99.16 | **96.55** | **99.40** | 0.04 | 0.23 |
| Statlog | 82.59 | 89.80 | **82.81** | **89.94** | 0.22 | 0.14 |
| Focality | 81.34 | **60.45** | **81.40** | 60.38 | 0.06 | -0.08 |
| Bladder Cancer | 64.93 | **67.04** | **65.49** | 66.77 | 0.56 | -0.27 |

TABLE 5.5. Training Accuracy (%) and AUC (%) of ML GrCNNC and ML ensemble.

| Data sets | GrCNNC | | Ensemble | | Improvement | |
|---|---|---|---|---|---|---|
| | Acc. | AUC | Acc. | AUC | Acc. | AUC |
| Pima | 76.58 | **83.60** | **76.80** | 83.36 | 0.22 | -0.25 |
| BUPA | 70.51 | 73.53 | **71.53** | **74.85** | 1.02 | 1.32 |
| ILPD | 70.00 | 67.60 | **70.42** | **68.25** | 0.42 | 0.65 |
| Wisconsin | 96.51 | 99.16 | **96.55** | **99.40** | 0.04 | 0.23 |
| Statlog | 82.59 | 89.80 | **82.81** | **89.94** | 0.22 | 0.14 |
| Focality | 81.34 | **60.45** | **81.40** | 60.38 | 0.06 | -0.08 |
| Bladder Cancer | 64.93 | **67.04** | **65.49** | 66.77 | 0.56 | -0.27 |

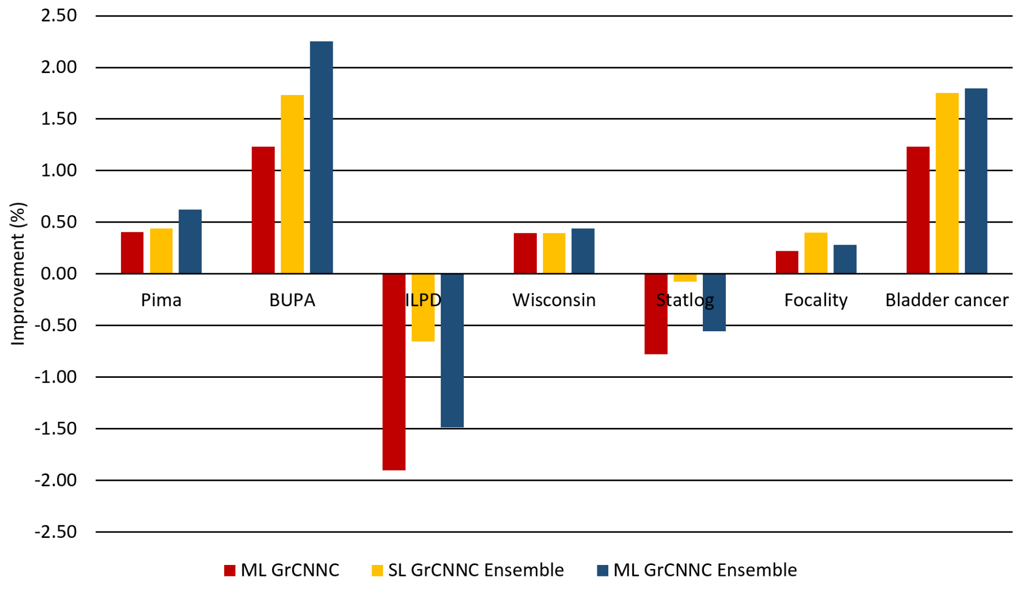

FIGURE 5.8. Improvement in test accuracy of multi-level GrCNNC NN, SL ensemble and ML ensemble.

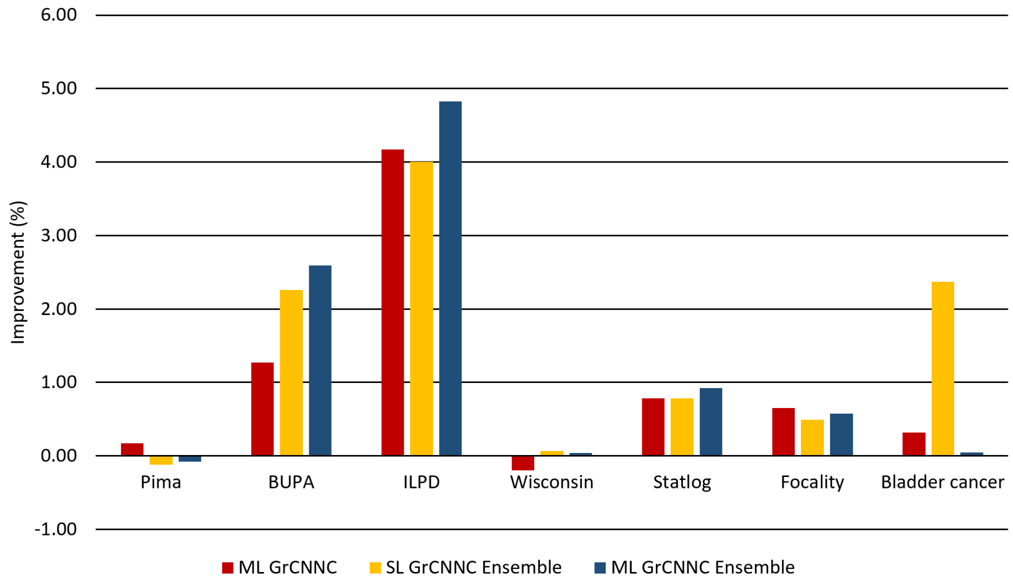

FIGURE 5.9. Improvement in test AUC of multi-level GrCNNC NN, SL ensemble and ML ensemble.

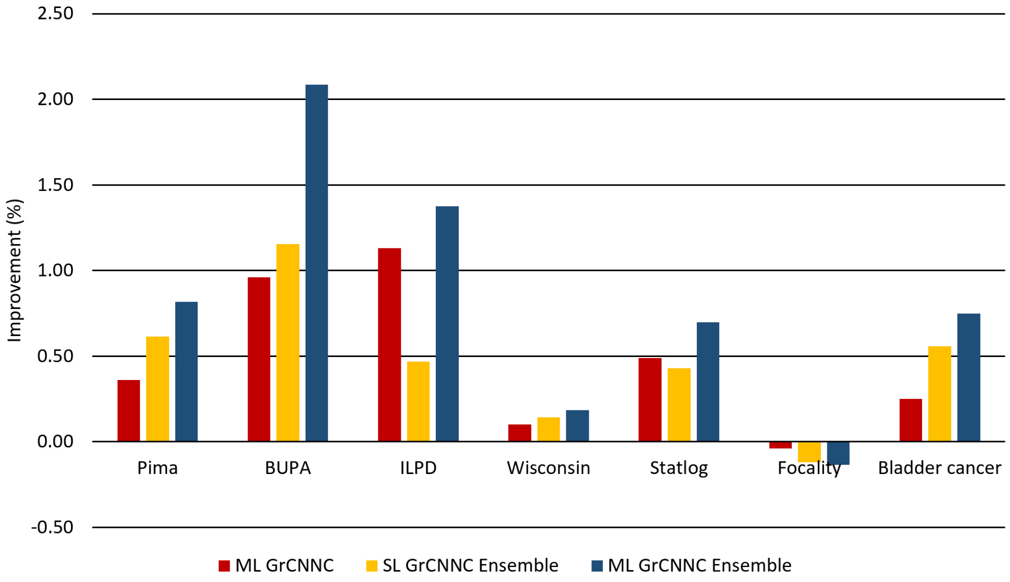

FIGURE 5.10. Improvement in training accuracy of multi-level GrCNNC NN, SL ensemble and ML ensemble.

