Machine Learning Ensemble Methods for Classifying Multi-media Data

Saleh Alyahyan

A thesis submitted for the degree of
Doctor of Philosophy
at the University of East Anglia
September 2020

Machine Learning Ensemble Methods for Classifying Multi-media Data

Saleh Alyahyan

Abstract

Multimedia data have, over recent years, been produced in many fields. They have important applications for such diverse areas as social media and healthcare, due to their capacity to capture rich information. However, their unstructured and separated nature gives rise to various problems. In particular, fusing and integrating multi-media datasets and finding effective ways to learn from them have proven to be major challenges for machine learning.

In this thesis we investigated the development of the ensemble methods for classifying multi-media data in two key aspects: data fusion and model selection. For the data fusion, we devised two different strategies. The first one is the Feature Level Ensemble Method (FLEM) that aggregates all the features into a single dataset and then generates the models to build ensembles using this dataset. The second one is the Decision Level Ensemble Method (DLEM) that generates the models from each sub dataset individually and then aggregates their outputs with a decision fusion function. For the model selection we derived four different model selection rules. The first rule, R0, uses just the accuracy to select models. The rules R1 and R2 use firstly accuracy and then diversity to select models. In R3, we defined a generalised function that combines the accuracy and diversity with different weights to select models to build an ensemble.

Our methods were compared with existing well known ensemble methods using the same dataset and another dataset that became available after our methods had been developed. The results were critically analysed and the statistical significance analyses of the results show that our methods had better performance in general and the generalised R3 is the most effective rule in building ensembles.

Acknowledgements

Firstly, I would like to express the deepest appreciation to my primary supervisor, Dr Wenjia Wang, for unlimited advice, guidance and encouragement he has provided me during my PhD journey. I have been incredibly fortunate to have his supervision. He has shown great care about my work and responded to my questions and queries so promptly. Without his guidance and persistent help this thesis would not have been possible. It is a great honour for me to do this research under his supervision.

Secondly, I would also like to thank my secondary supervisor, Professor Anthony Bagnall, for his valuable comments and meaningful guidance. In addition, appreciation is due to all my collages in the School of Computing Science at University of East Anglia for their supporting during the PhD journey. In particular, I am grateful to Dr Majed Farrash for enlightening me the first glance of research Chapter4, and Dr Geoffrey Guile for proofreading the thesis.

Last but not the least, besides people at the university, my gratefulness and love go to my parents, wife and all of my family for their support, patience and love.

List of Publications:

- **Alyahyan, S.**, Farrash, M. & Wang, W. (2016) Heterogeneous Ensemble for Imaginary Scene Classification. In *Proceedings of the* 8^{th} *International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, pages 197-204.

- **Alyahyan, S.** & Wang, W. (2017) Feature Level Ensemble Method for Classifying Multi-Media Data. In *Artificial Intelligence XXXIV*, pages 235-249.

- **Alyahyan, S.** & Wang, W. (2018) Decision level ensemble method for classifying multi-media data. In *Wireless Networks*.

- **Alyahyan, S.** & Wang, W. (2018) Generalised Decision Level Ensemble Method for Classifying Multi-media Data. In *Artificial Intelligence XXXV*, pages 326-339. (This paper was awarded for the prize of the best PhD student paper at the Thirty-eighth SGAI International Conference on Artificial Intelligence, and we have been invited to submit an extended version for publication in *Künstliche Intelligenz*.)

Acronyms:

HES Heterogeneous Ensemble System.

HEST Heterogeneous Ensemble System for Text dataset.

HESG Heterogeneous Ensemble System for Graphics dataset.

FLEM Feature Level Ensemble Method.

DLEM Decision Level Ensemble Method.

MMD Multi-media Dataset.

D_t Text Dataset.

D_g Graphics Dataset.

Φ Ensemble.

m_i Model with index i.

Acc Model Accuracy.

PM Pool of Models.

MDM Maximum Divers Model.

MAM Maximum Accurate Model.

CFD Coincident Failure Diversity.

DF Double-Fault.

HOMOFLEM Homogeneous ensemble method used FLEM model selections rules

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

In recent years, as computing sciences and electronic technologies have advanced rapidly, the information collected that relates to a problem often involves multiple media. That is, the data can be represented by multiple datasets in many different formats, such as numbers, text, images, audio and video. For example as shown in Fig 1.1, in healthcare, in order to diagnose a complex disease, several tests and screenings may need to be carried out to collect the information on a patient. They may include: descriptions of symptoms—textual data, blood test and temperature measurments—numerical values, X-ray or MRI screenings—image data, ECG or EEG tests—time series data, endoscopy—video data, and so on. These datasets need to be analysed by some domain experts all together in order to make a more accurate diagnosis. However, when applying artificial intelligence and machine learning, both old and modern techniques and algorithms face many challenges in using these multi-media datasets effectively and efficiently. The key issues are (1) how to merge multi-media datasets without losing useful information and (2) how to apply machine learning to these datasets in order to generate

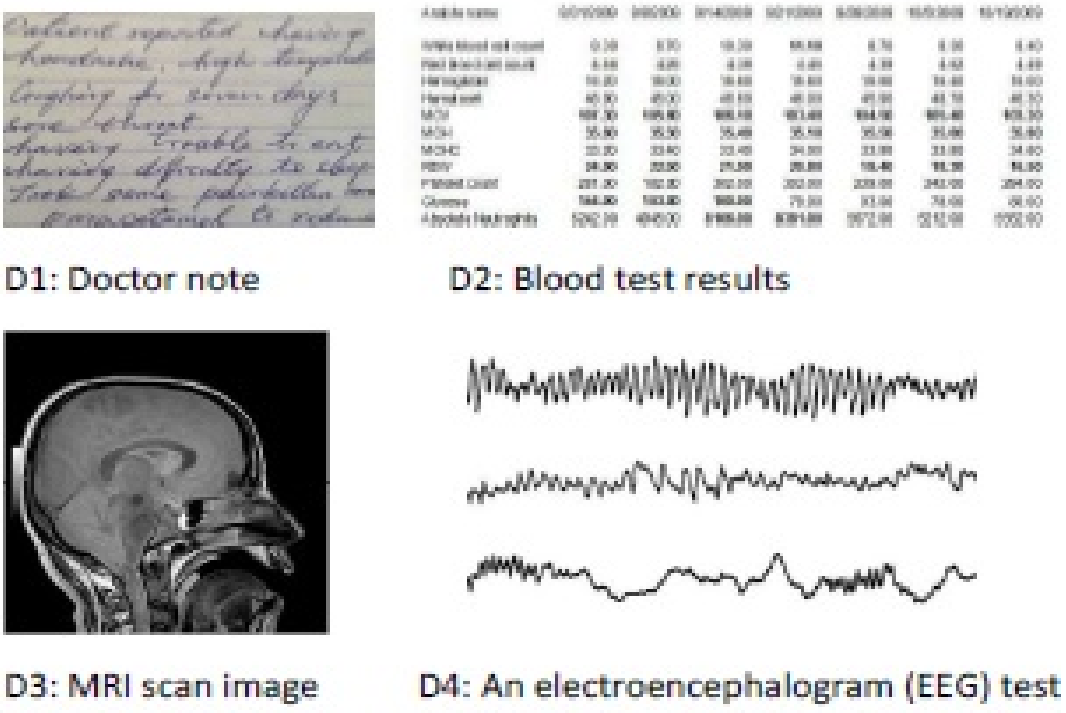

Figure 1.1: An example of multimedia medical data.

more accurate and reliable models.

1.2 Motivation

In order to tackle these two issues, many researchers have applied machine learning methods on multi-media data, for example for text classification (Williams and Gong, 2014), image classification (Grauman and Darrell, 2005), video classification (Karpathy et al., 2014) and time series classification (Lines et al., 2012). However, the majority of the researchers have not included more than one kind of media in their research. In addition, in these studies, they primarily used single methods and their results show that individual models are overwhelmed by the complexity and sheer quantity of multi-media data. Mining these kinds of data using ensemble approaches should be more effective for two reasons. Firstly,

many studies have shown that the performance of an ensemble is better than the performance of an individual model (Dietterich, 2000; Breiman, 2001; De Stefano et al., 2011; Zhang and Zhou, 2011). Secondly, the heterogeneous characteristics of multimedia datasets (MMDs) may provide ensemble models with more diversity due to the variety of types of data used.

Therefore, in this research, we will apply the ensemble paradigm to solve these two issues. In particular, we will attempt to explore and utilise the diverse characteristics of MMD to construct machine learning ensembles with two different stages, which are feature level ensemble (FLE) and decision level ensemble (DLE).

1.3 Research Aim and Objectives

This research aims to investigate ensemble techniques for classifying MMDs, and we have set the following objectives:

1. To identify the best procedure for transforming and / or combining several multi-media data sets into a form suitable for use by ensemble classification methods.

2. To develop a methodology for building an effective ensemble classifier for MMDs at two levels, feature level and decision level.

3. To test and critically evaluate our new implemented methods.

Note: Regarding to the last objective, because no previous researchers have worked with combined data types in the way we intend to, there are no existing methods with which we will be able to compare ours.

1.4 Research Questions

The main question in this research is:

How can we develop effective ensemble methods to classify MMD?

This research considers the following questions associated with the main question:

1. Given multimedia datasets, how can we transform them and integrate them into a single dataset?
2. How can we effectively use them?
3. What factors affect the performance of feature level ensemble systems for classifying MMD?
4. What factors affect the performance of decision level ensemble systems for classifying MMD?

The factors that will be investigated include: (1) accuracy of individual models, (2) diversity among the models, and (3) the number of models used in an ensemble.

1.5 The Process of the Research and Outcomes

For achieving our objectives in this research, we will follow these phases:

- Phase 1: Background study and literature review.
  - Studying basics in data mining.
  - Literature review on ensemble methods.

  - Literature review on multi-media data mining.

- Phase 2: Refinement of the research proposal and initial investigation.
  - To analyse existing research.
  - To refine the proposal for our research.

- Phase 3: Design of research methodology and experimental framework.
  - Design the ensemble framework for the experiments.
  - Collect data and pre-process data.
  - Choose appropriate methods/software packages for feature extraction.
  - Select learning algorithms.

- Phase 4: Implementation and experiments.
  - Design and implementation of software platform for experiments.
  - Evaluation of the framework with benchmark datasets.

- Phase 5: Writing up PhD thesis.
  - Discuss the results obtained.
  - Writing up the thesis.

1.6 Novelty and Contribution of the Research

Most previous work on the application of classification methods to multimedia data has, in fact, only used one type of data, for example text or images. Our proposed research is completely novel in that it will use more than one type of

data, for example, text and image data, together. This research, if carried out as planned, is expected to improve the understanding of ensemble techniques and their appliction to multimedia datasets. We expect it to generate the following contributions:

1. An optimum procedure for combining more than one type of data into a single dataset for analysis by ensemble classification methods.

2. An effective heterogeneous ensemble system to classify MMDs at the feature level (FLE).

3. An effective heterogeneous ensemble system to classify MMDs at the decision level (DLE).

4. A generalized rule system using any number or type of criteria for effective model selection for inclusion in an ensemble.

1.7 Thesis Structure

The remainder of this thesis is structured as follows:

Chapter 2: Literature review. This chapter reviews existing work relevant to this thesis. It focuses on presenting an overview of data mining, ensemble methods and multi-media data classification.

Chapter 3: Methodology. This chapter presents the research methodology and design. It also describes the datasets used in all the experiments we performed.

Chapter 4: Heterogeneous ensemble for Image Scene Classification. This chapter presents an empirical investigation of applying a heterogeneous ensemble system (HES) to classify image scenes. In addition, it presents three different rules for model selection based on individual model accuracy, pairwise diversity and the diversity among all candidate models.

Chapter 5: Feature Level Ensemble Method. This chapter presents an empirical investigation of applying FLEM.

Chapter 6: Generalised Decision Level Ensemble Method. This chapter presents an empirical investigation of applying GDLEM. Thus, it focuses on some issues raised in this system, especially model selection and decision fusion problems.

Chapter 7: General Discussion. In this chapter we present an overall discussion and evaluation of the work undertaken for this thesis, and the results obtained.

Chapter 8: Conclusions. We present our overall conclusions and give some suggestions for future work to extend this research.

Chapter 2

Literature Review

2.1 Introduction

This chapter gives an overview of the basics of data mining and machine learning, then reviews the related work on ensemble methods and multi-media data mining.

2.2 Data Mining

"Data mining is the process of automatically discovering useful information in large data repositories"(Tan et al., 2006). Since the 1990's data mining has been used to discover useful knowledge and information from large data sets, so it is called Knowledge Discovery in Databases (KDD) (Bian, 2006). There are several techniques used in data mining including classification, clustering, regression and association. In this research we will give a brief description of all these techniques but we will focus on classification since it is the main focus of our research.

Classification is one of the fundamental supervised learning methods used in data mining. The aim of classification is to learn from a training dataset which has its class labelled in advance, and then apply what has been learned from the training data set to a new dataset to get its class label value (Shah and Limbad,

2015). There are many methods and techniques for classification, including k-nearest neighbour (KNN), decision trees, Naive Bayesian methods and neural networks.

Clustering is an unsupervised learning method used in data mining. Clustering methods and techniques work to divide a dataset into meaningful sub-sets and each sub-set has its characteristics that show differences from the other sub-sets (Chen et al., 2015). A good cluster must offer high correlation between objects in the same sub-set and weak correlation between objects in different sub-sets (Bian, 2006).

Regression is a statistical method that tries to discover a relationship between one dependent variable and one or more independent variables (Tomar and Agarwal, 2013). Regression is appropriate for analyzing quantitative data (Bian, 2006).

Association rule methods aim to determine association relationships between objects in the dataset (Gosain and Bhugra, 2013). It is mostly used in market basket analysis or transaction data analysis (Chen et al., 2015).

2.3 Classification

Classification is a data mining technique that assigns items in collection to target categories or classes, and is useful for predicting group membership for data instances. Classification has the goal of accurately predicting the target class for each case of the data. Categorical class labels, which are either discrete or normal can be predicted and data classified, based on the training sets and class label, or values. The resulting classifying attribute can then be used to generate a model

which can classify new data (Phyu, 2009; Han et al., 2011; Tan et al., 2006).

Since target classes for each case in the data can be accurately predicted, and the dataset at the start of the classification task has unknown class assignments, one example of the use of a classification model is in the area of credit rating. Classification can take observed data for loan applicants over a long period and use it in identifying those who are low, medium or high credit risks. Other factors, such as home ownership or renting, work history and investment history may also be taken into account. In this case, the target of the classification is credit rating and the predictors are the other attributes. One case consists of the data for each customer.

There are two main steps in classification: model construction (learning step, or training step), and model usage (classifying future or unknown objects). Model construction involves the description of a set of classes which are predetermined. In each case it is assumed that the record is part of a predefined class, in accordance with the class label attribute. The training set is the set of instances for the model construction. Classification rules, mathematical formulae or decision trees may be used to represent the model. Model usage allows the classification of unknown objects. For this it is necessary to estimate the model's accuracy through a comparison of a known sample label and the classified results from the model. It is important to keep the training set separate from the test set, in order to avoid over-fitting. If the level of accuracy is adequate, the model can be used to classify data objects with unknown class labels.

The implementation of machine learning classification techniques has been

applied in several fields of study including: healthcare (Seera and Lim, 2014; Huang and Zheng, 2006; Meng et al., 2013), social media (Abboute et al., 2014; Shoeb and Ahmed, 2017; Salloum et al., 2017), education (Sobolevsky et al., 2014; Baradwaj and Pal, 2011) and economic (Sobolevsky et al., 2014; Ghose and Ipeirotis, 2011)

2.3.1 Algorithms Used for Classification

There are many different algorithms used for classification. Some of the most commonly used types are decision trees, artificial neural networks, support vector machines and K-nearest neighbour. These are described below.

2.3.1.1 Decision Trees

A decision tree (DT) algorithm is a supervised machine learning technique that builds a tree structure to represent the decision making process. A decision tree has three types of nodes: a root node, internal nodes and leaf nodes. The decision process starts from the root node going to leaf nodes through internal nodes (Rokach and Maimon, 2014).

A decision tree can be automatically generated with a learning algorithm in a relatively efficient way and a tree is usually easy for humans to interpret. Some common DT algorithms include C4.5, C5 and CART.