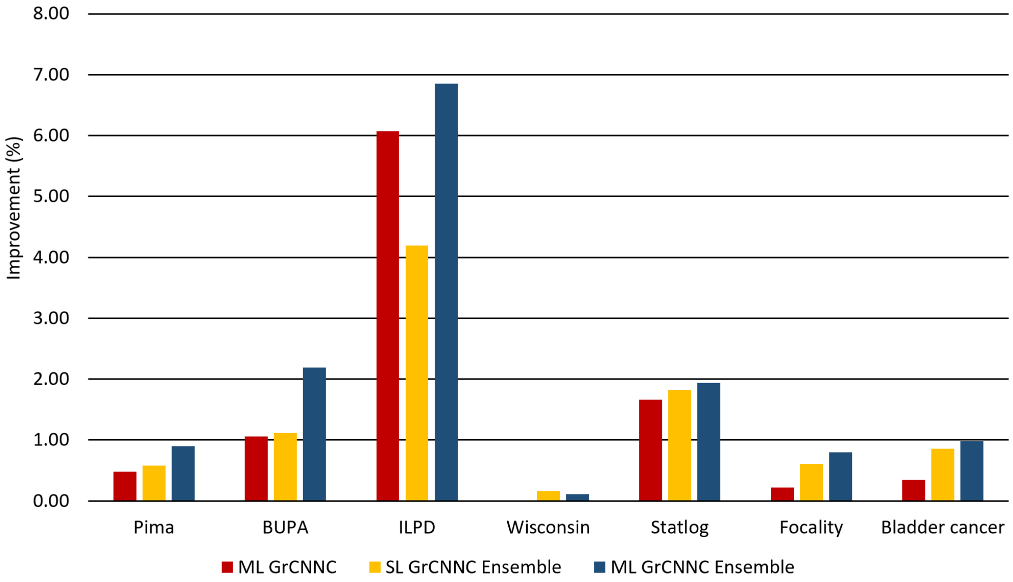

FIGURE 5.11. Improvement in training AUC of multi-level GrCNNC NN, SL ensemble and ML ensemble.

## 5.4 Genetic Algorithm for Weights Assignment

So far, the weights of the ensemble combiner were determined experimentally by searching for the weights that produce best classification performance. However, it is desirable to develop an approach for automatic weights assignment. In this section, GA is used to achieve this task.

GA is a stochastic optimization algorithm that imitates natural evolution. Like natural evolution, genetic algorithms generate populations of different possible solutions successively. Each generation consists of a number of individual solutions, or chromosomes, which compete with each other. The fittest chromosomes with respect to some criteria get higher probabilities of being selected to reproduce the next generation. A chromosome comprises a set of coded properties, or genes, which describe the solution. Genes are either binary coded or real coded. The next generation of chromosomes is produced from selected chromosomes, called parents, by a reproduction method. New parents are selected for each new child to be generated [179].

There are various procedures to select the parents like: roulette wheel, linear ranking and geometric ranking. The reproduction of a new generation from parents is usually done using two main operators: crossover and mutation. In crossover, chromosomes of one child or more are produced from genes of two or more parent chromosomes using a crossover technique. On the other hand, mutation produces one new chromosome by altering one or more genes in a single chromosome [180].

To use GA in determining the weights of the ensemble combiner, a fitness must be first defined. GA use this function to evaluate the chromosomes. Aiming for optimising the classification performance, training AUC is used as a fitness function. In other words, the error function $F(x)$ that the GA tries to minimise is given by:

$$F(x) = 1 - AUC \tag{5.2}$$

Using this fitness function, a GA is used to determine the five combiner weights of the proposed ensemble. Maximum number of generations is set to 50 and population size is set to 40. Plots of fitness function values for each generation for two example data sets (BUPA and Statlog) are shown in Figures 5.12 and 5.13.

Determining combiner weights using GA uses training data only as opposed to the experimental method where the weights that result in best test results are selected. Therefore, GA may not result in optimum test results, but it provides a practical method since the desired test outputs are not available in practice.

Table 5.6 shows classification performance results for the ensemble tuned by GA. The results are compared to single level GrCNNC and SL ensemble whose weights are determined experimentally. Although optimal results have not been achieved by the GA, ensemble tuned by the GA achieved higher accuracy and AUC than single level GrCNNC for 5 and 6 data sets respectively. This is shown in Figures 5.14 and 5.15. In these figures, improvement is computed relative to the performance of single level GrCNNC.

TABLE 5.6. Accuracy (%) and AUC (%) of SL GrCNNC, SL ensemble and ensemble tuned by GA.

| Data sets | SL GrCNNC | | SL Ensemble | | GA Ensemble | |
|---|---|---|---|---|---|---|
| | Acc. | AUC | Acc. | AUC | Acc. | AUC |
| Pima | 76.18 | 83.43 | 76.62 | 83.32 | 76.10 | 83.21 |
| BUPA | 69.28 | 72.26 | 71.01 | 74.52 | 70.86 | 74.60 |
| ILPD | 71.90 | 63.43 | 71.25 | 67.43 | 69.35 | 67.94 |
| Wisconsin | 96.11 | 99.36 | 96.51 | 99.43 | 96.50 | 99.42 |
| Statlog | 83.37 | 89.01 | 83.30 | 89.80 | 82.56 | 89.89 |
| Focality | 81.12 | 59.80 | 81.52 | 60.29 | 81.20 | 60.59 |
| Bladder Cancer | 63.70 | 66.73 | 65.45 | 69.09 | 64.59 | 68.93 |

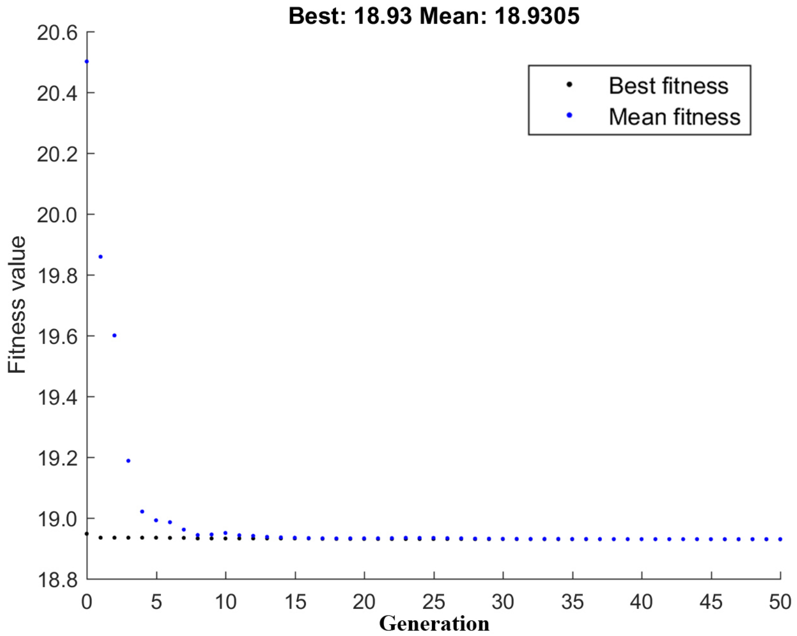

FIGURE 5.12. Best score value and mean score of GA for Bupa data set.

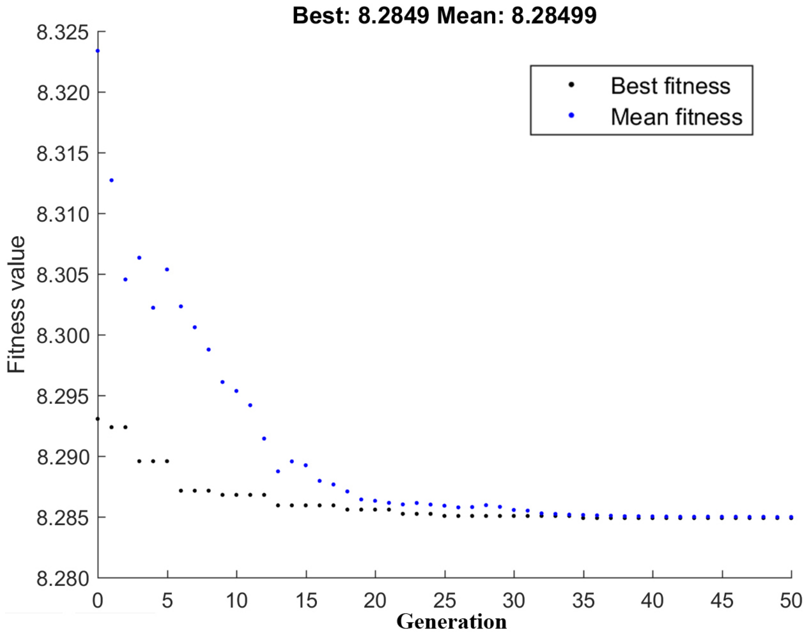

FIGURE 5.13. Best score value and mean score of GA for Statlog data set.