2.3.1.2 Artificial Neural Networks

Artificial neural networks (ANN) were originally inspired from the biological neural system where each nerve cell is connected with many others via axons. Axons

end with dendrites that connect with dendrites of the other cells, and their connecting points are called synapses (Tan et al., 2006). There are several types of ANN, such as perceptions, feedforward networks, concurrent network (Hopfield network) and deep learning networks. A feedforward ANN usually has three layers of neurons which are input layer, hidden layer and output layer (Michie et al., 1994).

2.3.1.3 Support Vector Machines

Support Vector Machines (SVM) are one of the most popular classification techniques. The background of this technique is statistical learning theory (Tan et al., 2006). SVM works as a hyperplane that linearly separate binary sets. The best hyperplane is that which has the maximum margin between the two classes in the dataset.

2.3.1.4 K-Nearest Neighbour

K-Nearest Neighbour algorithm (KNN) is an easy and effective algorithm. KNN needs a number of training vectors which are used to identify the K nearest neighbours for the selected object regardless of the label of the object (Cover and Hart, 1967). KNN methods are based on similarity metrics and the Euclidean distance is one of the most popular.

2.4 Ensemble Classification

In the context of machine learning, the ensemble of classifiers method is a combination of predictions, produced by multiple classifiers, with the aim of discovering

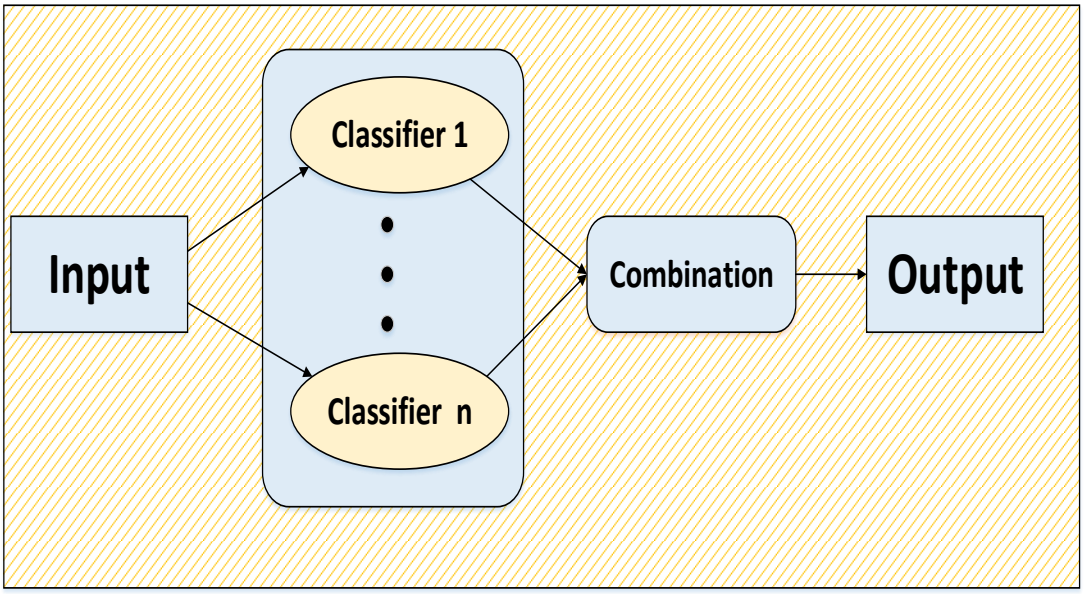

Figure 2.1: Ensemble construction (Woźniak et al., 2014a)

a better solution for the classification problem than that of any individual classifier (Dietterich, 2000; Kotsiantis et al., 2007; Gomes et al., 2017; Veni and Rani, 2014; Keedwell and Narayanan, 2009). It is widely recognized that ensembles outperform individual models most of the time, as it is shown empirically in (Breiman, 1996; Dietterich, 2000; Kolter and Maloof, 2003; Bifet et al., 2009; Mishra, 2013; Matikainen et al., 2012).

2.4.1 Constructing an Ensemble

Construction of an ensemble system to solve a classification problem can be performed using a number of different different methods. These methods aim to manipulate the training dataset, or the base classifier algorithms, or combine different base classifiers (Dietterich, 2000; Jurek et al., 2014; Mendes-Moreira et al., 2012; Wang, 2008). This section provides an overview of the generally applicable methods that can be used with a range of learning algorithms.

2.4.1.1 Manipulating the Training Set

Constructing an ensemble by manipulating the training set method involves a re-sampling of the original data for the purpose of producing multiple training sets. Each of these training sets is then used to build a classifier by means of a specific learning algorithm. According to some research, this method is particularly applicable to those learning algorithms which are not stable. These include neural networks, decision trees and rule learning algorithms. In contrast, linear regression, linear thresholds and nearest neighbours are usually very stable (Dietterich, 2000). Two ensemble methods which manipulate their training sets are Bagging and Boosting (Tan et al., 2006).

2.4.1.2 Manipulating the Input Features

This approach involves selecting a subset of input features from the training set, either randomly or using a specific method. This is especially useful where datasets have a high level of redundant features. One example of an ensemble method which manipulates its input features is random forest. Decision trees are used in this method as its base classifiers (Tan et al., 2006).

2.4.1.3 Manipulating the Class Labels

Manipulating class labels, for example error-correcting output coding method, is an approach which proves useful where there is a high number of classes. Random partitioning of the class labels is used to transform the data into the form of a binary class problem. Thus two disjointed subsets are formed (Tan et al., 2006).

2.4.1.4 Manipulating the Learning Algorithm

Most ensemble learning systems generate homogeneous ensembles, which use just one type of learning algorithm. For example, an ensemble of ANNs can be constructed by altering the network topology a number of times and generating an individual ANN each time. Initial weights and neuron links can also be changed (Tan et al., 2006).

2.4.2 Types of Ensemble

There are two types of ensemble: homogeneous and heterogeneous.

2.4.2.1 Homogeneous ensemble

Methods combining multiple models that are created from the same base-classifier are called homogeneous ensemble methods (Mendes-Moreira et al., 2012).

Bagging

Bagging (**b**ootstrap **agg**regat**ing**) is a classic homogeneous ensemble technique for generating many predictors and combining them together in a simple way to make a better prediction. It is a technique for learning from many classifiers, each using only portions of the data and then combining them through a model averaging technique. The idea in bagging is that we have a particular dataset and generate similar sub-datasets by sampling that with replacement (Breiman, 1996).

Boosting

Boosting works on the principle that a weak learning algorithm, which is only slightly better than random guessing in terms of performance, can be improved by

means of a set of iterative trainings. It is thus "boosted" and becomes a strong learning algorithm. Boosting algorithms vary according to how their training data points are weighted.

A popular boosting algorithm is AdaBoost (Rätsch et al., 2001). Its implementation procedure begins with a training set in which a uniform selection bias is used for each observation. A number of adaptive iterations are then carried out. A bootstrap sample is then generated based on random selection. Thus each observation has the same probability of being selected as part of the bootstrap sample for which a hypothesis is created by a new base learner. The selection bias for each wrongly classified observation can be increased using the results obtained by testing the hypothesis. Increasingly difficult samples are then focussed on by subsequent base learners. The predictions of base classifiers are then used by the final classifier in classification problems. Less weight, in the final voting, is given to the weak base learners. In the case of regression problems, a weighted mean from the base classifiers is taken by the final classifier. Better ensembles are generally produced where base learners are random and diverse.

Random forest

Random forest operates efficiently when datasets are large because it is an ensemble of decision trees. It can also deal with a large number of features (Kulkarni and Sinha, 2012). Randomness is generated firstly by selecting a random training bootstrap set for each decision tree base learner. The second factor introducing randomness is that the feature used to split the trees at the nodes and grow it to the next level will be randomly selected from the training set.

2.4.2.2 Heterogeneous Ensemble

Heterogeneous ensembles combine models created using different phase classifier methods. Heterogeneous ensembles are useful for classification especially when we do not know which base classifier is better to use for solving a specific problem, for example, when we do not know whether SVM, ANN, DT or NB would be the better method (Lertampaiporn et al., 2013; Haque et al., 2016; Tsoumakas et al., 2004). Heterogeneous ensembles are most likely to have more diversity among the models because they are generated from different base classifiers.

In order to generate heterogeneous models, various different learning algorithms are run on the same dataset. Models of this type do not take the same views about the data because the assumptions they make about it are different. A KNN classifier, for example is not as strong against noise as a neural network (Partalas et al., 2009).

Many studies have demonstrated that a heterogeneous ensemble is superior to either a single classifier (Gashler et al., 2008; Smetek and Trawiński, 2011) or a homogeneous ensemble (Dong and Han, 2004; Kang et al., 2015; Borji, 2007; Woloszynski and Kurzynski, 2011; Smetek and Trawiński, 2011). Given that so many high quality studies, using a variety of techniques and types of data, have come to similar conclusions, the superiority of the heterogeneous ensemble is now widely accepted in the field.

2.4.3 Factors Affecting Ensemble Performance

There are several aspects that affect the performance of an ensemble, including the accuracy of individual models, the diversity among the individual models in

an ensemble, the number of models used for constructing an ensemble and the decision fusion function. All these aspects will be briefly discussed in this section.

2.4.3.1 Accuracy of Individual Models

The accuracy of an ensemble is influenced by the accuracy for each individual member in the ensemble. Using members that have an accuracy greater than the random guess is a common suggestion (Wang, 2008; Jurek et al., 2014). The accuracy of a random guess equals $\frac{1}{|L|}$, where $|L|$ is the number of the classes in the dataset.

2.4.3.2 Diversity Among the Individual Models

The importance of the diversity among the individual models is letting the system see the problem from different points of view. If the models used in an ensemble system are identical, then the ensemble's performance will be equal to the individual model. The literature documents many ways in which diversity can be measured, which are classified either as pairwise or non-pairwise diversity measures. Ten approaches are summarized by Kuncheva and Whitaker (2003).

Pairwise diversity measures.

Pairwise diversity measures are used to calculate the diversity between two classifiers (Giacinto and Roli, 2001; Woźniak et al., 2014b). In a training set, an N-dimensional binary vector can be used to represent the output of the classifier. Various methods can be used to measure diversity between classifiers for the purpose of a binary classification problem. A table of the relationship between a pair of classifiers can be produced as in Table 2.1.

One example of this approach is the double fault (DF) measure, which can

Table 2.1: The relationship between a pair of classifiers

		Classifier k	
		Correct (1)	Incorrect (0)
Classifier i	Correct (1)	N^{11}	N^{10}
	Incorrect (0)	N^{01}	N^{00}

be applied "to form a pairwise diversity classifier matrix for a classifier pool and subsequently to select classifiers that are least related" (Kuncheva and Whitaker, 2003). Calculation involves the ratio of wrongly classified examples to the number of examples classified altogether, as illustrated by the equation 2.4.1.

$$DF_{i,k} = \frac{N^{00}}{N^{11} + N^{10} + N^{01} + N^{00}} \tag{2.4.1}$$

Non Pairwise Diversity measures.

In general, it is useful to calculate how one group of models differs from another. One example is Coincident Failure Diversity (CDF) proposed by Partridge and Krzanowski, which is a modification of their Generalized Diversity (GD) measure (Partridge and Krzanowski, 1997).

In proposing GD Partridge and Krzanowski argued that maximum diversity between two classifiers occurs when failure of one of a pair of classifiers is accompanied by correct classification by the other. (Partridge and Krzanowski, 1997) It is defined as follows, where p_m denotes the probability that m randomly chosen classifiers fail on a randomly chosen sample:

$$p(1) = \sum_{m=1}^{M} \frac{m}{M} \times p_m \tag{2.4.2}$$

$$p(2) = \sum_{m=1}^{M} \frac{1}{M} \times \frac{(m-1)}{(M-1)} \times p_m \tag{2.4.3}$$

$$GD = 1 - \frac{p(2)}{p(1)} \tag{2.4.4}$$

The possible values of GD vary between 0 (no diversity) and 1 (maximum diversity).

CFD is defined as follows Partridge and Krzanowski (1997):

$$CFD = \sum_{m=1}^{M} \frac{(M-m)}{(M-1)} \times f_n \tag{2.4.5}$$

Where

$$f_n = \frac{\text{number of samples misclassified by } n \text{ models}}{\text{number of samples misclassified by at least one model}} \tag{2.4.6}$$

The maximum possible value of CFD is 1, the minimum value of 0 is obtained when all models are identical regardless of their accuracy.

CFD was designed to have minimum value of 0 when all classifiers are always correct, or when they are all either simultaneously right or wrong. Its maximum value of 1 is achieved whan all misclassifications are unique, that is, when no sample is misclassified by more than one classifier.

2.4.3.3 Number of models in an ensemble

An influential factor affecting the accuracy of an ensemble is the number of members involved in the ensemble. Thus, increasing the number of classifiers in an

ensemble system leads to an increase in its accuracy (Grove and Schuurmans, 1998; Maimon and Rokach, 2005).

2.4.3.4 Decision fusion function

This is the stage for deciding about assigning an instance to a specific class label (Woźniak et al., 2014b). Majority voting is a useful decision fusion method for ensemble classification (Van Erp et al., 2002). Weighted averages are another kind of method for the decision fusion function, which gives some classifiers higher weight when we believe they are more accurate than others.

2.5 Multi-media Data Mining

Multi-media data mining (MDM) is a kind of data mining which allows useful information to be extracted from such media datasets as video, audio, speech, text, graphics and images. In addition, it can assist in extracting knowledge from combinations of various kinds of datasets, which is an extremely useful application, given the vast amounts of data currently available (Vijayarani and Sakila, 2015; Bhatt and Kankanhalli, 2011). Furthermore, technological developments over recent decades have led to dramatic changes in multi-media functions and activities (Wlodarczak et al., 2015). It is, therefore, currently an important research area. Because MDM deals with two different fields, multimedia and data mining, it is a complex research area (Bhatt and Kankanhalli, 2011).

Multi-media data are usually classified as either unstructured or semi-structured, and multi-media databases are used to store them. A number of multi-media tools and techniques can be applied to discover useful material from within large databases. Similarity searches, entity resolution and the identification of associations are all tasks which multi-media data mining can perform (Manjunath and Balaji, 2014). A further, extremely important, task of multi-media data mining is that of classification, which is the focus of this study. There is a strong need to develop new powerful tools for use with multi-media datasets (Wlodarczak et al., 2015).

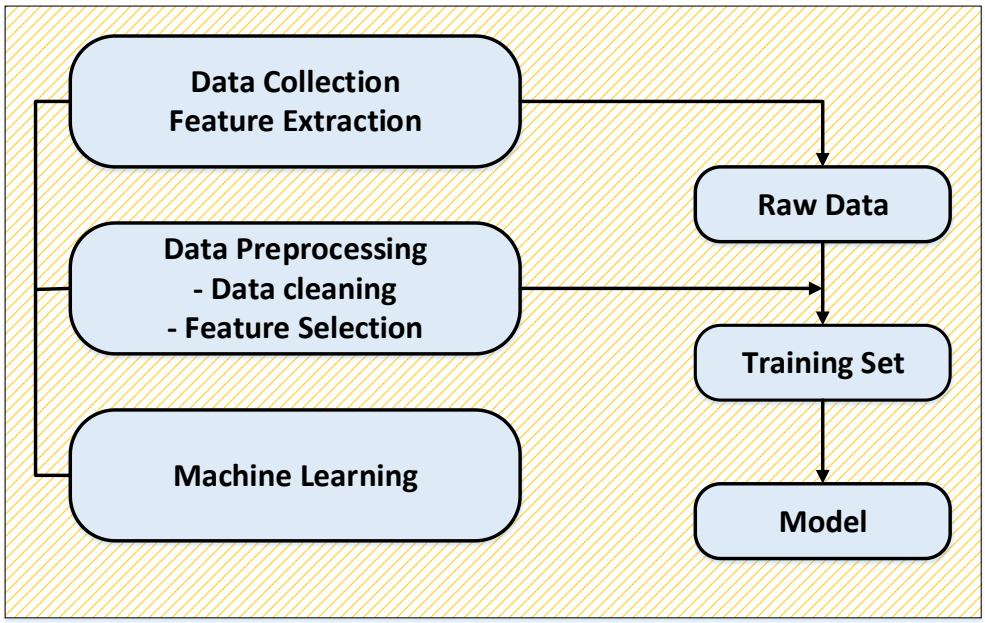

Figure 2.2: Multimedia mining process (Manjunath et al., 2010)

2.5.1 Multi-media Data Mining Process

Multi-media data mining involves a number of steps: data collection, pre-processing and applying machine learning. Figure 2.2 illustrates the stages involved in the process.