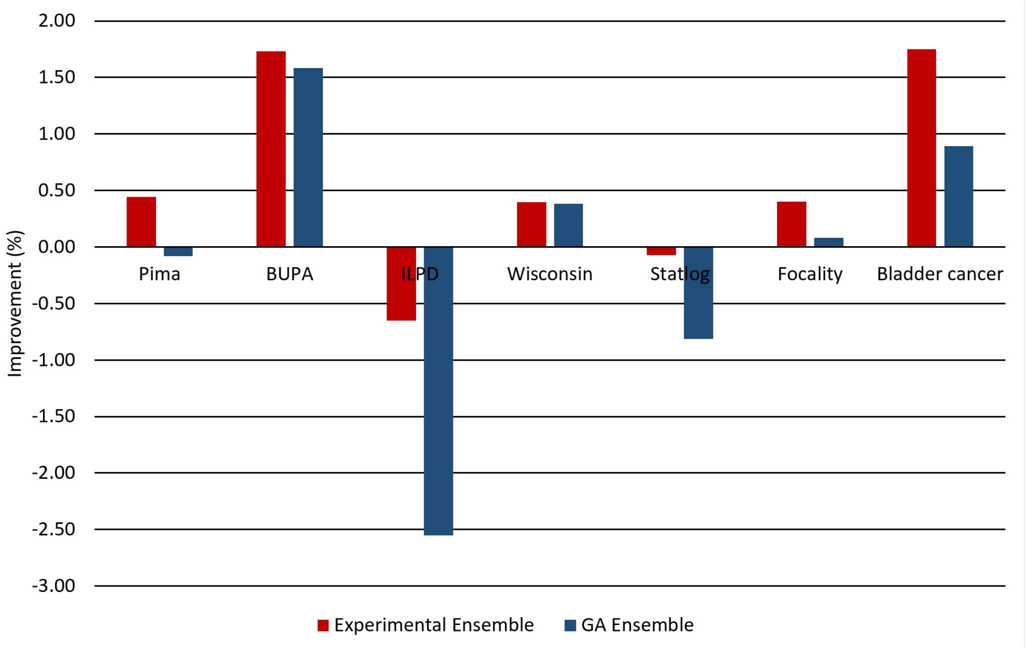

FIGURE 5.14. Improvement in accuracy of SL ensemble and GA ensemble.

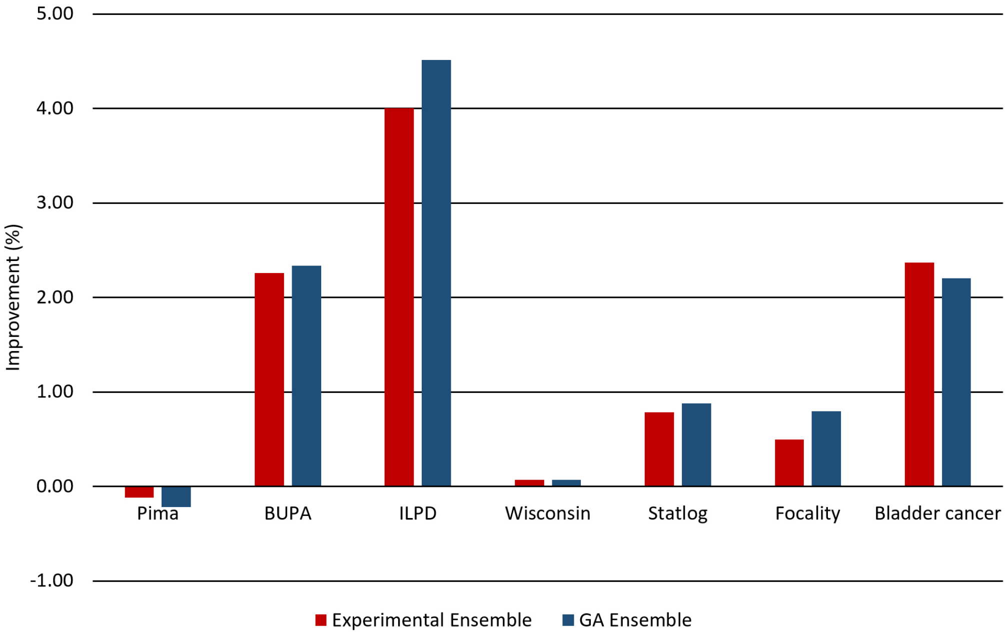

FIGURE 5.15. Improvement in AUC of SL ensemble and GA ensemble.

## 5.5 Conclusions

In this chapter, a GrC approach for classifier ensemble was proposed. The proposed approach relies on using different granularity levels to produce diversity in the constructed base classifiers. GrCNNC was used for the generation and training of the base classifiers and a weighted sum combiner was used to produce the final ensemble output.

The proposed approach was implemented and used to build classifier ensembles for each data set. Each ensemble consists of five classifiers trained with granules at five different granularity levels. Granularity levels were selected of relatively low granularity to avoid higher complexity and increase generalisation.

To investigates the effectiveness of the proposed method in generating divers base classifiers, the weights of the ensemble combiner were determined experimentally by searching for the weights that produce best classification performance. Based on the experimental results, the proposed approach has improved the classification performance of the base classifiers for most of the data sets in terms of both accuracy and AUC.

To address the problem of automatic weights assignment, a GA was used to tune the combiner weights. Although optimal results have not been achieved by the GA, ensemble tuned by the GA achieved higher accuracy and AUC than single level GrCNNC for most of the data sets.

CHAPTER 6

# Conclusions and Future Work

## 6.1 Conclusions

This thesis was focused on adopting, implementing and evaluating a GrC approach for the design of classification systems. In the proposed approach, granulation is performed by using a data clustering method and the resulted granules are used to build classifiers of different types. One or more granulation levels are used to construct the classifiers and their performance is evaluated and compared to standard classifiers. Classification systems used to investigate the proposed approach are FIS, NN, ANFIS and classifier ensemble.

This thesis is made up of four main chapters:

- Chapter 2: In this chapter, an introduction to the fundamentals and principles of GrC was provided. In addition, it included a review of the methods, frameworks and applications of GrC focusing on the synergy of GrC and computational methods, including data clustering and fuzzy theory. The aim of the review was to provide a theoretical background of information granulation methods and their applications.

- Chapter 3: In this chapter, a method for information granulation was proposed. The proposed method, FCMGr, aims to produce information granules from data at multiple levels of granularity. The granulation process is performed using a modified FCM clustering method followed by a coarsening process achieved by merging smaller clusters into larger clusters to form the next lower granularity level. The modified FCM starts by choosing initial clusters centres then computes the new fuzzy partition matrix and the updated clusters centres. In addition, the fuzziness factor $m$ is chosen dynamically for each granularity level rather than being fixed prior the algorithm start.

  FCMGr was used to build a fuzzy classifier for each granulation level. A fuzzy rule is constructed from each cluster where Gaussian membership functions are used to represent the fuzzy variables. The performance of the classifiers was evaluated using the selected data sets and measured using two classification performance measures: accuracy and AUC. Results show that fuzzy systems constructed using FCMGr achieved better classification performance.

  Furthermore, a method of automatically selecting the granulation level that results in better classification performance was proposed. Results show that the automatically selected granulation levels resulted in better accuracy than the selected FCM classifiers for all the data sets over a wide range of number of clusters ($N_C$ and fuzziness factor $m$.

  In addition to using one granulation level for the construction of a FIS, multi-levels of the FCMGr were used. Results show that multi-level FCMGr has better accuracy than single-level FCMGr for three data sets.

- Chapter 4: In this chapter, FCMGr was used in an approach for the design of NN classifiers. In the proposed approach, GrCNNC, the overall classification task is divided into smaller subtasks of predicting the inclusion of a data sample to different granules. Granules resulted from FCMGr were used to train the NN. The GrCNNC can use one or more granulation level to train the NN (single level and multi-level GrCNNC).

  NNs trained using GrCNNC are more interpretable than standard NNs due to the fact that GrCNNC uses information granules to train the NN rather than raw data. As a result, the classification problem is broken down into the modelling of classification rules represented by the information granules.

Results of comparing GrCNNC to standard NNs show that GrCNNC has higher accuracy for all the data sets used. The improvement in accuracy is accompanied by higher AUC for most of the datasets, which indicates that the improvement in accuracy does not come at the cost of lower specificity or sensitivity. In addition, the results show that GrCNNC has lower STD for all data sets except one which indicates that GrCNNC is more invariant to the change in initial weights and training samples. In addition, results show that multi-level GrCNNC multi-level GrCNNC has higher accuracy improvement in five data sets and higher AUC improvement over NN for most of the data sets.

- Chapter 5: In this chapter, a GrC approach for classifier ensemble was proposed. The proposed approach uses different granularity levels to produce diversity in the constructed base classifiers. GrCNNC was used for the generation and training of the base classifiers and a weighted sum combiner was used to produce the final ensemble output. Each ensemble consists of a number of classifiers trained with granules at different granularity levels. Based on the experimental results, the proposed approach has improved the classification performance of the base classifiers for most of the data sets in terms of both accuracy and AUC for test and training. In addition, a GA was used to tune the combiner weights to assign combiner weights automatically. Although optimal results have not been achieved by the GA, ensemble tuned by the GA achieved higher accuracy and AUC than single level GrCNNC for most of the data sets.

## 6.2 Future Work

As a future work direction, the following is suggested:

- While information granulation was performed through data clustering in this thesis, other methods of information granulation can be used. In particular, the use of rough sets can be investigated.

- One of the central problems of GrC is determining the most useful granularity level automatically. Although an attempt was proposed in this thesis for the case of fuzzy systems, the selection of the best granularity level for NN, ANFIS and classifier ensemble was made experimentally. Therefore, more general and robust methods are needed. For example, the use of entropy-based measures can be investigated.

- The proposed approach for classifier ensemble can be used with different learning/combining methods. In addition, the effect of ensemble size and granularity levels can be investigated.

- In addition to GA used in this thesis, other optimisation methods, like particle swarm optimisation, can be used for the tuning of ensemble combiner weights.

- Although the proposed approach is a supervised learning approach that uses target output in the learning process, it can be adopted to address semi-supervised learning tasks where only partial knowledge of the target output is available. Once granules are generated from data, knowledge represented by these granules may allow for better performance.

- The performance of the proposed approach on large high-dimensional data sets can be studied to investigate the advantage of data granulation in dealing with large number of data samples and reducing complexity and/or dimensionality.

# References

[1] Y. Cao, “Aggregating multiple classification results using choquet integral for financial distress early warning,” *Expert Systems with Applications*, vol. 39, no. 2, pp. 1830 – 1836, 2012.

[2] J. A. Sanz, D. Bernardo, F. Herrera, H. Bustince, and H. Hagras, “A compact evolutionary interval-valued fuzzy rule-based classification system for the modeling and prediction of real-world financial applications with imbalanced data,” *IEEE Transactions on Fuzzy Systems*, vol. 23, pp. 973–990, Aug 2015.

[3] Y. x. Jiang, H. Wang, and Q. f. Xie, “Classification model of companies’ financial performance based on integrated support vector machine,” in *2009 International Conference on Management Science and Engineering*, pp. 1322–1328, Sept 2009.

[4] G. Schaefer, B. Krawczyk, M. E. Celebi, and H. Iyatomi, “An ensemble classification approach for melanoma diagnosis,” *Memetic Computing*, vol. 6, no. 4, pp. 233–240, 2014.

[5] J. Y. Choi, D. H. Kim, K. N. Plataniotis, and Y. M. Ro, “Classifier ensemble generation and selection with multiple feature representations for classification applications in computer-aided detection and diagnosis on mammography,” *Expert Systems with Applications*, vol. 46, pp. 106 – 121, 2016.

[6] M. Heydari, M. Teimouri, Z. Heshmati, and S. M. Alavinia, “Comparison of various classification algorithms in the diagnosis of type 2 diabetes in Iran,”

*International Journal of Diabetes in Developing Countries*, vol. 36, no. 2, pp. 167–173, 2016.

[7] D. Dou and S. Zhou, "Comparison of four direct classification methods for intelligent fault diagnosis of rotating machinery," *Applied Soft Computing*, vol. 46, pp. 459 – 468, 2016.

[8] A. Moosavian, M. Khazaee, H. Ahmadi, M. Khazaee, and G. Najafi, "Fault diagnosis and classification of water pump using adaptive neuro-fuzzy inference system based on vibration signals," *Structural Health Monitoring*, vol. 14, no. 5, pp. 402–410, 2015.

[9] K. Wang, Y. Wan, and S. Shen, "Classifications of remote sensing images using fuzzy multi-classifiers," in *Intelligent Computing and Intelligent Systems, 2009. ICIS 2009. IEEE International Conference on*, vol. 4, pp. 411–414, Nov 2009.

[10] O. Gordo, E. Martinez, C. Gonzalo, and A. Arquero, "Classification of satellite images by means of fuzzy rules generated by a genetic algorithm," *IEEE Latin America Transactions*, vol. 9, pp. 743–748, March 2011.

[11] J. H. Wang and H. Y. Wang, "Incremental neural network construction for text classification," in *Computer, Consumer and Control (IS3C), 2014 International Symposium on*, pp. 970–973, June 2014.

[12] M. Ghiassi, M. Olschimke, B. Moon, and P. Arnaudo, "Automated text classification using a dynamic artificial neural network model," *Expert Systems with Applications*, vol. 39, no. 12, pp. 10967 – 10976, 2012.

[13] C. C. Aggarwal, ed., *Data Classification: Algorithms and Applications (Chapman & Hall/CRC Data Mining and Knowledge Discovery Series)*. Chapman and Hall/CRC, 1 ed., 7 2014.

[14] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. Wiley, 2 ed., 9 2014.

[15] F. M. Bianchi, S. Scardapane, A. Rizzi, A. Uncini, and A. Sadeghian, “Granular computing techniques for classification and semantic characterization of structured data,” *Cognitive Computation*, vol. 8, no. 3, pp. 442–461, 2016.

[16] L. Kuncheva, *Fuzzy Classifier Design (Studies in Fuzziness and Soft Computing)*.
Physica, softcover reprint of hardcover 1st ed. 2000 ed., 10 2010.

[17] D. M. J. Tax and R. P. W. Duin, “Using two-class classifiers for multiclass classification,” in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, vol. 2, pp. 124–127 vol.2, 2002.

[18] W. Pedrycz, “Granular computing: an introduction,” in *IFSA World Congress and 20th NAFIPS International Conference, 2001. Joint 9th*, vol. 3, pp. 1349–1354 vol.3, July 2001.

[19] J. T. Yao, A. V. Vasilakos, and W. Pedrycz, “Granular computing: Perspectives and challenges,” *IEEE Transactions on Cybernetics*, vol. 43, pp. 1977–1989, Dec 2013.

[20] H. Liu, A. Gegov, and M. Cocea, “Rule-based systems: a granular computing perspective,” *Granular Computing*, pp. 1–16, 2016.

[21] S. K. Meher, S. K. Pal, and S. Dutta, “Granular computing models in the classification of web content data,” in *Proceedings of the The 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology - Volume 02*, WI-IAT ’12, (Washington, DC, USA), pp. 175–179, IEEE Computer Society, 2012.

[22] F. M. Bianchi, L. Livi, A. Rizzi, and A. Sadeghian, “A granular computing approach to the design of optimized graph classification systems,” *Soft Computing*, vol. 18, no. 2, pp. 393–412, 2014.

[23] A. Rizzi and G. D. Vescovo, “Automatic image classification by a granular computing approach,” in *2006 16th IEEE Signal Processing Society Workshop on Machine Learning for Signal Processing*, pp. 33–38, Sept 2006.