2.5.1.1 Data Collection

Multi-media data mining begins with data collection. This stage is of great importance since the results are dependent on the quality of the raw data. This raw data could come from just one kind of media or a combination of types. This research will focus on multi-media data mining from material from a combination of media types.

2.5.1.2 Pre-processing

The aim of the pre-processing stage is to highlight significant features in the raw data. It involves such tasks as cleaning, feature selection and transformation

as necessary. The identification of salient features at this point facilitates the learning. The details of the process must be tailored to suit the problem's domain and the kind of raw data that has been collected. After this step the dataset is split into testing and training sets.

2.5.1.3 Applying Machine Learning

Once the training set has been produced from the pre-processing stage, a learning model must be selected. This is a complicated process because of the vast amounts of data involved and its diversity. In addition, the meaning of the multi-media content is subjective.

2.6 Text Classification

Text classification is the task of categorizing a text document under a predefined category by using the machine learning classifier technique. Formally, if we have d_i as a document on the text dataset D_T and $L = \{l_1, l_2, ..., l_n\}$ is the set of the class labels, then text classification is assigning the document d_i to one category l_j (Ikonomakis et al., 2005).

There are many machine learning classification algorithms useful for text classification including: decision trees, rule based classifiers, SVM classifiers, neural network classifiers, Bayesian classifiers and nearest neighbour classifiers (Prasad and Sebastian, 2014)

2.6.1 Text Classification Process

The process of text classification is illustrated in Figure 2.3. The task for constructing a text classifier is quite similar to any machine learning classification

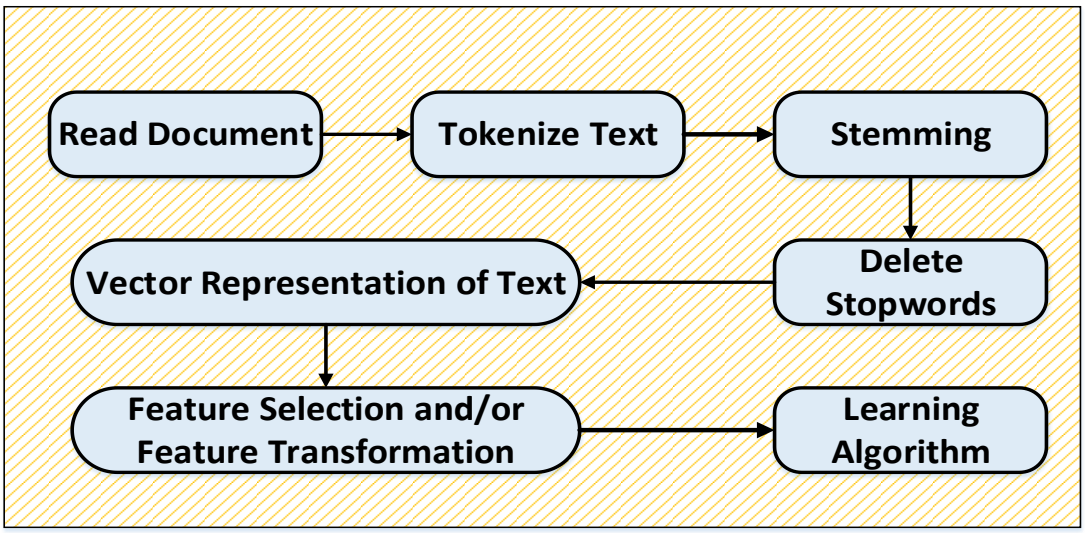

Figure 2.3: Text classification processing (Ikonomakis et al., 2005)

task. The main fundamental issue in text classification is the representation of a document (Leopold and Kindermann, 2002).

Vector space document representation

D_T is a component of documents that contain sequences of words. Machine learning algorithms need to understand the unstructured dataset. Representation methods and techniques must be implemented on D_T to be understandable for machine learning algorithms.

Vector-space models are one of the most familiar structured representations of text. They represent the text as a vector that has its elements indicated by the occurrence of the word in the text (Miner, 2012).

Dimensionality reduction

Vector-space models usually generate models in high-dimensional space, especially, when every single word in the text has been chosen as a dimension in the space. Some words will contribute to the space such as conjunctions, auxiliary verbs, pronouns and articles, and these words are called 'stop words' (Madsen

et al., 2004). Usually, these stop words are useless when classification is applied to D_T. The best representation methods are careful to ignore stop words in the analysis (Ikonomakis et al., 2005; Leopold and Kindermann, 2002).

Feature selection is used in text classification for the purpose of reducing dimensionality (Rogati and Yang, 2002). Text feature selection was categorised by Miner (Miner, 2012) based on three categories: information theory, statistics and frequency. The frequency methods, which are dependent on determining the frequency of the term in the document, are less effective for the task of text classification. The most important approaches regarding these categories include: Information Gain (IG) (Shannon, 2001), Chi-squared (X^2) (Yang and Pedersen, 1997), Correlation-based Feature Selection (CFS) (Witten and Frank, 2005), and Term Frequency Inverse Document Frequency (TF-IDF) (Bramer, 2007).

Information Gain (IG)

Information Gain, introduced by Claude Shannon in 1948 (Shannon, 2001), is categorized as a theory feature selection approach. Miner (2012) state, "IG measures how much the uncertainty about the target variable, called entropy, is reduced when the feature is used". For calculating IG for an attribute x respecting a class y and know the value of x, the value of y measured by its entropy $H(y)$. The uncertainty about y, given the value of x, is calculated using the conditional probability of y given x, $H(y|x)$:

$$I(y, x) = H(y) - H(y|x) \tag{2.6.1}$$

Where y is a discrete variable that takes a value in $\{y1.....y_c\}$ and x is a discrete variable that takes a value in and $\{x1....x_d\}$, the entropy of y is calculated using

the following equation:

$$H(y) = -\sum_{i=1}^{c} P(y = y_i) log_2 P(y = y_i) \tag{2.6.2}$$

The conditional entropy of y given x is:

$$H(y|x) = -\sum_{j=1}^{d} P(x = x_j) H(y|x = x_j) \tag{2.6.3}$$

Applying machine learning algorithms

This is the stage where we can apply machine learning classification algorithms. There are several machine learning classification algorithms useful for text classification tasks, including support vector machine, Naïve Bayes, decision trees and k-nearest neighbours to name a few. In this research, a heterogeneous ensemble machine learning classifier will be implemented to solve this issue.

2.7 Image Classification

Image classification is a form of image mining. Image mining as a whole, is defined as extracting useful patterns from a sizeable group of images. Image classification involves a process by which the class label for a new image can be predicted. This occurs by learning from image datasets which have already been labelled. Feeding image datasets into a machine learning algorithm is not possible until the features from this dataset have been extracted by using a suitable image-feature extraction method (Lu and Weng, 2007).

2.7.1 The process of Images Classification

The process of classifying image datasets involves two main steps. The first step is the data representation. The second step is the application of machine learning

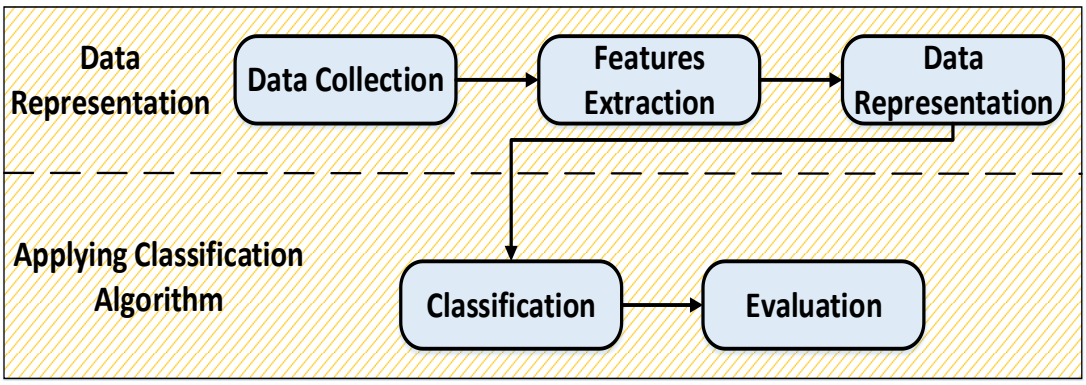

Figure 2.4: Image classification processing (Kamavisdar et al., 2013)

classification algorithms to the represented data. Figure 2.4 shows the steps of the image classification process. It starts with the collection of labelled image data from which appropriate features are extracted using suitable tools. Once this has been achieved, the extracted features can be presented as a dataset which can then be processed by machine learning algorithms. Once this dataset has been prepared, it is then possible to apply normal classification methods to it.

2.7.2 Image Feature Extraction

Feature extraction is the representation of a dataset which contains many images. It is a kind of efficient dimensionality reduction which allows the interesting aspects of an image to be represented as a compact feature vector. It is thus a method of constructing combinations of variables, which can be used to solve the problems surrounding data mining and analysis while maintaining adequate levels of accuracy when describing data. Common techniques used for feature extraction include, Histogram of Oriented Gradients (HOG), Speeded up robust features (SURF), Colour Histograms, Haar Wavelets and Local Binary Patterns (LBP).

2.7.2.1 HOG

The HOG feature descriptor can be defined as an image or image patch created by means of the simplification of an image through the extraction of useful information and the discarding of superfluous information. The image can then be represented a vector of features (Dalal and Triggs, 2005).

An example of a HOG descriptor being extracted from a 64x64 image was given by Xiao et al. (2010). In this example, a cell is made of 8x8 pixels. Each block is made up of 2x2 cells and there are nine bins (K). The total number of cells is therefore 64 (8x8) and there are 49 (7x7) blocks. This means that the HOG feature has an overall dimension of 1746 (9x2x2x49).

There are a number of studies which have been conducted using HOG features for classification (e.g. (Abidin et al., 2018; Xiao et al., 2010)).

The first study combined HOG with Local Binary Pattern (LBP) feature extraction method for analyzing image textures due to its robustness to pixel variation and computational simplicity. The 1536 dimension HOG features were extracted from the dataset. This method outperformed the traditional LBP method.

The researchers in the second study used HOG to extract features for the classification of the leaves of a plant. They combined the HOG feature extraction with the Maximum Margin Criterion (MMC) proposed by Li et al. (2004) (dimensionality reduction method). The results they obtained were of a high quality, when comparing their results to others in the literature.

2.8 Combined Multimedia Data Mining

Despite a thorough search of the literature, it was difficult to find studies which used combined multi-media data as defined in Chapter 1. There are a number of studies which define multi-media data somewhat differently so consider image data to be a kind of multi-media data. However, according to the definition we present of multi-media data, these studies are only dealing with one type of data. Nevertheless, examples of such studies are discussed below.

For example Wlodarczak et al. (2015), used a machine learning approach for multimedia data mining. However, they used each type of media individually. Furthermore, in their studies, they used different experiments for different data. They dealt with one type of media in some experiments but other types of media were considered separately, in other experiments. Other studies (Naaman, 2012; Yan et al., 2015) used image data as multi-media data, and others used video data as multi-media data (Oh and Bandi, 2002; Chen et al., 2001). However, none of these studies combined different forms of data (text, image, video) in one experiment.

There is however, one example of a study presented by Mojahed et al. (2015); Mojahed and de la Iglesia (2017) which uses heterogeneous data and defines it in a way which is very similar to the definition used in the present study. This study, though, differs from ours because the method used was for clustering and the datasets collected were to examine clustering problems. The limitation of the study was that the researchers could not find the required data so created it themselves, and the datasets used were very small.

2.9 Information Fusion

Atrey et al. (2010) reviewed multi-modal fusion for multimedia analysis. In their work, they divide the level of fusion to three levels (feature, decision and hybrid). They show some advantages and disadvantages to these levels. The benefits of feature level fusion are that it is less complicated, and it requires only one learning stage. Also, it can utilise the correlations between features that have been combined from different data sources. On the other hand, the disadvantages of fusion at the feature level arethat adding more features from different modalities increases the difficulty of learning the cross-correlations among the heterogeneous features. Furthermore, fusing features from different datasets results in high dimensionality of the combined data which then needs to be reduced using feature reduction techniques. The advantages of decision level fusion are that it gives the flexibility for using suitable methods to analyse each modality. Moreover, it enables a decision to be obtained from each data representation and then to fuse these decisions. However, the disadvantage of decision level fusion is that it cannot utilise the correlation between features that come from multi-modalities.

Imani and Ghassemian (2020) reviewed spectral and spatial information fusion for hyperspectral image classification. In the review, they discuss information fusion by three different methods: segmentation-based methods, feature fusion methods and decision fusion methods. They mention that fusing spatial and spectral features at the feature level can be done by two different approaches. The first approach involves extracting the spatial features and the spectral features separately, then combining them using a combination method. The second

approach involves extracting the spectral-spatial features together to preserve the correlated nature where the spectral and spatial information is dependently and jointly contained. In decision level fusion, a number of sub-features can be extracted using a number of feature extraction methods; then learning models can learn from each sub-featuri and gave a local decision; finally, the global decision can be obtained by fusing all the local decisions.

They note some advantages for feature level fusion and decision level fusion. The advantages for feature level are simple implementation, and that it is effective if an appropriate feature extracting method is used. The advantage of the decision level is that it gives the ability to combine several robust classifiers. They state that the recent research for fusion spectral-spatial features methods has been in three main areas: design of new feature extraction methods, hybrid fusion methods where two or more types of fusion methods are used for feature fusion, and deep learning methods for joint spectral-spatial feature generation with the extraction of detailed features.

In (Garcia-Ceja et al., 2018), the authors studied activity recognition by fusing two datasets extracted from two different sensors: accelerometer and sound. They reviewed nine research projects similar to their own, and they distinguished their work by combining the data from various types of sensors to increase the performance of their methods. In their experiments, they used RF as the base learning algorithm, which outperforms NL and SVM. The accelerometer data were collected by wrist-band and the sound data collected by cell-phone. In their experiments, 10-fold cross-validation was used to evaluate their methods, and the

accuracy results (%) were: Audio view 83.8, Accelerometer view 85.4, Aggregated views 92.1 and Multi-View Stacking 94.1. Their work shows that the ability to combine different types of data coming from different sources could improve the performance of the analysis methods.

In health care, van Loon et al. (2020) used information fusion to predict heart disease by fusing two different data sources: electronic medical records (EMRs) and sensors. The classification prediction was obtained by ensemble deep learning algorithms. Their results showed a clear improvement of accuracies obtained from the fused data compared with accuracies obtained from each data source individually. Moreover, deep learning performed better when it was used with the fused data, but Naïve Bayes performed better when it used each data source individually.

From these publications, we can see that information fusion is a promising research area in data analysis, and that it has been shown to give benefits to data analysis by improving the accuracy of classification methods.

2.10 Related Work

In this section we will present some work which is related directly to this thesis. This section will consist of sub-sections including ensemble in the context of data fusion and images feature extraction and classification.

2.10.1 Scene Classification

Many scene classification studies have been conducted previously. A notable study was done by Oliva and Torralba (2001) using a dataset called 8 Scene

Categories Dataset. Their experiment involved classifying images and their annotations into eight categories using the support vector machine technique, by training 100 instances from each class and testing the rest. They achieved 83.70% accuracy.

Bosch et al. (2006) also studied scene classification. They started the study by recognizing all possible objects in the image, and then classifying each image regrading to its objects. They used pLSA (Hofmann, 2001) to represent objects in the images. The pLSA originally devolved as topic discovery in a text but it was used in this research because images were represented as frequency of visual words. The k-Nearest Neighbour (k-NN) algorithm was used as a classification method in three different datasets.

Yang et al. (2007) conducted an experiment on scene classification using keypoint as a method to extract features from images. In their experiment, images were described as a bag of visual words. They demonstrated that their methods outperform others using two benchmark datasets: TRECVID 2005 corpus and PASCAL 2005 corpus. The keypoint approach was originally created to classify text datasets, and was found to be useful for image classification as conducted in this experiment and others, including in (Lowe, 2004; Ke and Sukthankar, 2004; Mikolajczyk and Schmid, 2004).