[24] L. Polkowski and P. Artiemjew, *A Study in Granular Computing: On Classifiers Induced from Granular Reflections of Data*, pp. 230–263.
Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[25] M. M. Eissa, M. Elmogy, and M. Hashem, “Rough ‚Äì granular computing knowledge discovery models for medical classification,” *Egyptian Informatics Journal*, pp. –, 2016.

[26] H. Liu, S. Xiong, and Z. Fang, “Fl-grcca: A granular computing classification algorithm based on fuzzy lattices,” *Computers & Mathematics with Applications*, vol. 61, no. 1, pp. 138 – 147, 2011.

[27] H. Liu, S. Xiong, and C. an Wu, “Hyperspherical granular computing classification algorithm based on fuzzy lattices,” *Mathematical and Computer Modelling*, vol. 57, no. 3‚Äì4, pp. 661 – 670, 2013.

[28] J. T. Yao and Y. Y. Yao, *Induction of Classification Rules by Granular Computing*, pp. 331–338.
Berlin, Heidelberg: Springer Berlin Heidelberg, 2002.

[29] H. Liu, C. Liu, and C. an Wu, “Granular computing classification algorithms based on distance measures between granules from the view of set,” *Computational Intelligence and Neuroscience*, vol. 2014, 2014.

[30] X. Zhang, Y. Yin, X. Meng, and H. Zhao, “Text classification based on rule mining by granule network constructing,” in *Fuzzy Systems and Knowledge Discovery, 2008. FSKD ’08. Fifth International Conference on*, vol. 2, pp. 514–518, Oct 2008.

[31] P. Hońko, “Granular computing for relational data classification,” *Journal of Intelligent Information Systems*, vol. 41, pp. 187–210, 10 2013.

[32] T. Qiu, X. Chen, Q. Liu, and H. Huang, “Granular computing based text classification,” in *2006 IEEE International Conference on Granular Computing*, pp. 313–316, May 2006.

[33] F. Possemato and A. Rizzi, "Automatic text categorization by a granular computing approach: Facing unbalanced data sets," in *Neural Networks (IJCNN), The 2013 International Joint Conference on*, pp. 1–8, Aug 2013.

[34] L. Zadeh, "Fuzzy sets and information granularity," in *Advances in fuzzy set theory and applications* (M. M. Gupta, R. K. Ragade, and R. R. Yager, eds.), pp. 3–18, North-Holland Publishing Company, 1979.

[35] L. A. Zadeh, "Fuzzy sets: Where do we stand? where do we go? toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic," *Fuzzy Sets and Systems*, vol. 90, no. 2, pp. 111 – 127, 1997.

[36] Y. Y. Yao, "Granular computing," in *Chinese National Conference on Rough Sets and Soft Computing, 2004. 4th*, vol. 31, pp. 1–5 vol.31, 2004.

[37] "Merriam-webster. [online].." `http://www.merriam-webster.com/`. Accessed: 2016-06-02.

[38] W. Pedrycz and F. Gomide, *Fuzzy Systems Engineering: Toward Human-Centric Computing*.
Wiley-IEEE Press, 1 ed., 8 2007.

[39] W. Pedrycz and S.-M. Chen, eds., *Granular Computing and Intelligent Systems: Design with Information Granules of Higher Order and Higher Type (Intelligent Systems Reference Library)*.
Springer, 2011 ed., 5 2013.

[40] Y. Yao, "Perspectives of granular computing," in *2005 IEEE International Conference on Granular Computing*, vol. 1, pp. 85–90 Vol. 1, July 2005.

[41] Y. Yao, *Rough Sets and Intelligent Systems Paradigms: International Conference, RSEISP 2007, Warsaw, Poland, June 28-30, 2007. Proceedings*, ch. The Art of Granular Computing, pp. 101–112.
Berlin, Heidelberg: Springer Berlin Heidelberg, 2007.

[42] W. A. Lodwick, "Fundamentals of interval analysis and linkages to fuzzy set theory," in *Handbook of Granular Computing* (V. K. Witold Pedrycz, Andrzej Skowron, ed.), ch. 3, pp. 55–79, John Wiley & Sons, Ltd, 2008.

[43] W. Pedrycz and G. Vukovich, “Granular neural networks,” *Neurocomputing*, vol. 36, no. 1,Äì4, pp. 205 – 224, 2001.

[44] H. Tahayori, W. Pedrycz, and G. D. Antoni, “Distributed intervals: A formal framework for information granulation,” in *2007 Canadian Conference on Electrical and Computer Engineering*, pp. 1409–1412, April 2007.

[45] W. Pedrycz, B. Park, and S. Oh, “The design of granular classifiers: A study in the synergy of interval calculus and fuzzy sets in pattern recognition,” *Pattern Recognition*, vol. 41, no. 12, pp. 3720 – 3735, 2008.

[46] N. Zhang and X. Yue, *Rough Sets and Knowledge Technology: 9th International Conference, RSKT 2014, Shanghai, China, October 24-26, 2014, Proceedings*, ch. Knowledge Granulation in Interval-Valued Information Systems Based on Maximal Consistent Blocks, pp. 49–58.
Cham: Springer International Publishing, 2014.

[47] N. Zhong and J.-j. Huang, *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing: 15th International Conference, RSFDGrC 2015, Tianjin, China, November 20-23, 2015, Proceedings*, ch. Granular Structures Induced by Interval Sets and Rough Sets, pp. 49–60.
Cham: Springer International Publishing, 2015.

[48] J. Han and N. Cercone, “Principles, applications, and trends of granular computing,” in *Granular Computing (GrC), 2010 IEEE International Conference on*, pp. 24–25, Aug 2010.

[49] W. Pedrycz, “Fuzzy sets as a user-centric processing framework of granular computing,” in *Handbook of Granular Computing* (V. K. Witold Pedrycz, Andrzej Skowron, ed.), ch. 5, pp. 97–139, John Wiley & Sons, Ltd, 2008.

[50] S. M. Gu, S. X. Zhu, and Q. H. Ye, “An approach for constructing hierarchy of granules based on fuzzy concept lattices,” in *Fuzzy Systems and Knowledge Discovery, 2008. FSKD ’08. Fifth International Conference on*, vol. 5, pp. 679–684, Oct 2008.

[51] G. Bortolan and W. Pedrycz, “Fuzzy descriptive models: an interactive framework of information granulation [ecg data],” *IEEE Transactions on Fuzzy Systems*, vol. 10, pp. 743–755, Dec 2002.

[52] P. F. Castro and G. B. Xexéo, *Granules of Words to Represent Text: An Approach Based on Fuzzy Relations and Spectral Clustering*, pp. 379–391.
Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[53] A. Pedrycz, K. Hirota, W. Pedrycz, and F. Dong, “Granular representation and granular computing with fuzzy sets,” *Fuzzy Sets and Systems*, vol. 203, pp. 17 – 32, 2012.

[54] Z. Pawlak, “Rough sets,” *International Journal of Computer & Information Sciences*, vol. 11, no. 5, pp. 341–356, 1982.

[55] S. Salehi, A. Selamat, and H. Fujita, “Systematic mapping study on granular computing,” *Knowledge-Based Systems*, vol. 80, pp. 78 – 97, 2015.

[56] Z. Pawlak, “Granularity of knowledge, indiscernibility and rough sets,” in *Fuzzy Systems Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*, vol. 1, pp. 106–110 vol.1, May 1998.

[57] Z. Pawlak, *Some Issues on Rough Sets*, pp. 1–58.
Berlin, Heidelberg: Springer Berlin Heidelberg, 2004.

[58] Y. Yao, P. Lingras, R. Wang, and D. Miao, *Interval Set Cluster Analysis: A Re-formulation*, pp. 398–405.
Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.

[59] J. Zhao and L. Liu, “Construction of concept granule based on rough set and representation of knowledge-based complex system,” *Knowledge-Based Systems*, vol. 24, no. 6, pp. 809 – 815, 2011.

[60] Y. Liu and W. Zhu, “On three types of covering-based rough sets via definable sets,” in *2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 1226–1233, July 2014.

[61] Z. Pawlak and A. Skowron, “Advances in the dempster-shafer theory of evidence,” ch. Rough Membership Functions, pp. 251–271, New York, NY, USA: John Wiley & Sons, Inc., 1994.

[62] Z. Pawlak, “Rough sets, rough relations and rough functions,” *Fundam. Inf.*, vol. 27, pp. 103–108, Aug. 1996.

[63] L. Polkowski, “A model of granular computing with applications. granules from rough inclusions in information systems,” in *2006 IEEE International Conference on Granular Computing*, pp. 9–16, May 2006.

[64] Y. Yao, *Information Granulation and Approximation in a Decision-Theoretical Model of Rough Sets*, pp. 491–516.
Berlin, Heidelberg: Springer Berlin Heidelberg, 2004.

[65] P. Hońko, *Rough-Granular Computing Based Relational Data Mining*, pp. 290–299.
Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[66] L. Polkowski, *Granulation of Knowledge in Decision Systems: The Approach Based on Rough Inclusions. The Method and Its Applications*, pp. 69–79.
Berlin, Heidelberg: Springer Berlin Heidelberg, 2007.

[67] Y. Qian, J. Liang, Y. Yao, and C. Dang, “Mgrs: A multi-granulation rough set,” *Information Sciences*, vol. 180, no. 6, pp. 949 – 970, 2010.

[68] W. Xu, Q. Wang, and S. Luo, “Multi-granulation fuzzy rough sets.,” *Journal of Intelligent & Fuzzy Systems*, vol. 26, no. 3, pp. 1323 – 1340, 2014.

[69] T. Feng and J.-S. Mi, “Variable precision multigranulation decision-theoretic fuzzy rough sets,” *Knowledge-Based Systems*, vol. 91, pp. 93 – 101, 2016.

[70] W. Wei, J. Liang, Y. Qian, and F. Wang, “Variable precision multi-granulation rough set,” in *Granular Computing (GrC), 2012 IEEE International Conference on*, pp. 536–540, Aug 2012.

[71] W. Ziarko, “Variable precision rough set model,” *Journal of Computer and System Sciences*, vol. 46, no. 1, pp. 39 – 59, 1993.

[72] H. Dou, X. Yang, J. Fan, and S. Xu, *The Models of Variable Precision Multigranulation Rough Sets*, pp. 465–473.
Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[73] G. Lin, J. Liang, and Y. Qian, "Multigranulation rough sets: From partition to covering," *Information Sciences*, vol. 241, pp. 101 – 118, 2013.

[74] C. Liu, D. Miao, and J. Qian, "On multi-granulation covering rough sets," *International Journal of Approximate Reasoning*, vol. 55, no. 6, pp. 1404 – 1418, 2014.

[75] J. Xie, T. Y. Lin, and W. Zhu, "Granular and rough computing on covering," in *Granular Computing (GrC), 2012 IEEE International Conference on*, pp. 547–552, Aug 2012.

[76] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques, Third Edition (The Morgan Kaufmann Series in Data Management Systems)*.
Morgan Kaufmann, 3 ed., 7 2011.

[77] S. Ding, M. Du, and H. Zhu, "Survey on granularity clustering," *Cognitive Neurodynamics*, vol. 9, no. 6, pp. 561–572, 2015.

[78] X. Wang, W. Pedrycz, A. Gacek, and X. Liu, "From numeric data to information granules: A design through clustering and the principle of justifiable granularity," *Knowledge-Based Systems*, vol. 101, pp. 100 – 113, 2016.

[79] P. Fazendeiro and J. Valente de Oliveira, *Fuzzy Clustering as a Data-Driven Development Environment for Information Granules*, pp. 153–169.
John Wiley & Sons, Ltd, 2008.

[80] A. Balamash, W. Pedrycz, R. Al-Hmouz, and A. Morfeq, "An expansion of fuzzy information granules through successive refinements of their information content and their use to system modeling," *Expert Systems with Applications*, vol. 42, no. 6, pp. 2985 – 2997, 2015.

[81] G. Castellano, A. M. Fanelli, and C. Mencar, *Fuzzy Information Granulation with Multiple Levels of Granularity*, pp. 185–202.
Berlin, Heidelberg: Springer Berlin Heidelberg, 2011.

[82] W. Pedrycz, “A dynamic data granulation through adjustable fuzzy clustering,” *Pattern Recognition Letters*, vol. 29, no. 16, pp. 2059 – 2066, 2008.

[83] A. P. Engelbrecht, *Computational Intelligence: An Introduction*.
Wiley, 2 ed., 11 2007.

[84] N. Siddique and H. Adeli, *Computational Intelligence: Synergies of Fuzzy Logic, Neural Networks and Evolutionary Computing*.
Wiley, 1 ed., 5 2013.

[85] S. Sakinah, S. Ahmad, and W. Pedrycz, “Fuzzy rule-based system through granular computing,” in *2013 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 800–805, Oct 2013.

[86] X. Wang, X. Liu, and L. Zhang, “A rapid fuzzy rule clustering method based on granular computing,” *Applied Soft Computing*, vol. 24, pp. 534 – 542, 2014.

[87] D. Guliato and J. C. de Sousa Santos, *Granular Computing and Rough Sets to Generate Fuzzy Rules*, pp. 317–326.
Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.

[88] S. Ding, H. Jia, J. Chen, and F. Jin, “Granular neural networks,” *Artificial Intelligence Review*, vol. 41, no. 3, pp. 373–384, 2014.

[89] M. Song and W. Pedrycz, “Granular neural networks: Concepts and development schemes,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, pp. 542–553, April 2013.

[90] D. Stathakis and A. Vasilakos, “Satellite image classification using granular neural networks,” *International Journal of Remote Sensing*, vol. 27, no. 18, pp. 3991–4003, 2006.

[91] A. Vasilakos and D. Stathakis, “Granular neural networks for land use classification,” *Soft Computing*, vol. 9, no. 5, pp. 332–340, 2005.

[92] D. Leite, P. Costa, and F. Gomide, “Evolving granular neural networks from fuzzy data streams,” *Neural Networks*, vol. 38, pp. 1 – 16, 2013.

[93] S. Dick, A. Tappenden, C. Badke, and O. Olarewaju, "A granular neural network: Performance analysis and application to re-granulation," *International Journal of Approximate Reasoning*, vol. 54, no. 8, pp. 1149 – 1167, 2013.

[94] X. Xu, G. Wang, S. Ding, X. Jiang, and Z. Zhao, "A new method for constructing granular neural networks based on rule extraction and extreme learning machine," *Pattern Recognition Letters*, vol. 67, Part 2, pp. 138 – 144, 2015.

[95] A. Skowron, S. K. Pal, H. S. Nguyen, A. Ganivada, S. Dutta, and S. K. Pal, "Rough sets and fuzzy sets in natural computing fuzzy rough granular neural networks, fuzzy granules, and classification," *Theoretical Computer Science*, vol. 412, no. 42, pp. 5834 – 5853, 2011.

[96] A. Ganivada and S. K. Pal, "A novel fuzzy rough granular neural network for classification," *International Journal of Computational Intelligence Systems*, vol. 4, no. 5, pp. 1042–1051, 2011.

[97] J. Morente-Molinera, R. Al-Hmouz, A. Morfeq, A. S. Balamash, and E. Herrera-Viedma, "A decision support system for decision making in changeable and multi-granular fuzzy linguistic contexts.," *Journal of Multiple-Valued Logic & Soft Computing*, vol. 26, 2016.

[98] F. J. Cabrerizo, R. Ureña, J. A. Morente-Molinera, W. Pedrycz, F. Chiclana, and E. Herrera-Viedma, *A New Selection Process Based on Granular Computing for Group Decision Making Problems*, pp. 13–24.
Cham: Springer International Publishing, 2015.

[99] J. Morente-Molinera, I. Pèrez, M. Ureña, and E. Herrera-Viedma, "On multi-granular fuzzy linguistic modeling in group decision making problems: A systematic review and future trends," *Knowledge-Based Systems*, vol. 74, pp. 49 – 60, 2015.

[100] S. H. Nguyen, T. T. Nguyen, M. Szczuka, and H. S. Nguyen, "An approach to pattern recognition based on hierarchical granular computing.," *Fundamenta Informaticae*, vol. 127, no. 1-4, pp. 369 – 384, 2013.