Scene classification has been studied from the view of homogeneous ensemble methods. Yan et al. (2003) applied an homogeneous ensemble of SVM models to classify rare classes on scene classification. Their experiment was conducted on a dataset called TREC 02 Video Track, and was compared with other approaches

applied to the same dataset. The results obtained in the experiment outperformed other methods with 11% improvement in the best case.

Medjahed (2015) conducted a number of experiments by applying four different classifiers to the public image dataset "Caltech 101". The 4 different classifiers which were used were: Linear SVM, SVM with Gaussian kernel, Least Square SVM (LS-SVM) and k-nearest neighbour. The researcher used 14 Feature extraction methods. PHOG was one of these methods and was found to exceed all the feature extraction methods in the multi class classification.

Seeger et al. (2016) conducted their study on road type classification with occupancy grids. The image dataset used for the experiment contained about 700 local occupancy grids per class for training and 150 for testing. Both SVM and CNN were applied to classify the data. The feature extraction methods used for the purpose of training SVM were PCA, CENTRIST, Gist and PHOG. For training CNN the different network topologies used were AlexNet, GoogLeNet, VGG16. The results showed that combining PHOG with Gist produces good results which are comparable to the best results achieved using CNN. They both produce 94% accuracy.

2.10.2 Ensemble methods in classification in information fusion context

2.10.2.1 Heterogeneous Ensemble Methods for Classifying Multi-media Data

Lertampaiporn et al. (2013) applied a heterogeneous ensemble for classifying pre-miRNA by using voting for a set of classifiers including a support vector machine, k-NN and random forest.

Giacinto and Roli (2001) enforced neural network ensemble for image classification on a dataset of multi-sensor remote-sensing images. They focused on classifying a bunch of pixels related to different images for different classes. The experimental results they obtained demonstrated the effectiveness of homogeneous neural network ensemble, with the level of accuracy achieved in the experiment being higher than the best accuracy of individual neural network models.

Mojahed et al. (2015) applied the machine learning clustering method to heterogeneous (though not necessarily multimedia) datasets. Due to the fact that there were not many heterogeneous datasets publicly available, they created their own heterogeneous datasets, which contained different types of media. Their combined data achieved a significant advantage on clustering performance over that of using only one type of data.

Tuarob et al. (2014) applied the machine learning heterogeneous ensemble approach to classify social media datasets. They conducted their experiments using three datasets, two collected from Twitter and one from Facebook. They used five different feature extraction methods to generate the data needed for machine learning algorithms. Each of them created a subset of all the combined data. Five base classifiers were used in their experiments, and the classifiers' results were combined using different methods, including majority voting and weighted voting. They suggested that the additional features may increase the accuracy of classifiers. However, strictly speaking, in this study, the datasets are not of multimedia, but a single media of multiple textual datasets.

Ballard and Wang (2016) developed a dynamic ensemble selection methods

for heterogeneous data mining. Although their datasets are not multimedia, their basic idea of combining multiple datasets at decision level inspired this work.

Ensemble voting schemes

The Majority Voting scheme is a popular method when constructing heterogeneous ensembles. For example, Aburomman and Reaz (2017) conducted a survey study of intrusion detection systems which are based on ensembles. They reviewed nine different heterogeneous ensemble methods. Seven of these used the Majority Voting scheme for their ensembles, which demonstrates their popularity.

2.10.2.2 Classification and Ensemble Methods in Feature Level

Mehmood and Rasheed (2015) classified microbial habitat preferences, based on codon/bi-codon usage. They obtained a high dimensional dataset by combining different datasets from different data sources. They showed that the combination, on the feature level, leads to a high dimensional dataset. Thus, they focused on feature selection to reduce the dimensionality of the combined dataset. They reduced a huge number of variables with acceptable classification accuracy.

Chen et al. (2015) also conducted an experiment on combining heterogeneous datasets to a single dataset, and applied homogeneous ensemble classification methods to it. They used a support vector machine as the base classifier. In addition, they used real-word microblog datasets, provided by Tencent Weibo. Their results show that the aggregated dataset outperforms any single dataset. Nevertheless, the datasets they used are not of multimedia data. Hence, the level of effectiveness of these ensemble methods on multimedia data is unknown.

Liu et al. (2017) used both image and audio data to detect drones. They used the feature level integration approach to generate a big dataset for training SVM models for classifying drones. The images' features were extracted using HOG.

Zhang et al. (2019) conducted their experiments using an images dataset for Chinese hand writing. They extracted the features from the dataset using four different methods (PCA, HOG, PHOG and GIST). SVM was the only base learning algorithm used. The results obtained shows that of the four features used, the best single feature extraction method was GIST at 87.89% accuracy. However, combining PHOG + GIST produced 92.33% accuracy. But when combining PHOG+GIST+PCA, the accuracy level dropped to 92.22%. This shows that simply adding more feature extraction methods does not necessarily produce better results in terms of accuracy. Thus, care must be taken when combining different data at the feature level.

Koh et al. (2018) examined the use of automated detection in the field of retinal health. They investigated the use of PHOG and SURF for features extracted from images of the fundus of the eye. The fundus images were obtained from the Ophthalmology Department of Kasturba Medical College (KMC) in Manipal, India. The researchers extracted a total of 111,272 features (680 PHOG and 110,592 SURF) in this work. The classifier they used was the k-nearest neighbour (kNN) and the features were combined using Canonical Correlation Analysis (CCA). Using this method they obtained 505 features. They demonstrated that the best results were produced by combining SURF and PHOG.

Another study investigating fundus images was conducted by Gour and Khanna

(2019), specifically of the detection of glaucoma, using two different image datasets (Drishti-GS1 and HRF) The classification method used in this study was SVM and the feature extraction methods used were PHOG and GIST. The results show that their method outperformed other methods including methods that used CNN for features extraction for example (Orlando et al., 2017). Using Drishti-GS1 dataset,Gour and Khanna (2019) got 0.86 AUC and in HRF they got 0.88 AUC. In comparison, Orlando et al, using the Drishti-GS1 dataset, got only 0.76 AUC and in HRF they got 0.78 AUC.

Bai et al. (2009) investigated a feature extraction method for smile recognition using PHOG. The Cohn-Kanade AU-Coded Facial Expression Database was used to train and test the smile recognition system. The feature extraction methods used were Gabor and PHOG. The base learning algorithms used for classification were SVM and AdaBoost. The results shows that combining PHOG with Gabor features was a good approach for smile recognition and outperforms using just a single feature extraction method.

2.10.2.3 Classification and Ensemble Methods in Decision Level

Bagnall et al. (2012, 2015) applied ensemble methods to time series data analysis . In their work, massive time series datasets were transformed into four different representations, which were equivalent to multimedia datasets, and these were used to train seven different base classifiers including Random Forest, Naive Bayes, Decision Tree and Support Vector Machine. Some of these classifiers were then used to build ensembles. They demonstrated that they could achieve significantly improved accuracy on more than 75 datasets. Do et al. (2017) conducted

experiments in the same area using series Nearest Neighbours classification. Their methods out-performed other methods, including Random-Forest and Support Vector Machine.

Yamanishi (1999) conducted a study of the distributed learning system for Bayesian learning strategies. In their system each instance was observed by different classifiers which were called agents. They aggregated the outputs from the agents to give significantly better results. They demonstrated that distributed learning systems work approximately (or sometimes exactly) as well as the non-distributed Bayesian learning strategy. Thus, by employing their method, they were able to achieve a significant speeding-up of learning.

Onan (2018) applied ensemble classification methods to text datasets. In his experiment the data sets ware represented by 5 different formats. Five types of classifiers were used: Naive Bayes, Support Vector Machine, K-Nearest Neighbour, Logistic Regression and Random forest. He compared individual classifiers and their homogeneous ensemble using Bagging and Boosting. The results showed that ensembles out-performed individuals.

Tah (2016) investigated an ensemble learning method for scene classification. They based their work on the Hidden Markov Model for image representation. They used only one learning algorithm (SVM) with 4 different representations: SIFT, Gist, Centrist and Gabor. The data set was an images dataset comprised of 15 natural scenes. Their experimental results showed that the classification accuracy obtained by combining classifiers is superior to using each classifier separately.

Audebert et al. (2017) presented a novel method to perform fusion of heterogeneous data using fully convolutional networks for urban semantic labelling. They introduced residual correction as a means of learning how to fuse predictions merging from a double stream architecture. They carried out fusion of DSM and IRRG optical data, for an urban area, using the ISPRS Vaihingen dataset and obtained new up to date results. A naive approach to data fusion was taken, and used deep networks. The researchers used both the decision level and the feature level. The results showed the feature level to be better.

Oramas et al. (2018) studied multi modal deep learning for classifying music genre. The datasets used were multi media datasets which were represented by more than one type of media. Two experiments were conducted. One used the multimedia data set MSD-1, which is a combination of images and audio. The other experiment used MUMU datasets, which were a combination of audio, image and text. They used CNN to extract features from each type of media. Their results were presented by using a single media or combined media datasets in both decision and feature levels. Feature level combination outperformed the overall results.

Chapter 3

Methodology

3.1 Introduction

In this chapter, we describe the research methodology, methods and tools that will be used to conduct this research. In addition, we describe the datasets and the measures that will be used to evaluate our proposed methods.

The structure of this chapter as follows: Section 2 will detail the research design and methods. Section 3 will provide details of the evaluation methods. Section 4 will show the details of the datasets that used in our research. Section 5 will present a summary of the chapter.

3.2 Research Design and Methods

The proposed framework in this research will be comprised of two fundamental phases as illustrated in Figure 3.1, and each phase will contain several stages. The first phase is mainly data representation and fusion, that is, transforming several multi-media datasets into a single dataset or several sub-datasets. The second phase is to build a heterogeneous ensemble for the transformed dataset.

It should be noted that in this research, the multi-media data include numerical, textual and image data. The audio, video and time series data are not

considered, although they can be used once they are convert into features.

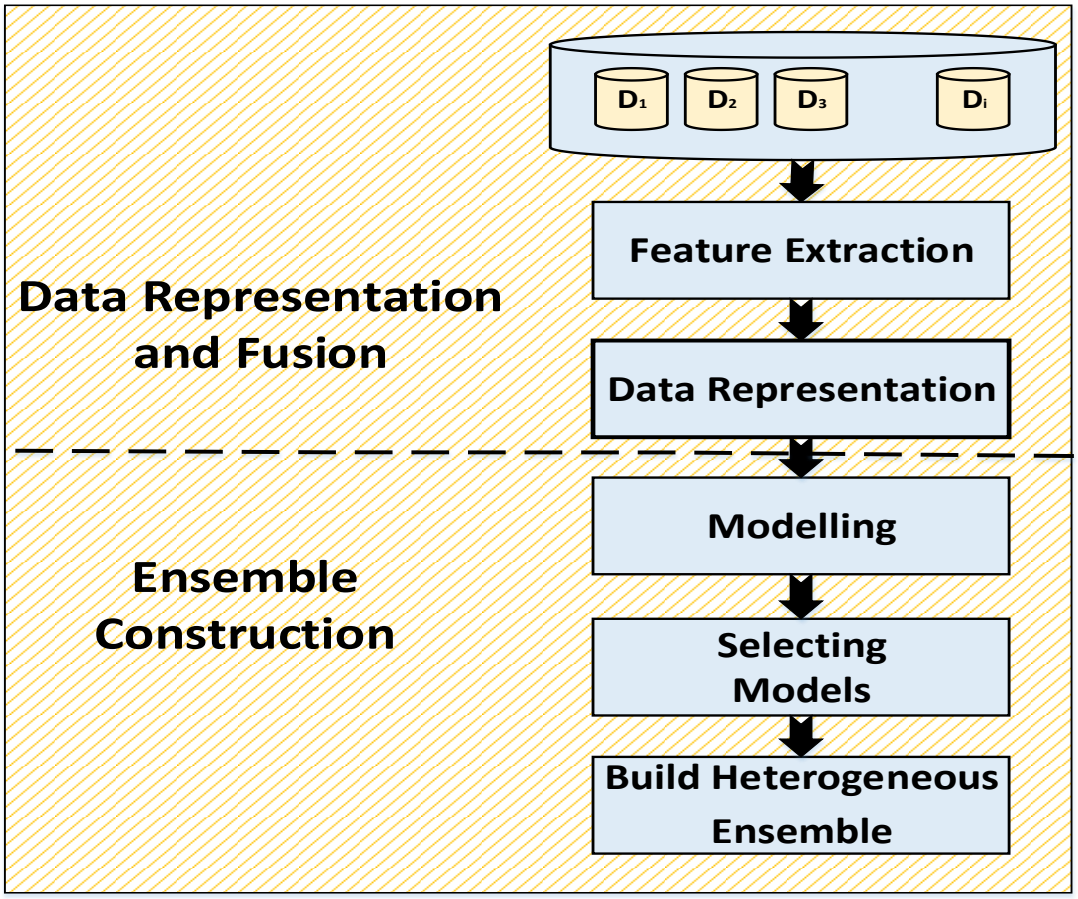

Figure 3.1: The proposed conceptual framework

The first phase of the framework will begin with a multi-media dataset **MMD** $= \{D_1, D_2, ..., D_i\}$, which may include several types of media, for example text and images. Certain features will then be extracted from each type of media using suitable feature extraction techniques. These extracted features will then be stored in sub-datasets depending on the type of media, or aggregated into a single dataset. The decision of whether to store features in sub-datasets or aggregate them together will be determined by ensemble construction strategies, as shown in the next phase. Once this has been carried out, the first phase of the methodology will be complete.

The second phase will involve building a heterogeneous ensemble for the dataset composed of the features which were extracted and aggregated during phase one.

All stages of the conceptual framework will be described in detail below.

3.2.1 Data Representation and Fusion

This first phase will transform MMD into numeric datasets that will allow machine learning algorithms to analyse them. This phase will include two stages: feature extraction and data representation. These stages will be described below.

3.2.1.1 Feature Extraction

In this stage of the conceptual framework, features will be extracted from a MMD. For this research, the extraction of features from text and image datasets will be done with suitable feature extraction techniques. These techniques will represent every instance in a MMD using vector values. The numeric dataset will not need to go through this phase. Therefore, at the end of this phase we will be able to evaluate all three types of media in one dataset.

Extracting Features from Text Dataset

The features from the unstructured text dataset will be extracted using the String to Word Vector filter. This filter transforms unstructured text data into numeric attributes by marking each word in the data as a feature and treating each text document as a potential instance. Then, if the word appears in a given document, it takes the value of 1 in the dataset; if it is absent from the document, it takes the value of 0. The structured features from the text dataset, which are made

up of XML files, will follow the same procedure as that of the unstructured text dataset, replacing the vector words according to the tags contained in the dataset.

Extracting Features from Image Dataset

There are several techniques which can be used to extract features from images for the purpose of classification, including a histogram of oriented gradients (HOG) and a generalised search tree (GiST). In this research, image features will be extracted using a HOG because it has the capability to transform an image to a matrix. To create a HOG, the image is partitioned into a number of blocks, each having the same number of cells, and each cell contains the same number of pixels. Blocks are then used to identify the image features that we will examine in next stage.

It should be pointed out that, since this research is about developing heterogeneous ensembles for multi-media data, it will not investigate the pros and cons of available methods for feature extraction. Rather, commonly available techniques have been selected to allow us to move directly to data collection and analysis.

3.2.1.2 Data Representation

The data generated during feature extraction will be organised into a suitable representation to facilitate the ensemble construction phase of the project. There are two different ways to do it. The first is to aggregate all the features into one dataset; the second is to store extracted features in multiple datasets, one for the feature-set extracted from each type of media. In each case, the dataset creation method will be determined by the ensemble to be constructed subsequently.

3.2.2 Ensemble Construction

Ensemble construction is the second phase of the proposed framework for this research. In this phase, machine learning ensemble methods will be applied, to learn and classify the datasets created in the previous phase.

As described earlier, two different representations of multi-media data can be produced, a single aggregated dataset or in a number of sub-datasets, each made up of data from different media. Having these two kinds of data representations will permit us to construct ensemble machine learning methods at two different levels. The first ensemble level can be built at the feature level, hence it is called Feature Level Ensemble (FLE), while the second will be at the decision level, and is called Decision Level Ensemble (DLE). The main issue in this research is to investigate the differences of the construction and performance of these two ensemble methods for MMDs. The following sections will describe these two types of ensemble, as well as the stages necessary to implement these ensembles in this research.

3.2.2.1 Feature Level Ensemble

The FLE (as illustrated in Figure 3.2) will start by extracting features from an MMD. After that is complete, all features will be aggregated to one big dataset. Then all individual models $M = \{m_1, m_2, ..., m_i\}$ will be built from this dataset using all available base classifiers $C = \{c_1, c_2, ..., c_i\}$ and added to the pool of models (PM). Next, model section criteria will be applied to the PM. Finally, the results of the selected models will be aggregated using a combination method to produce the final results of the ensemble.

The FLE will contain an equal number of models M and base classifiers C because it will be built on one dataset. This will give less variety in the models, but the models will be built on a single dataset with a large amount of variety in its attributes. This part of our research will investigate how this will affect the performance of the ensemble.

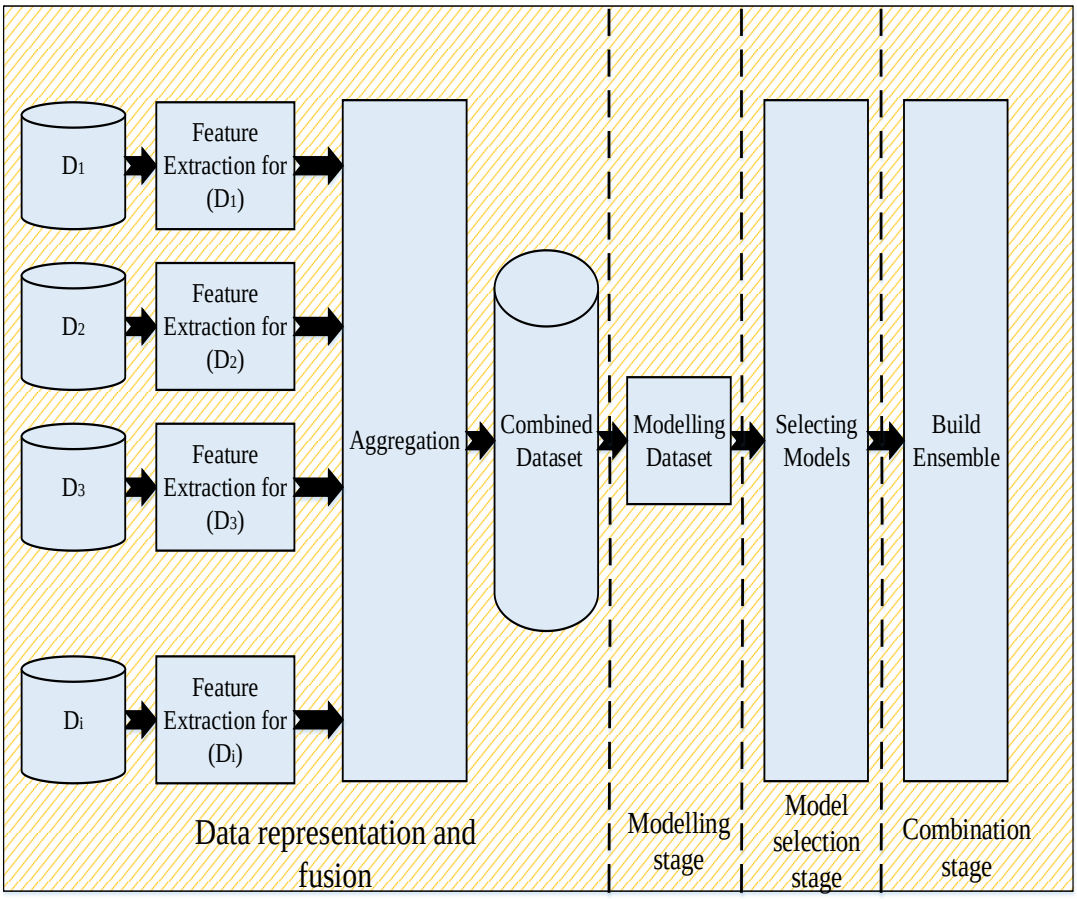

Figure 3.2: Ensemble at the feature level combination

3.2.2.2 Decision Level Ensemble

The DLE (as illustrated in Figure 3.3) will also begin with the extraction of features from MMDs. Features will be extracted from each type of media into a separate sub-dataset, then the modelling stage will be implemented immediately for each sub-dataset. This procedure will lead to the models being added to the

PM. After that, model selection will be conducted in the PM to determine which models will be utilised. The final stage of construction for this ensemble will be to combine the results of the selected models using a model combination method to obtain the final results.

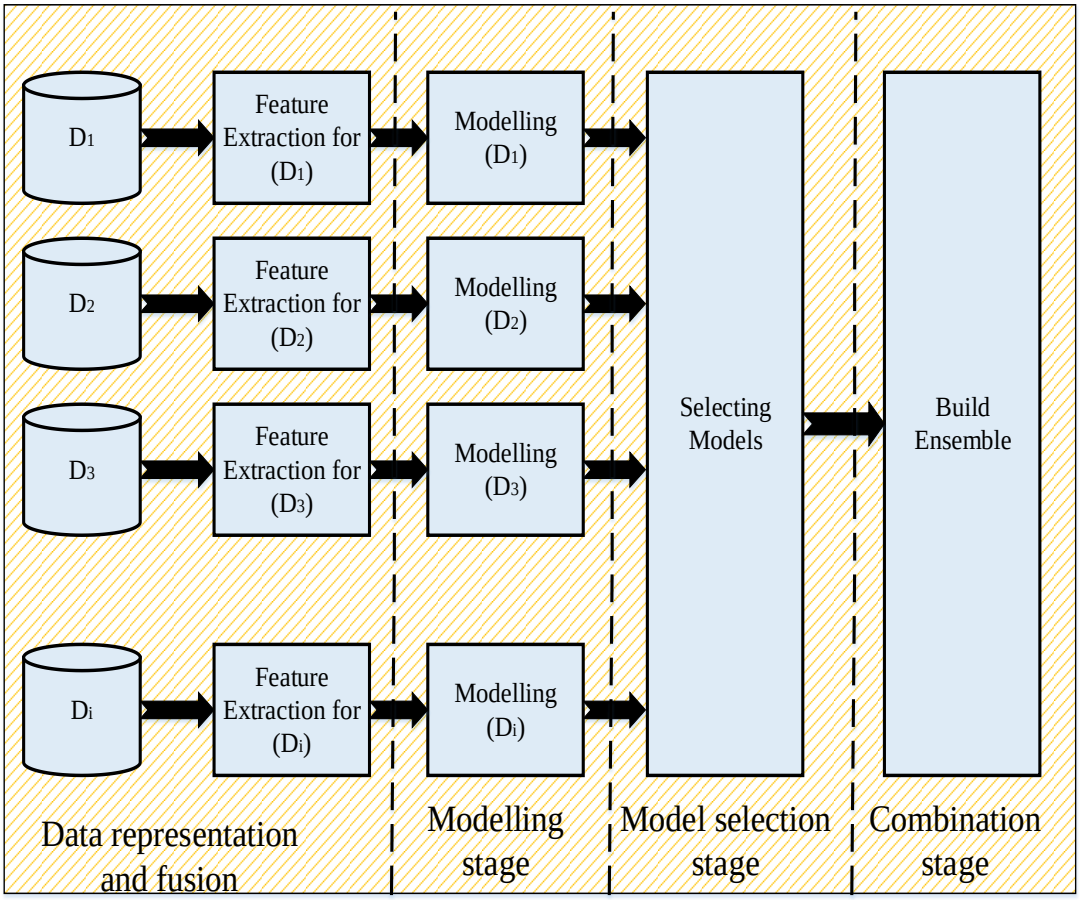

Figure 3.3: Ensemble at the decision level combination

In this type of ensemble the number of models $|M|$ is dependent on the number of base classifiers C and the number of types of media in the MMD, $|D_i|$, as shown in Equation 3.2.1.

$$|M| = |C| \times |D_i| \tag{3.2.1}$$

This will result in more variety in the models that will be built from the

multiple sub-datasets than those from the single dataset used at the FLE, and could have an effect on the results, therefore this is another issue that will be investigated in this research.

3.2.2.3 Ensemble Stages

The process of building ensembles in this research will consist of three different stages: modelling, model selection and combination of selected models.

Modelling

In this stage, the base classifiers $C = \{c_1, c_2, ..., c_i\}$ will learn from the training datasets (Tr) to produce models $M = \{m_1, m_2, ..., m_i\}$. All models in this stage will be created from the datasets described in the previous phase, and then stored in the PM. There are several base classifiers that could be used in this research. These include decision trees, Naive Bayes, support vector machine, artificial neural networks and others. The most important considerations in the modelling stage will be (1) determining which base classifiers are suitable, and (2) determining the necessary number of models to be used.

In this research we have selected 10 different base classifiers that are provided in the WEKA library. These base classifiers are: (*trees J48, RandomTree, REPTree*), bayes (*NaiveBayes, BayesNet*), function (*SMO*), rules (*JRip, PART*) and Lazy (*IBk, LWL*). The purpose for using these five different categories for the base learning algorithms is to maximise heterogeneity. Since our research focuses on the ensemble, these learning algorithms are used as black-boxes. Therefore, we utilised the default values of their parameters as set by WEKA. There are

many classification studies which have used the WEKA default parameters including, in images classification (Xue et al., 2015), and in heterogeneous ensemble classification (Seijo-Pardo et al., 2017; Haque et al., 2016; Haq and Wilk, 2017).

Model Selection

Next, we will carry out the task of selecting models from the many candidates stored in the PM. During this stage, three factors can affect the selection of models, and the performance of an ensemble and must be taken into consideration: (1) the accuracy of the models, (2) the diversity of the models, and (3) the number of models to be combined. In this research diversity will be measured using two different measures: Double Fault (DF) diversity and Coincident Failure Diversity (CFD), which were described in Chapter 2.

It is important to note that this research will use two mechanisms of model selection to determine if an individual model should be added to the ensemble. The first mechanism applies a single measurement criterion to select a single model, for example using just an accuracy measurement or just a diversity measurement. The second mechanism uses more than one measurement to select a single model, and gives a weight to each measurement. For example, using both an accuracy measure and a diversity measure to select a single model, and giving equal weight to both.

Combination of Selected Models

This third stage is where the selected models will be combined. The combination method used in this research is Majority Voting (MV). An example of MV scheme: assuming we have 3 models that classified an instance to be *yes* class or *no* class.

And the result obtained results was two of them classified the instance as *yes* so that the final ensemble decision will be *yes*.

3.3 Evaluation

This section describes the existing methods and measures that will be used to carry out the experiments and to evaluate the results.

Since we could not find similar methods in the literature that build heterogeneous ensembles for classifying multi-media data to compare with our methods, we will use established homogeneous ensemble methods like Random Forest and AdaBoostM1 for the purpose of comparison. Our methods will be compared with these methods statistically to compare their performance.

3.3.1 Data Partition Strategies for Training and Testing

For the purpose of building and evaluating the classifier, the dataset needs to be divided into separate subsets for training, validation and testing. This research will employ some common techniques detailed below.

3.3.1.1 Percentage Split

This validation technique is straightforward because it splits a dataset D into two parts. The first part is used for building the intended classifiers, and the second part from the data is used to test them. The percentage of the split may 50:50 , 60:40 , 70:30 80:20, etc., depending on the size of a dataset and its quality.

Data Randomisation and Selection in Our Experiments

For this research, it was necessary to divide the initial dataset into four folds of 25% of the data each. Three folds (75%) were used for training and one

(25%) for testing. Of the training folds, 25% of the data was taken for validation and 75% for training, as shown in Figure 3.4.To achieve this, the WEKA filter, *StratifiedRemoveFolds*, was applied. This filter allows the data to be split into a number of folds each of which contains the same percentage of classes because this particular filter takes into consideration class distribution (Pooja, 2013). The package is used in the WEKA GUI environment under *weka.filters.supervised.instance.StratifiedRemoveFolds* (Bouckaert et al., 2016). The parameter seed is a random number seed for shuffling the dataset. In our experiments we benefit from this parameter when we change the number for each different run on the same dataset.

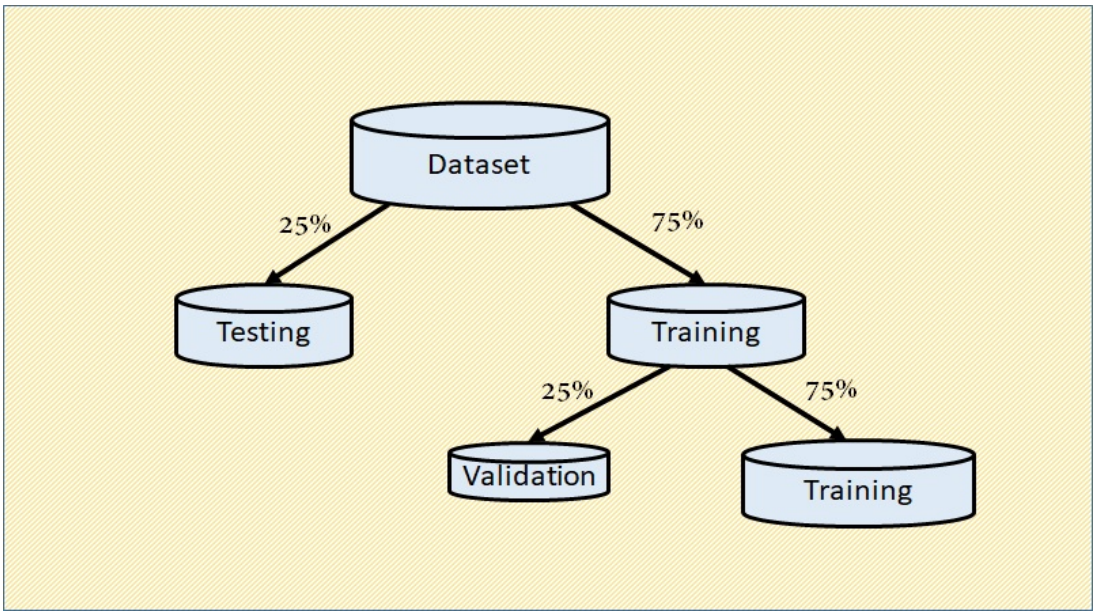

Figure 3.4: The procedure of partitioning the dataset

3.3.2 Measures of Accuracies

3.3.2.1 Binary Class Performance Measures

Binary Confusion Matrix

The binary confusion matrix is a table that contains the values of the actual and

predicted results for an intended classifier. An example of a binary class confusion matrix is shown in Table 3.1.

Table 3.1: An example of a binary class confusion matrix, ***TP*** is the number of correct predictions for positive cases, ***FP*** is the number of incorrect predictions for positive cases, ***FN*** is the number of incorrect prediction for negative cases, ***TN*** is the number of correct predictions for negative cases.

		Actual	
		Positive	Negative
Predicted	Positive	***TP***	***FP***
	Negative	***FN***	***TN***

Binary Class Performance Measures

Many performance measures can be calculated with the use of the binary class confusion matrix Table 3.1. The most important measure is accuracy (Acc). It can be calculated with Equation 3.3.1.

$$Acc = \frac{TP + TN}{TP + FP + FN + TN} \tag{3.3.1}$$

3.3.2.2 Multi Class Performance Measures

Multi Class Confusion Matrices

The multi-class confusion matrix is more complex than the binary confusion matrix. Table 3.2 shows a confusion matrix for four different classes. The true positive TP values in this table are the intersection of each actual class and its predicted class.

Table 3.2: A confusion matrix for 4 classes.

		Actual			
		Class A	Class B	Class C	Class D
Predicted	Class A	TP_{AA}	e_{AB}	e_{AC}	e_{AD}
	Class B	e_{BA}	TP_{BB}	e_{BC}	e_{BD}
	Class C	e_{CA}	e_{CB}	TP_{CC}	e_{CD}
	Class D	e_{DA}	e_{DB}	e_{DC}	TP_{DD}

Multi Class Performance Measures

The classification accuracy of a multi-class classifier is the ratio of the sum of the principal diagonal values to the total of the values in the confusion matrix. To calculate the Acc for a multi-class classifier, we use Equation 3.3.2.

$$Acc = \left(\frac{\sum_{i=1}^{N} C_{ii}}{\sum_{i=1}^{N} \sum_{j=1}^{N} C_{ij}} \right) \tag{3.3.2}$$

where

$\boldsymbol{N}$ is the number of classes

$\boldsymbol{i}$ is the row index for the confusion matrix

$\boldsymbol{j}$ is the column index for the confusion matrix

In case of multiple classes the sensitivity can be calculated for only one chosen class as the target.