[101] A. Skowron, J. Bazan, and M. Wojnarski, *Interactive Rough-Granular Computing in Pattern Recognition*, pp. 92–97.
Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.

[102] H. Hu and Z. Shi, "Granular computing in the information transformation of pattern recognition," in *Granular Computing, 2007. GRC 2007. IEEE International Conference on*, pp. 36–36, Nov 2007.

[103] A. V. Nandedkar, "Supervised colour image segmentation using granular reflex fuzzy min-max neural network," in *Second International Conference on Digital Image Processing*, pp. 75460T–75460T, International Society for Optics and Photonics, 2010.

[104] K. Xie, X. Hao, and J. Xie, "Image segmentation algorithm based on granular lattice matrix space," in *Granular Computing, 2009, GRC '09. IEEE International Conference on*, pp. 616–619, Aug 2009.

[105] X. Xinying, Z. Zhijun, X. Jun, and X. Keming, "Threshold image segmentation based on granular immune algorithm," in *2009 Chinese Control and Decision Conference*, pp. 3512–3515, June 2009.

[106] L. Zhong and J. Wu, "Granular computing applied to data-mining of tunnel information," in *Education Technology and Computer Science, 2009. ETCS '09. First International Workshop on*, vol. 1, pp. 670–674, March 2009.

[107] A. Wasilewska, "Descriptive data mining; a granular model," in *Fuzzy Information Processing Society, 2008. NAFIPS 2008. Annual Meeting of the North American*, pp. 1–5, May 2008.

[108] T. Y. Lin, "Granular data model: semantic data mining and computing with words," in *Fuzzy Systems, 2004. Proceedings. 2004 IEEE International Conference on*, vol. 2, pp. 1141–1146 vol.2, July 2004.

[109] E. Bair, "Semi-supervised clustering methods," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 5, no. 5, pp. 349–361, 2013.

[110] A. Gacek and W. Pedrycz, “Clustering granular data and their characterization with information granules of higher type,” *IEEE Transactions on Fuzzy Systems*, vol. 23, pp. 850–860, Aug 2015.

[111] A. Gacek, “From clustering to granular clustering: A granular representation of data in pattern recognition and system modeling,” in *IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), 2013 Joint*, pp. 502–506, June 2013.

[112] K. Rathinavel and P. Lingras, “A granular recursive fuzzy meta-clustering algorithm for social networks,” in *IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS), 2013 Joint*, pp. 567–572, June 2013.

[113] W. Pedrycz, V. Loia, and S. Senatore, “Fuzzy clustering with viewpoints,” *IEEE Transactions on Fuzzy Systems*, vol. 18, pp. 274–284, April 2010.

[114] Y. Xie, V. V. Raghavan, P. Dhatric, and X. Zhao, “Data mining and granular computing a new fuzzy clustering algorithm for optimally finding granular prototypes,” *International Journal of Approximate Reasoning*, vol. 40, no. 1, pp. 109 – 124, 2005.

[115] M. A. Sanchez, O. Castillo, J. R. Castro, and P. Melin, “Fuzzy granular gravitational clustering algorithm for multivariate data,” *Information Sciences*, vol. 279, pp. 498 – 511, 2014.

[116] J. Zhou, W. Pedrycz, and D. Miao, “Shadowed sets in the characterization of rough-fuzzy clustering,” *Pattern Recognition*, vol. 44, no. 8, pp. 1738 – 1749, 2011.

[117] P. Lingras and F. Haider, *Combining Rough Clustering Schemes as a Rough Ensemble*, pp. 383–394.
Cham: Springer International Publishing, 2015.

[118] H. Yu, S. Chu, and D. Yang, *Autonomous Knowledge-Oriented Clustering Using Decision-Theoretic Rough Set Theory*, pp. 687–694.
Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.

[119] D. Malyszko and J. Stepaniuk, "Adaptive multilevel rough entropy evolutionary thresholding," *Information Sciences*, vol. 180, no. 7, pp. 1138 – 1158, 2010.

[120] H. S. Nguyen and T. B. Ho, *Rough Document Clustering and the Internet*, pp. 987–1003.
John Wiley & Sons, Ltd, 2008.

[121] W. Zhaocong, Y. Lina, and Q. Maoyun, *Granular Approach to Object-Oriented Remote Sensing Image Classification*, pp. 563–570.
Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.

[122] M. Zhang and J.-X. Cheng, "Pattern classification with granular computing," in *2005 IEEE International Conference on Systems, Man and Cybernetics*, vol. 1, pp. 336–340 Vol. 1, Oct 2005.

[123] A. Gacek and W. Pedrycz, "A characterization of electrocardiogram signals through optimal allocation of information granularity," *Artificial Intelligence in Medicine*, vol. 54, no. 2, pp. 125 – 134, 2012.

[124] Y. Tang, B. Jin, Y.-Q. Zhang, H. Fang, and B. Wang, "Granular support vector machines using linear decision hyperplanes for fast medical binary classification," in *The 14th IEEE International Conference on Fuzzy Systems, 2005. FUZZ '05.*, pp. 138–142, May 2005.

[125] Y. Tang and Y.-Q. Zhang, "Granular support vector machines with data cleaning for fast and accurate biomedical binary classification," in *2005 IEEE International Conference on Granular Computing*, vol. 1, pp. 262–265 Vol. 1, July 2005.

[126] Y. Tang, B. Jin, Y. Sun, and Y.-Q. Zhang, "Granular support vector machines for medical binary classification problems," in *Computational Intelligence in Bioinformatics and Computational Biology, 2004. CIBCB '04. Proceedings of the 2004 IEEE Symposium on*, pp. 73–78, Oct 2004.

[127] R. Al-Hmouz, W. Pedrycz, A. Balamash, and A. Morfeq, "From data to granular data and granular classifiers," in *2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pp. 432–438, July 2014.

[128] P. Artiemjew, *On Classification of Data by Means of Rough Mereological Granules of Objects and Rules*, pp. 221–228.
Berlin, Heidelberg: Springer Berlin Heidelberg, 2008.

[129] W. Pedrycz, B. Park, and S. Oh, "The design of granular classifiers: A study in the synergy of interval calculus and fuzzy sets in pattern recognition," *Pattern Recognition*, vol. 41, no. 12, pp. 3720 – 3735, 2008.

[130] C.-T. Su, L.-S. Chen, and Y. Yih, "Knowledge acquisition through information granulation for imbalanced data," *Expert Systems with Applications*, vol. 31, no. 3, pp. 531 – 541, 2006.

[131] S. D. Connell and A. K. Jain, "Writer adaptation for online handwriting recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 329–346, Mar 2002.

[132] N. Slonim and N. Tishby, "Document clustering using word clusters via the information bottleneck method," in *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, (New York, NY, USA), pp. 208–215, ACM, 2000.

[133] Z. Yu, H.-S. Wong, and H. Wang, "Graph-based consensus clustering for class discovery from gene expression data," *Bioinformatics*, vol. 23, no. 21, pp. 2888–2896, 2007.

[134] K. Dhiraj, S. K. Rath, and A. Pandey, "Gene expression analysis using clustering," in *2009 3rd International Conference on Bioinformatics and Biomedical Engineering*, pp. 1–4, June 2009.

[135] B. L. Quéau, O. Shafiq, and R. Alhajj, "Analyzing alzheimer's disease gene expression dataset using clustering and association rule mining," in *Information Reuse and Integration (IRI), 2014 IEEE 15th International Conference on*, pp. 283–290, Aug 2014.

[136] J. R. Manjarrez Sanchez, J. Martinez, and P. Valduriez, *On the Usage of Clustering for Content Based Image Retrieval*, pp. 281–289.
Berlin, Heidelberg: Springer Berlin Heidelberg, 2007.

[137] H. Xu, D. Xu, and E. Lin, *An Applicable Hierarchical Clustering Algorithm for Content-Based Image Retrieval*, pp. 82–92.
Berlin, Heidelberg: Springer Berlin Heidelberg, 2007.