3.3.3 Statistical Tests for Comparison

Statistical tests are used to compare the performance of a classifier with that of another classifier. Several tests can be used, such as the Friedman test and the paired t-test, which are described below

3.3.3.1 Paired t-test

The paired t-test is a statistical test used to determine the difference in performance of two classifiers over different datasets. This test identifies whether a difference in the performance of classifiers over datasets is significantly different from zero.

3.3.3.2 Friedman Test

The Friedman test is a non-parametric test used to rank the performance of classifiers for each dataset. It ranks performance in a descending way, in which the best performance is ranked as 1, the second best as 2 and so on. Average ranks will be assigned whenever more than two classifiers are equal. Then, if the null hypothesis is rejected, a graphical critical difference diagram will be used.

3.3.3.3 F1-Score

In Chapter 7, we compared our results with the results obtained by Oramas et al. (2018). In their experiments, they used the macro-average F1-Score to evaluate their methods. For this reason, we used the same measurement. The macro-average F1-Score for multiple classes is calculated by computing the F1-score independently for each class as shown in equation 3.3.3, then the average of these scores is calculated. Hence, all classes are treated equally

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3.3.3}$$

3.3.3.4 Critical Difference Diagram

To make statistical comparisons on our results we used critical difference diagrams (CDDs) which were introduced by (Demšar, 2006). We used them as they used in (Bagnall et al., 2012) and the full description was presented in Jason Lines thesis (Lines, 2015). Because we only used one dataset in our experiments, the test was performed over five different runs. The comparison is performed in two stages as described below.

Stage one. The null hypothesis that there is no significant difference between the average ranks of k classifiers on n datasets, against the alternative hypothesis that at least one classifier's mean rank is different, is tested as follows:

Given M, the k by n matrix of classification accuracies where $m_{i,j}$ is the accuracy of the i^{th} classifier on the j^{th} dataset,

$$M = \begin{bmatrix} m_{1,1} & m_{1,2} & \cdots & m_{1,n} \\ m_{2,1} & m_{2,2} & \cdots & m_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ m_{k,1} & m_{k,2} & \cdots & m_{k,n} \end{bmatrix} \tag{3.3.4}$$

the corresponding n by k matrix R, is calculated, where $r_{i,j}$ is the rank of the i^{th} classifier on the j^{th} dataset:

$$R = \begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,n} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ r_{k,1} & r_{k,2} & \cdots & r_{k,n} \end{bmatrix} \tag{3.3.5}$$

From R the average rank of classifier j is calculated as

$$\bar{r} = \frac{\sum_{i=1}^{n} r_{i,j}}{n} \tag{3.3.6}$$

To test the hypothesis, the Friedman statistic Q,

$$Q = \frac{12_n}{k(k+1)} \cdot \left[\sum_{j=1}^{k} \bar{r_j}^2 - \frac{k(k+1)^2}{4} \right] \tag{3.3.7}$$

can be approximated using a Chi-squared distribution with $(k-1)$ degrees of freedom to test the null hypothesis that there is no difference between the mean ranks of the classifiers. However, because this calculation is often conservative, Demšar (2006) proposed and used the following statistic:

$$F = \frac{(n-1)Q}{n(k-1) - Q} \tag{3.3.8}$$

Under the null hypothesis this statistic follows an F distribution with $(k-1)$ and$(k-1)(n-1)$ degrees of freedom. If the result of this calculation is that we can reject the null hypothesis, stage two can then be performed.

Stage two. *Post-hoc* pair-wise Nemenyi tests are then employed to identify where significant differences occur. This test states that the average ranks of two classifiers are significantly different if they differ by at least the *critical difference*, which is calculated as:

$$CD = q_a \sqrt{\frac{k(k+1)}{6n}} \tag{3.3.9}$$

where q_a is based on the studentised range, where the difference between the largest and smallest values in the sample is measured in units of standard deviation. A critical difference diagram is then created which summarises the results. In such a diagram the average ranks for each classifier are labelled on a numerical range, classifiers that are not significantly different from one another are organised into "cliques" indicated by solid black lines. When two classifiers do not belong to at least one common clique, the average ranks of the classifiers are significantly different.

We have used critical difference diagrams to present comparisons of results obtained with different classifiers, for an example see Figure 4.4.

3.4 Data

The main important characteristic in which dataset used in this research is to have more than one type of media for each instance in the dataset. Finding a reasonable number of heterogeneous datasets D_M that contain more than one

type of media is one of the difficulties in this research.

The dataset which has been used to investigate the methodologies of this research is the 8 Scene Categories Dataset [1], (Oliva and Torralba, 2001). Table 3.3 describes the distribution of the database. This dataset was chosen because it has two distinct parts: structured text and images, comprising 2688 images and their annotations represented by XML files.

Table 3.3: Eight Scenes Category Database

class	# of instances	% of the data
tall buildings	356	13.24
inside city	308	11.46
street	292	10.86
highway	260	09.67
coast	360	13.39
open country	410	15.25
mountain	374	13.91
forest	328	12.20

3.4.1 The Text Extracted Features Dataset (D_t)

The String to Word Vector filter was used for the extraction of features from the structured text data as discussed in Section 3.2.1.1. Using this technique on the raw data gave us a feature space which contained 782 binary features. These all started in D_t .

3.4.2 The Images Extracted Features Dataset (D_g)

HOG was used to extract the images feature space from the raw data. HOG was used as described in section 2.7.2.1. The size of the images was 256 X 256 pixels and the parameters were set as a cell 32 X 32, block 8 X 8 nine bins. The resulting

[1] http://people.csail.mit.edu/torralba/code/spatialenvelope/

feature space was 567 histogram values between 0 and 1.0 for each instance. This started in D_g.

Moreover, we examined ten different feature extraction methods and found that HOG was the most appropriate for our experiments. It gives better results when we combine accuracy and diversity, which our approaches are investigating in the model selection stage.

It should be pointed out that, since this research is about developing heterogeneous ensembles for multi-media data, it will not investigate the pros and cons of available methods for feature extraction. Rather, commonly available techniques have been selected to allow us to move directly to data collection and analysis.

We only used one dataset in our research because despite an extensive search we were unable to identify any other datasets that were suitable. No other dataset suitable for supervised learning contained data of more than one type of media.

3.5 Summary

This chapter describes the research design and methodology for implementing the conceptual framework. The conceptual framework contains the two fundamental phases which are data representation and fusion phase, and constructing ensemble phase. Each of these phases has some stages. The stages in the first phase are feature extraction and data representation. The stages for the second phase are modelling the dataset, model selection and the combination stage. This conceptual framework can be implemented with different stages which: are feature level ensemble (FLE) and decision level ensemble (DLE). However, it can be seen that, the common and core structure of these two strategies is an ensemble that is built

with different types of models, which is defined as heterogeneous ensemble, because it is generally considered that different types of models may produce a high diversity between them and hence are more likely to generate better ensembles. Therefore we will investigate how to build a heterogeneous ensemble first, before working in the FLE and DLE. The work on building effective heterogeneous ensembles with different rules based on accuracy and diversity will be presented in Chapter 4.

Chapter 4

Heterogeneous Ensemble for Classifying Single Media Data

In Chapter 3 we discussed the research methodology, methods and tools that we will use to conduct our research into ensemble classification. In this chapter we investigate the use of the heterogeneous ensemble systems for classifying multimedia data. The research work and results have been published in the 8th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (Alyahyan et al., 2016).

4.1 Introduction

In this chapter we conduct two experiments. The main experiment was given the name Heterogeneous Ensemble System for the Text data (HEST). The same experiment was then applied to the graphic data and named Heterogeneous Ensemble System for Graphic data (HESG). In these experiments we constructed a heterogenous ensemble which deals with a single media dataset. Three rules for model selection were applied to this ensemble. These rules take into account accuracy and diversity.

The rest of the chapter is organized as follows. Section 4.2 briefly discusses

several of the previous studies in the field. Section 4.3 details our methods, listing the tools and programs used in the research. Section 4.4 provides details of the experiment conducted and our results. Section 4.5 presents our conclusions.

4.2 The Heterogeneous Ensemble System (HES)

4.2.1 The Framework of the HES

The proposed heterogeneous ensemble system as shown in Figure 4.1, consists of five main components: 1, feature extraction and data formation; 2, data partition; 3, heterogeneous model generation and evaluation; 4, ensemble construction and 5, decision fusion function. The key idea of the proposed heterogeneous ensemble system (HES) is to generate methodologically different models, hence called heterogeneous models, by different learning algorithms, as the member candidates and then build an ensemble with the rules as defined below.

The main operations of the HES are shown by Algorithm 1. It starts by dividing D into a training dataset and testing dataset Ts. The training dataset was further divided to train dataset Tr for training the classifiers $C_i \in C$ and validation dataset Val for evaluating each C_i. Different learning algorithms are called from the learning algorithms base to generate $|C|$ models,which are stored in a model pool PM.

4.2.2 Rules for Building Different HES

Different rules can be devised to build various heterogeneous ensembles based on different strategies and purposes. Three rules R0, R1, and R2 are defined in this study as the demonstration of concept in utilising the accuracy as a model

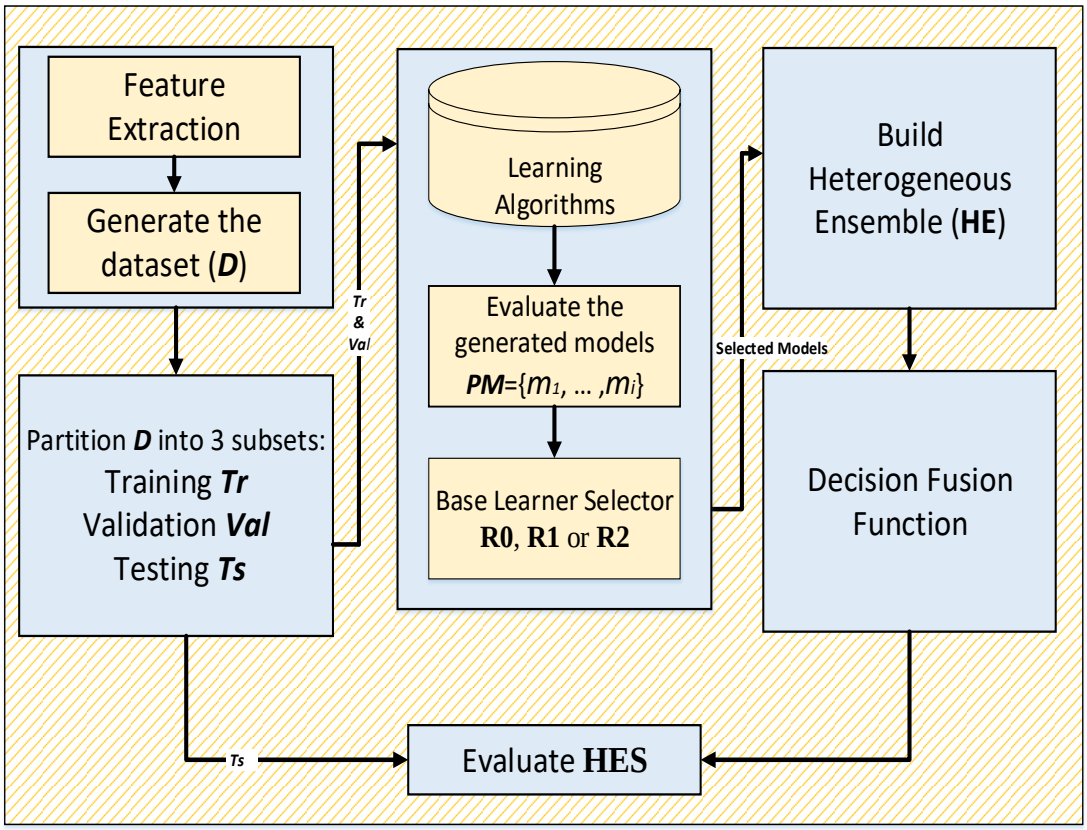

Figure 4.1: The general framework for HES

Algorithm 1 Algorithm for Building **HES**

Input: $\boldsymbol{D}$ dataset, **C** base learners, ensemble size $|\Phi|$ and the selected rule $\boldsymbol{R}$.
Output: Acc(**HES**).
Divide $\boldsymbol{D}$ to Train 75% and $\boldsymbol{Ts}$ 25%
Divide the training data to $\boldsymbol{Tr}$ 75% and $\boldsymbol{Val}$ 25%
let $\boldsymbol{N} = |\Phi|$
for $i = 1$ to $|\boldsymbol{C}|$ **do**
    m_i = model resulted from training $\boldsymbol{Tr}$ on C_i
    add m_i to $\boldsymbol{PM}$
    Evaluate m_i on $\boldsymbol{Val}$
end for
Call the selected rule R
Evaluate HES on $\boldsymbol{Ts}$

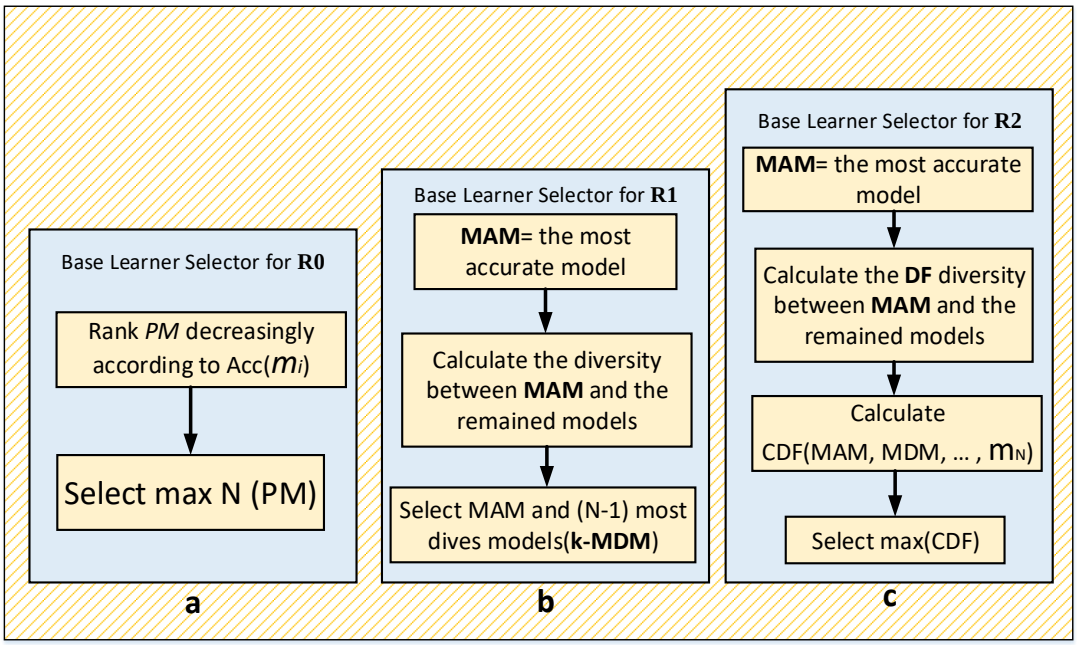

Figure 4.2: Main steps for R0, R1 and R2 in HES

selection criterion alone, or both accuracy and diversity measures.

Figure 4.2 shows all three rules and the details of these rules are described below.