[138] C. C. Aggarwal and C. K. Reddy, eds., *Data Clustering: Algorithms and Applications (Chapman & Hall/CRC Data Mining and Knowledge Discovery Series)*.
Chapman and Hall/CRC, 0 ed., 8 2013.

[139] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," pp. 226–231, AAAI Press, 1996.

[140] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, (Berkeley, Calif.), pp. 281–297, University of California Press, 1967.

[141] A. Likas, N. Vlassis, and J. J. Verbeek, "The global k-means clustering algorithm," *Pattern Recognition*, vol. 36, no. 2, pp. 451 – 461, 2003.
Biometrics.

[142] J. C. Dunn, "A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters," *Journal of Cybernetics*, vol. 3, no. 3, pp. 32–57, 1973.

[143] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*.
Norwell, MA, USA: Kluwer Academic Publishers, 1981.

[144] G. Gan, C. Ma, and J. Wu, *Data Clustering: Theory, Algorithms, and Applications (ASA-SIAM Series on Statistics and Applied Probability)*.
SIAM, Society for Industrial and Applied Mathematics, 5 2007.

[145] J. Nayak, B. Naik, and H. S. Behera, *Fuzzy C-Means (FCM) Clustering Algorithm: A Decade Review from 2000 to 2014*, pp. 133–149.
New Delhi: Springer India, 2015.

[146] J. C. Bezdek, R. Ehrlich, and W. Full, "Fcm: The fuzzy c-means clustering algorithm," *Computers & Geosciences*, vol. 10, no. 2, pp. 191 – 203, 1984.

[147] H. Guldemır and A. Sengur, "Comparison of clustering algorithms for analog modulation classification," *Expert Systems with Applications*, vol. 30, no. 4, pp. 642 – 649, 2006.

[148] M. Lichman, "UCI machine learning repository," 2013.

[149] C.-Y. Fan, P.-C. Chang, J.-J. Lin, and J. Hsieh, "A hybrid model combining case-based reasoning and fuzzy decision tree for medical data classification," *Applied Soft Computing*, vol. 11, no. 1, pp. 632 – 644, 2011.

[150] M.-C. Chen, L.-S. Chen, C.-C. Hsu, and W.-R. Zeng, "An information granulation based data mining approach for classifying imbalanced data," *Information Sciences*, vol. 178, no. 16, pp. 3214 – 3227, 2008.

[151] N. García-Pedrajas and D. Ortiz-Boyer, "An empirical study of binary classifier fusion methods for multiclass classification," *Information Fusion*, vol. 12, no. 2, pp. 111 – 130, 2011.

[152] E. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *International Journal of Man-Machine Studies*, vol. 7, no. 1, pp. 1 – 13, 1975.

[153] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-15, pp. 116–132, Jan 1985.

[154] M. J. Er and Y. Zhou, "Automatic generation of fuzzy inference systems via unsupervised learning," *Neural Networks*, vol. 21, no. 10, pp. 1556 – 1566, 2008.

[155] R. P. Prado, S. García-Galán, J. E. M. Exposito, and A. J. Yuste, “Knowledge acquisition in fuzzy-rule-based systems with particle-swarm optimization,” *IEEE Transactions on Fuzzy Systems*, vol. 18, pp. 1083–1097, Dec 2010.

[156] H. x. Zhang, B. Zhang, and F. Wang, “Automatic fuzzy rules generation using fuzzy genetic algorithm,” in *Fuzzy Systems and Knowledge Discovery, 2009. FSKD ’09. Sixth International Conference on*, vol. 6, pp. 107–112, Aug 2009.

[157] M. Sikora, *Fuzzy Rules Generation Method for Classification Problems Using Rough Sets and Genetic Algorithms*, pp. 383–391.
Berlin, Heidelberg: Springer Berlin Heidelberg, 2005.

[158] D. Wang, T. S. Dillon, and E. J. Chang, “A data mining approach for fuzzy classification rule generation,” in *IFSA World Congress and 20th NAFIPS International Conference, 2001. Joint 9th*, pp. 2960–2964 vol.5, July 2001.

[159] S. Haykin, *Neural Networks: A Comprehensive Foundation (2nd Edition)*.
Prentice Hall, 2 ed., 7 1998.

[160] K. L. Priddy and P. E. Keller, *Artificial Neural Networks: An Introduction (SPIE Tutorial Texts in Optical Engineering, Vol. TT68)*.
SPIE Publications, illustrated edition ed., 8 2005.

[161] M. Forouzanfar, H. R. Dajani, V. Z. Groza, M. Bolic, and S. Rajan, “Comparison of feed-forward neural network training algorithms for oscillometric blood pressure estimation,” in *Soft Computing Applications (SOFA), 2010 4th International Workshop on*, pp. 119–123, July 2010.

[162] J. Heaton, *Introduction to Neural Networks for Java, 2nd Edition*.
Heaton Research, Inc., 2 ed., 10 2008.

[163] G. Cybenko, “Approximation by superpositions of a sigmoidal function,” *Mathematics of Control, Signals and Systems*, vol. 2, no. 4, pp. 303–314, 1989.

[164] S. M. Kamruzzaman, M. A. Hamid, and A. M. J. Sarkar, “Erann: An algorithm to extract symbolic rules from trained artificial neural networks,” *IETE Journal of Research*, vol. 58, no. 2, pp. 138–154, 2012.

[165] L. Zarate, R. Vimieiro, and N. Vieira, “Reasoning based on rules extracted from trained neural networks via formal concept analysis,” in *2006 IEEE International Conference on Engineering of Intelligent Systems*, pp. 1–6, 2006.

[166] J. Chorowski and J. M. Zurada, “Extracting rules from neural networks as decision diagrams,” *IEEE Transactions on Neural Networks*, vol. 22, pp. 2435–2446, Dec 2011.

[167] J. S. R. Jang, “Anfis: adaptive-network-based fuzzy inference system,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, pp. 665–685, May 1993.

[168] J. Tukey, *Exploratory Data Analysis*.
Addison-Wesley series in behavioral science, Addison-Wesley Publishing Company, 1977.

[169] B. V. Dasarathy and B. V. Sheela, “A composite classifier system design: Concepts and methodology,” *Proceedings of the IEEE*, vol. 67, pp. 708–713, May 1979.

[170] A.-m. Yang, Y.-m. Zhou, and M. Tang, *A Classifier Ensemble Method for Fuzzy Classifiers*, pp. 784–793.
Berlin, Heidelberg: Springer Berlin Heidelberg, 2006.

[171] S. Masarat, H. Taheri, and S. Sharifian, “A novel framework, based on fuzzy ensemble of classifiers for intrusion detection systems,” in *Computer and Knowledge Engineering (ICCKE), 2014 4th International eConference on*, pp. 165–170, Oct 2014.

[172] V. B. Kobayashi and V. B. Calag, “Detection of affective states from speech signals using ensembles of classifiers,” in *Intelligent Signal Processing Conference 2013 (ISP 2013), IET*, pp. 1–9, Dec 2013.

[173] Z. H. Kilimci, S. Akyokus, and S. I. Omurca, “The effectiveness of homogenous ensemble classifiers for turkish and english texts,” in *2016 International*

*Symposium on INnovations in Intelligent SysTems and Applications (INISTA)*, pp. 1–7, Aug 2016.

[174] G. Brown, J. Wyatt, R. Harris, and X. Yao, “Diversity creation methods: a survey and categorisation,” *Information Fusion*, vol. 6, no. 1, pp. 5 – 20, 2005.

[175] L. Breiman, “Bagging predictors,” *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[176] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms (Chapman & Hall/CRC Data Mining and Knowledge Discovery Serie)*.
Chapman and Hall/CRC, 1 ed., 6 2012.

[177] D. H. Wolpert, “Stacked generalization,” *Neural Networks*, vol. 5, no. 2, pp. 241 – 259, 1992.

[178] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, “Adaptive mixtures of local experts,” *Neural Comput.*, vol. 3, pp. 79–87, Mar. 1991.

[179] S. S. Rao, *Engineering Optimization: Theory and Practice*.
Wiley, 4 ed., 7 2009.

[180] S. Sivanandam and S. N. Deepa, *Introduction to Genetic Algorithms*.
Springer, 2008 ed., 12 2007.