4.2.2.1 Rule R0:

To build an HES, this rule only considers the accuracy of individual models. Algorithm 2 describes how it works where the HES will first sort models in the PM in a descending order according to the accuracy of each individual model $Acc(m_i)$ on Val. Then, the most accurate N models are selected from PM to be added to Φ. This is the basic rule applied in HES, and also forms a part of all other rules in the system. Figure 4.2a illustrates how this rule works. To select the models we need to use Equation 4.2.1.

$$m_i = max\{Acc(m_j), m_j \in PM\}\, i = 1...N \tag{4.2.1}$$

Algorithm 2 Algorithm for **R0**

1: **Input:** *PM*
2: **Output:** The selected models
3: sort models in the ***PM*** decreasingly according to $acc(m_i)$
4: select first ***N*** models from ***PM***
5: add selected models to Φ

4.2.2.2 Rule R1:

To build a HES, this rule considers both accuracy and diversity measured by pair-wise diversity. Algorithm 3 describes how it works. In this rule, **HES** first selects the most accurate model *MAM* from *PM* to be added to Φ. Then this model is removed from the pool *PM*.

$$m_1 = max\{Acc(m_j), m_j \in PM\} \tag{4.2.2}$$

Then, the diversity measured by Double-Fault (*DF*) (Giacinto and Roli, 2001) between *MAM* and every model in the pool *PM* is calculated using a pairwise strategy to fill the models needed for the final Φ. Then *PM* is sorted in the decreasing order according to their diversity *DF* to select *N-1* most diverse models from the pool *PM* to be added to the final Φ. Equation 4.2.3 is applied for this stage. The models selected in this rule are *MAM* and *N-1* most diverse models from *MAM* in the pool *PM*. Figure 4.2b illustrates how this rule works.

$$m_i = max\{DF(m_1, m_j), m_j \in PM\}\, i = 2...N \tag{4.2.3}$$

Algorithm 3 Algorithm for **R1**

Input: $\boldsymbol{PM}$
Output: The selected models
$\boldsymbol{MAM}$=the most accurate model in $\boldsymbol{PM}$
add $\boldsymbol{MAM}$ to Φ
remove $\boldsymbol{MAM}$ from $\boldsymbol{PM}$
for $i = 1$ to $|\boldsymbol{PM}|$ **do**
    calculate $\boldsymbol{DF}$_ diversity ($\boldsymbol{MAM}$,m_i)
end for
sort $\boldsymbol{PM}$ decreasingly according to their diversity
select first ($\boldsymbol{N}$-1)models
add selected models to Φ

4.2.2.3 Rule R2:

This rule uses both accuracy and two diversity measures: *DF* and Coincident Failure Diversity (*CFD*) (Partridge and Krzanowski, 1997). Algorithm 4 describes the procedure of R2. In this rule, the first model m_1 to be selected for the Φ is chosen as in equation (2) in R1, which is *MAM*. The second model m_2 to be selected for Φ is the most diverse model *MDM* from the most accurate model in the pool *PM*. To calculate *MDM*, Equation 4.2.4 is used.

$$m_2 = max\left\{DF(m_1, m_j), m_j \in PM\right\} \tag{4.2.4}$$

In this rule, we generate a number of combinatorics J, subsets of models ϕ_i from the pool of models PM and Equation 4.2.5 to calculate this number.

$$J = \binom{|PM|}{N-2} \tag{4.2.5}$$

Each combinatory ϕ_i includes *MAM* and *MDM*, and the remaining models needed to reach to N are added from the pool *PM* to compute the diversity *CFD*. Thus the maximum diverse subset ϕ_i ensemble is chosen for the final Φ. Figure 4.2c, illustrates how this rule works.

$$HES = max\{CFD(\Phi \Leftarrow m_j), m_j \in PM\} \quad (4.2.6)$$

Algorithm 4 Algorithm for **R2**

Input: *PM*
Output: The selected models
MAM=the most accurate model in ***PM***
remove ***MAM*** from ***PM***
for $i = 1$ to $|\boldsymbol{PM}|$ **do**
    calculate ***DF***_diversity (***MAM*** ,m_i)
end for
MDM = the most divers model from ***MAM***
remove ***MDM*** from ***PM***
J= The number of Combinations subsets $\binom{|PM|}{N-2}$
for $i = 1$ to$j$ **do**
    ϕ_i =the i^{th} combinations subset from ***PM***
    add MAM and MDM to ϕ_i
    calculate ***CFD***_diversity ϕ_i
end for
add the most divers ϕ_i to Φ

4.2.3 Implementation of HES

The HES is implemented with Java, based on Weka API. Thus, the experiment was carried out on a standard PC, with an Intel i7 processor and 16 GB RAM. As HES is flexible for selecting candidate classifiers, we have selected 11 different base classifiers that are provided in the WEKA library. These base classifiers are: trees (*J48, RandomTree, REP-Tree*), bayes (*NaiveBayes, BayesNet*), function (*SMO*), rules (*JRip, PART*) and Lazy (*IBk, LWL*).

The HES framework applied in two different representation of the MMD. The first one was in D_t to generate Heterogeneous Ensemble System for the Text data(HEST). The second one was in D_g to generate Heterogeneous Ensemble System for the Graphic data(HESG).

4.3 HEST Experiment

4.3.1 Experiment Procedure and Set-up

We conducted a series of experiments investigating three rules in HES. They are generated by changing two factors. The first is the rule used in the experiment, which is R0, R1 or R2. The second is the ensemble size, which is 3, 5, 7 or 9. Running all possible combinations of these parameters, and repeating them for five different runs lead to conduct 60 experiments in total.

4.3.2 HEST Results

The 10 base learning algorithms used in all conducted experiments in HEST are trained and examined in the test part and the validation part. Table 4.1 shows the result for the results of the accuracy for the 10 base learning algorithms in the testing set and the validating set for the five different runs.

The results (mean and standard deviation) of using R0, R1 and R2 with different numbers of models in HEST are shown in Appendix C Figure C.1, over 5 runs on each sub-figure. Moreover, the selected models and its diversity CFD for each experiment are shown in Appendix B, Tables B.1–B.4.

Table 4.1: The results for all single models used in HEST for all five runs.

	Run1				Run2			Run3			Run4			Run5		
	Test	Time	Val	Time	Test	Time	Val	Time	Test	Time	Val	Time	Test	Time	Val	Time	Test	Time	Val	Time
J48	87.20	0.01	86.31	0.003	87.50	0.009	87.70	0.003	87.65	0.002	89.48	0.001	87.80	0.001	89.48	0.001	87.05	0.002	89.09	0.001
NaiveBayes	90.48	1.086	87.90	0.835	91.67	0.946	90.87	0.697	90.33	0.96	91.27	0.739	90.63	0.947	91.87	0.708	90.48	0.934	92.06	0.706
SMO	92.56	0.047	89.68	0.018	91.96	0.026	92.46	0.018	91.07	0.025	91.47	0.019	91.37	0.03	92.26	0.018	91.96	0.024	91.47	0.018
BayesNet	90.33	0.204	89.29	0.14	91.37	0.212	92.46	0.147	90.03	0.195	92.26	0.154	91.52	0.182	90.67	0.137	90.77	0.187	92.06	0.141
IBk	86.90	3.339	88.89	2.522	85.71	3.298	88.29	2.47	87.35	3.365	88.29	2.561	85.42	3.249	85.52	2.508	86.31	3.202	89.68	2.505
JRip	87.05	0.002	85.52	0.001	85.27	0.002	83.33	0.001	83.18	0.001	84.13	0.001	85.12	0.002	85.32	0	84.52	0.001	83.53	0.001
RandomTree	78.72	0.005	80.56	0.002	82.29	0.002	80.75	0.001	80.21	0.002	80.95	0.001	80.95	0.002	80.16	0.002	79.61	0.002	79.96	0.001
PART	85.86	0.004	88.10	0.002	85.12	0.002	88.89	0.001	88.54	0.002	90.67	0.001	90.18	0.002	90.87	0.001	87.95	0.002	89.88	0.002
REPTree	87.35	0.448	85.32	0.001	85.71	0.329	85.52	0.001	82.14	0.315	83.93	0.001	86.01	0.355	84.52	0.001	83.33	0.312	85.71	0.001
LWL	58.93	78.822	58.73	59.089	54.46	80.958	57.34	62.756	57.89	78.067	61.31	58.115	55.65	75.329	57.94	56.389	60.42	74.596	58.33	56.402

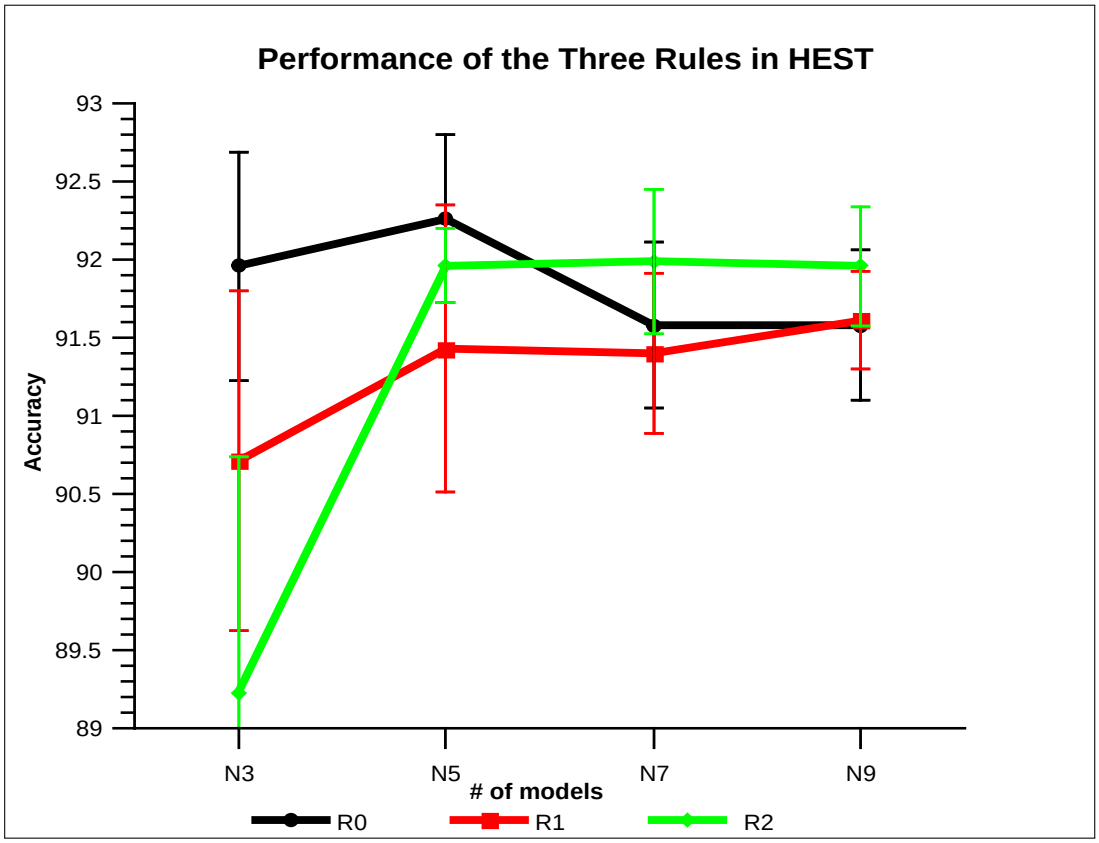

Figure 4.3: Comparing all three rules in four different sizes of the HES

The results for all five runs on all three rules are about as accurate as those of the most accurate model MAM but more reliable because the single best model varied in different runs and could be much worse in some runs. In this study the most accurate model was not stable for all the five runs it was some times *BayesNet* and other times *SMO*. This negatively impacts reliability. Thus, ensemble accuracy wins against the most accurate model in certain instances.

The most significant finding from applying the three rules was the stable improvement of the level of the accuracy when R2 is applied, as seen in Figure 4.3. The observable reason for that is R2 considers more diversity measures than R0 and R1. Considering more diverse models provided an opportunity to achieve stable results even if the mean accuracy for these models was low. This is a clear evidence that R2 can increase reliability whilst maintaining high accuracy.

Another notable finding from the results is that increasing the number of models used in the ensemble supported with the diversity among them lead to more stable results, as shown in Figure 4.3. For R2, when more than five models were selected for the ensemble, the results became more stable.

When there were three models in R2, the accuracy was lower than for the other rules. That was probably because when the size of an HES is as small as 3, and we add a more diverse but less accurate model to it, the diversity introduced is not enough to compensate the loss of the accuracy caused by the third less accurate model, so the best chance of making use of the diversity measure to improve performance is more likely to be effective when the number of models for the ensemble is increasing.

4.3.3 Comparison of the HEST Results

The comparison was carried out with some other ensemble methods, including various homogeneous ensembles built with the AdaBoost algorithm for each base classifier used in HEST.

Table 4.2 shows the results for homogeneous ensembles over all the five runs conducted. It can be seen that these homogeneous ensembles produced quite different or unstable accuracies for the task with the highest being 90.95% and lowest 78.07%.

Table 4.2: The accuracy of five runs using the AdaBoostM1 method for each base classifier in HEST.

	Run 1	Run 2	Run 3	Run 4	Run 5	Mean	SD
J48	89.29	90.03	89.43	89.73	89.29	89.55	0.32
NaiveBayes	89.73	89.58	90.48	90.63	89.29	89.94	0.58
SMO	90.48	**91.67**	**90.63**	**90.92**	91.07	90.95	0.46
BayesNet	**90.92**	90.77	90.33	90.03	**91.37**	90.68	0.52
IBk	86.61	84.97	87.20	85.57	86.01	86.07	0.87
JRip	88.10	88.24	87.35	88.54	88.24	88.10	0.45
RandomTree	83.78	82.29	81.55	82.44	87.05	83.42	2.18
PART	87.50	88.99	90.18	89.58	89.43	89.14	1.01
REPTree	89.29	87.35	88.84	88.99	88.69	88.63	0.75
LWL	76.64	81.70	71.28	81.40	79.32	78.07	4.30
Mean	87.23	87.56	86.73	87.78	87.98	87.46	
SD	4.27	3.48	6.07	3.44	3.45	4.00	

Table 4.3 shows the comparison between the homogeneous ensemble and (R0, R1 and R3) in HEST. It is very clear that heterogeneous ensemble constructed by any of the three rules are the best and improved the average accuracy as much as 3.5% from the mean of AdaBoostM1.

The statistical test was carried out over the five different runs. For R0, R1 and R2 we counted the mean for four different sizes which are 3, 5, 7 and 9. For

Table 4.3: Comparison of results with the homogeneous ensemble AdaBoostM1 and HEST for all the three rules.

	Mean Accuracy	SD
Best AdaBoostM1	90.95	0.46
Mean AdaBoostM1	87.51	3.84
Rule R0	91.85	0.33
Rule R1	91.29	0.39
Rule R2	91.29	1.37

the best homogeneous ensemble we chose the most accurate ensemble each run, no matter which base learning algorithm was used. The highest two were AdaBoostM1 obtained by SMO and BayesNet so we included them on the statistical test.

Figure 4.4 shows the critical difference diagram for the purpose of comparison. It shows that the best homogeneous ensemble ranked as the highest result and our method R2 was the second, but in fact the best homogeneous ensemble was not stable in all five runs. It was AdaBoostM1(SMO) sometimes and AdaBoostM1(BayesNet) other times. So that we included them in the statistical test, and we opined that R2 was better than them. R0 and R1 were not better than AdaBoostM1(SMO) and AdaBoostM1(BayesNet) but it was not significantly worse because they were within the same horizontal line.

4.4 HESG Experiment

This section describes an experiment conducted using HESG, which uses the same experimental design as HEST, as described earlier. In this experiment we feed our framework with D_g instead of using D_t. The main purpose of running the HESG experiment was to prepare for the later multimedia experiments which combined HEST and HESG, and to investigate the relative effectiveness of using

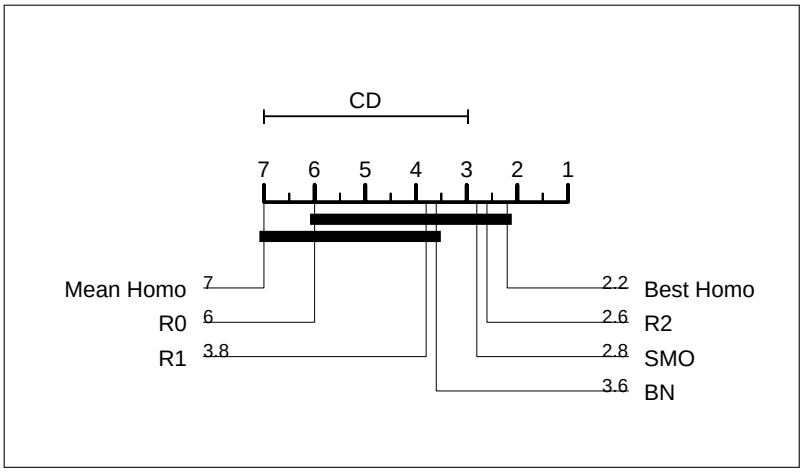

Figure 4.4: Critical difference digram showing the differences between the results obtained by R0, R1, R2, Mean of homogeneous ensemble, Best homogeneous ensemble, AdaBoostM1(SMO) and AdaBoostM1(BayesNet).

single and multi-media datasets.

4.4.1 Images Features Extraction

There are several feature extraction methods which are used to represent an image dataset in a vector space. WEKA has implemented 10 of them and we tested all of them experimentally in the dataset. Table 4.4 presents the result for each method, applying all the base learning algorithms used in the previous experiments. It was important to select the most appropriate WEKA extraction method to apply in the HESG experiment. Therefore, we calculated the mean accuracy and the standard deviation for each method. In order to find a good measurement with which to make the selection, the mean of accuracy and the stranded deviation were combined to take into account both the accuracy and diversity. HOG produced the best result and was therefore selected.

Table 4.4: Accuracies of all base learning classifiers for different features extraction methods using 10 fold cross-validations.

Features Extraction Method	SMO	BayesNet	NaiveBayes	Ibk	Jrip	PART	J48	RandomTree	REPTree	LWL	Mean	SD	Mean + SD	#Att
HOG FeatuersV2 576F	80.92	73.92	72.69	68.45	61.38	59.86	55.21	53.27	56.70	37.39	61.98	12.55	**74.53**	576
AutoColorCorrelogramFilter	54.65	48.18	43.60	54.32	41.63	39.58	37.80	35.90	41.96	29.02	42.66	8.01	50.67	1024
BinaryBatternsPyramidFilter	67.41	57.11	54.02	56.66	50.63	47.88	46.13	38.24	48.33	36.94	50.33	9.09	59.43	756
ColorLayoutFilter	51.34	55.39	55.65	51.64	45.05	46.39	46.50	41.26	46.39	23.40	46.30	9.28	55.58	33
EdgeHistogramFilter	76.90	73.10	73.29	72.36	64.17	61.64	57.48	54.69	60.97	53.83	64.84	8.47	73.31	80
FCTHFilter	68.04	56.21	54.06	58.97	54.24	53.13	54.02	46.58	54.91	31.44	53.16	9.34	62.50	192
FuzzyOpponentHistogramFilter	36.31	38.36	23.21	38.47	31.25	35.16	35.90	29.72	38.69	22.36	32.94	6.13	39.07	576
GaborFilter	21.47	23.66	23.40	16.85	17.15	19.98	19.31	17.45	22.99	21.54	20.38	2.62	23.00	60
JpegCoefficientFilter	79.80	59.45	58.22	70.13	62.83	61.38	58.26	54.20	60.38	31.21	59.59	12.35	71.93	192
PHOGFilter	73.47	69.83	68.60	67.67	63.73	61.46	60.27	54.87	62.09	46.21	62.82	7.95	70.77	630
SimpleColorHistogramFilter	46.58	46.84	30.69	44.83	37.61	42.78	43.30	37.43	42.82	31.36	40.42	5.89	46.31	64

4.4.2 HESG Results

The 10 base learning algorithms used in all conducted experiments in HESG are trained and examined in the test part and the validation part. Table 4.5 shows the result for the results of the accuracy for the 10 base learning algorithms in the testing set and the validating set for the five different runs.

Table 4.5: The results for all single models used in HESG for all five runs.

	Run1				Run2				Run3				Run4				Run5			
	Test	Time	Val	Time	Test	Time	Val	Time	Test	Time	Val	Time	Test	Time	Val	Time	Test	Time	Val	Time
J48	54.61	0.011	58.53	0.004	57.74	0.009	57.54	0.005	57.59	0.001	56.35	0.002	50.89	0.001	52.58	0.001	56.55	0.001	56.35	0.001
NaiveBayes	70.09	0.654	74.60	0.533	72.02	0.516	72.22	0.379	70.09	0.517	71.03	0.381	72.92	0.515	69.64	0.38	72.02	0.503	72.02	0.373
SMO	78.13	0.047	79.96	0.011	77.08	0.015	78.77	0.012	76.79	0.013	77.18	0.01	78.42	0.013	75.20	0.01	78.42	0.013	79.76	0.009
BayesNet	71.28	0.172	75.00	0.093	72.47	0.117	73.41	0.088	72.32	0.115	71.63	0.087	74.70	0.118	73.02	0.087	73.66	0.111	74.60	0.085
IBk	67.56	2.947	70.44	2.138	66.67	2.76	67.66	2.112	67.71	2.84	65.08	2.094	68.15	2.734	66.07	2.092	63.10	2.702	67.86	1.985
JRip	60.27	0.002	61.90	0.002	59.38	0.001	56.94	0.001	59.38	0.002	58.33	0.001	56.85	0.002	53.77	0	60.86	0.001	60.91	0.001
RandomTree	50.89	0.003	55.36	0.001	50.89	0.001	51.59	0.001	48.81	0.001	50.60	0.001	51.04	0.001	50.99	0.001	49.85	0.001	53.77	0.002
PART	57.29	0.005	62.70	0.002	55.80	0.002	55.95	0.001	53.27	0.002	59.33	0.002	55.65	0.003	54.76	0.003	61.31	0.002	58.53	0.002
REPTree	55.21	0.593	56.75	0.001	55.80	0.464	50.40	0.001	56.40	0.465	58.33	0.001	52.38	0.515	54.17	0.001	54.17	0.437	59.72	0.002
LWL	39.73	296.648	40.08	221.907	39.58	298.616	41.67	220.318	36.01	298.536	37.50	221.351	41.67	294.805	43.25	217.056	35.27	287.247	36.31	215.33

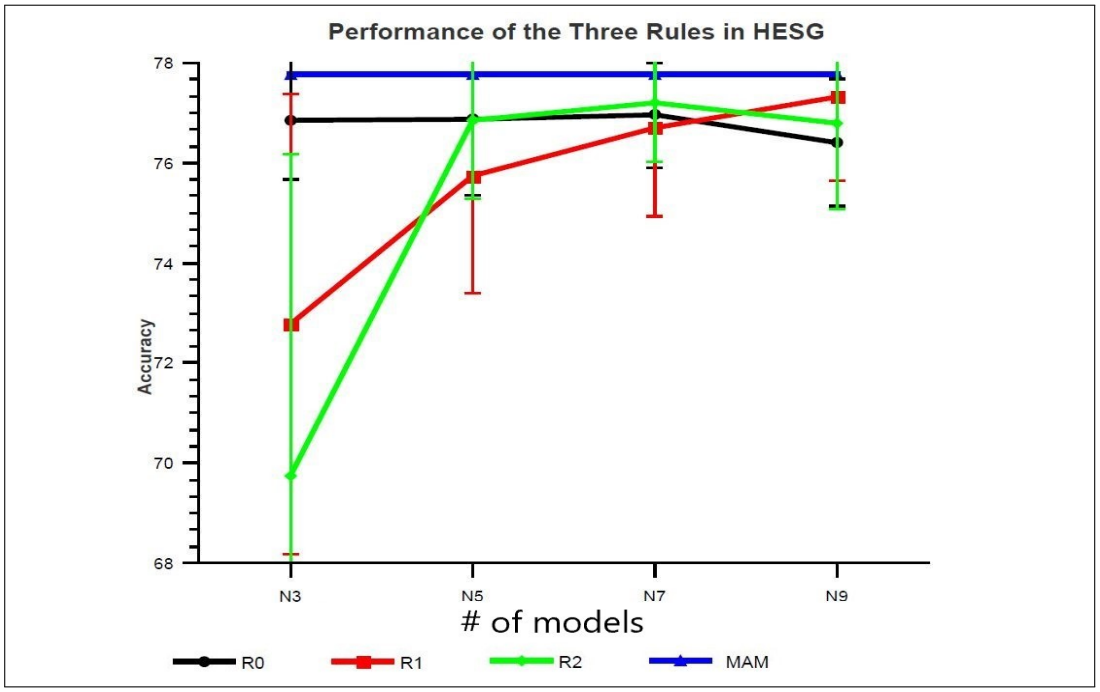

Figure 4.5: Comparing all three rules in four different sizes of the HESs for the image dataset only.

The results of HESG, obtained in these experiments, are shown in Appendix C Figure C.2, and the summary of the results of varying the ensemble size from 3 to 9 on the test dataset for each of the three rules is shown by Figure 4.5. Moreover, the selected models and its diversity CFD for each experiment are shown in Appendix B, Tables B.5–B.8.

4.4.3 Comparison of the HESG Results

The comparison was carried out with some other ensemble methods, including various homogeneous ensembles built with the AdaBoost algorithm for each base classifier used in HESG.

Table 4.6 shows the results for homogeneous ensembles over all the five runs conducted. It can be seen that these homogeneous ensembles produced quite different or unstable accuracy for the task with the highest being 76.46% and lowest 39.38%.

Table 4.6: The accuracy for five runs using AdaBoostM1 method for each base classifier in HESG.

	Run 1	Run 2	Run 3	Run 4	Run 5	Mean	SD
BayesNet	73.36	75.45	74.11	75.60	**75.45**	74.79	1.00
NaiveBayes	70.09	72.02	70.09	72.92	72.02	71.43	1.28
SMO	**76.79**	**77.23**	**75.15**	**77.68**	75.44	76.46	1.11
IBk	67.56	66.22	68.90	68.15	63.24	66.82	2.22
Jrip	70.54	70.98	72.32	70.54	71.88	71.25	0.81
PART	72.02	69.79	71.13	71.58	71.13	71.13	0.84
J48	70.39	73.36	71.73	72.32	69.94	71.55	1.40
RandomTree	53.13	61.90	49.26	59.97	53.42	55.54	5.24
REPTree	69.05	68.15	72.47	67.41	70.09	69.43	1.97
LWL	43.01	39.73	36.61	42.11	35.42	39.38	3.32
Mean	66.59	67.49	66.18	67.83	65.80	66.78	
SD	10.36	10.72	12.73	10.26	12.51	11.32	

Table 4.7 shows the comparison between the homogeneous ensemble and (R0, R1 and R3) in HES. It is very clear that heterogeneous ensemble constructed by any of the three rules are the best and improved the average accuracy as much as 3.5%.

The statistical test was carried out over the five different runs. For R0, R1 and R2 we counted the mean for the different four sizes which are 3, 5, 7 and 9.

Table 4.7: Comparison of results with the homogeneous ensemble and HESG for all the three rules.

	Mean Accuracy	SD
Best AdaBoostM1 Ensemble	76.46	1.11
Mean AdaBoostM1 Ensemble	66.78	0.86
Rule R0	76.77	1.26
Rule R1	75.63	2.60
Rule R2	75.23	2.95

For the best homogeneous ensemble we chose the most accurate ensemble each run, no matter which base learning algorithm was used.

Figure 4.6 shows the critical difference digram for the purpose of comparison. It shows that our method R0 is ranked highest and out performed the best homogeneous ensemble, which in this case is AdaBoostM1(SMO). R1 and R2 are equal and worse than AdaBoostM1(SMO) but they are not significantly worse because the were within in the same horizontal line.

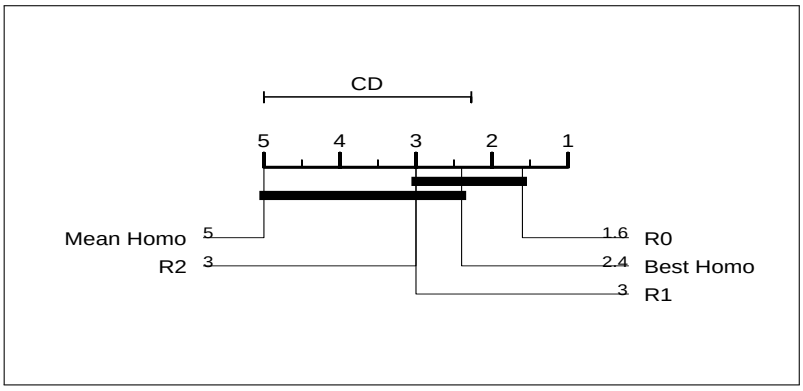

Figure 4.6: The critical difference digram shows the differences between the results obtained by R0, R1, R2, Mean of homogeneous ensemble, Best homogeneous ensemble.

4.5 Summary

This chapter used an image scene classification problem as a testing case to investigate the capability of heterogeneous ensembles built with the rules that

consider either accuracy of individual models or diversity, or both. Three rules are devised specifically using accuracy of individual models and the diversity measurements among these models for an ensemble. The increasing diversity among the models selected for the ensemble was found to be advantageous, leading to more stable and reliable results. Our research found that increasing the number of models also affects the ensemble's results. This indicated that diversity is more effective when used with a higher number of models selected for the ensemble. It can therefore be concluded that combining models results in high accuracy and diversity for an ensemble has considerable advantages in terms of the ensemble's accuracy. The results show that HESG gives poor performance compared with the result obtained by HEST, which is because dataset D_g is hard for base learning algorithms to learn and it gives the worst results in individual models.

In this chapter we have investigated the use of heterogeneous ensemble systems for classifying a single medium data. In Chapter 5 we will investigate heterogeneous ensemble systems that combine data from each individual type of medium at the feature level prior to data aggregation and ensemble generation using the combined dataset.

Chapter 5

Feature Level Ensemble Method

In Chapter 4 we investigated the use of heterogeneous ensemble systems for classifying multimedia data. In this chapter we will investigate ensemble systems that combine data from each individual type of medium at the feature level prior to data aggregation and ensemble generation using the combined dataset. Hence this method will be called Feature Level Ensemble Method (FLEM). The research work and results have been published in the Thirty-seventh SGAI International Conference on Artificial Intelligence (Alyahyan and Wang, 2017).

5.1 Introduction

In this chapter we conduct Feature Level Ensemble Method (FLEM) experiments. We use heterogeneous ensembles to classify multimedia data which was prepared by aggregating HESG and HEST into one single dataset. This was then fed through our heterogeneous ensemble system, which used three different model selection rules that took into account accuracy, diversity and both together.

5.2 The Feature Level Ensemble Method

5.2.1 The Framework of the Feature Level Ensemble Method

The proposed feature-level ensemble method (FLEM), as illustrated in Figure 5.1, consists of four modules/stages: multimedia data aggregation module, modelling module, model selection module and combination module.

In general, a multimedia dataset (MMD) should consist of several subsets of various media, e.g. text, images, audio, etc. The FLEM starts with extracting D_i's features ($1 \leq i \leq n$), from each subset of the MMD by using appropriate feature extraction methods. Then, all features are normalised and aggregated to form one big dataset, i.e, $D = N(D_1 \cup D_2 \cup D_3 \cup ... \cup D_n)$. These operations are usually called feature aggregation, which is why our approach is called Feature-Level Ensemble Method (FLEM).

The second stage is to generate various types of individual models, m_i ($1 \leq i \leq n$), to create a pool of models, $PM = \{m_1, m_2, ..., m_n\}$ as the member candidates of ensemble. The models are called homogeneous models if they are generated by using the same learning algorithm with variations on its parameters and/or data partitions, or called heterogeneous models if they are generated by using different algorithms. A homogeneous ensemble is built with just homogeneous models, whilst a heterogeneous ensemble is constructed with heterogeneous models. In this study, over 10 different base learning algorithms have been selected to generate homogeneous and heterogeneous individual models.

The third stage involves model selection based on a set of defined criteria and rules. In this study, *accuracy* and *diversity* are used as selection criteria either

separately or jointly. Three different rules for model selection, $R0, R1$ and $R2$ as described in Chapter 4, are used. Finally, the selected models are combined into one ensemble and their classification decisions are aggregated using a combination method *majority voting* to reach the final form of the ensemble.

5.2.2 Implementation of FLEM

The FLEM is implemented with Java, based on Weka API. The experiment was carried out on a standard PC with an Intel i7 processor and 16GB RAM. As FLEM is flexible for selecting candidate classifiers, we selected 10 different base classifiers provided in the WEKA library, which are: three types of decision trees (*J48, RandomTree, REP-Tree*), two Bayesian methods (*NaiveBayes, BayesNet*), Support vector machine(SMO), two rule induction methods(*JRip, PART*) and two lazy learners (*IBk and LWL*).

5.3 Experiment Design and Results

5.3.1 Experiment Design and Results

5.3.1.1 FLEM Experiments

We conducted a series of experiments to investigate the performance of FLEM working with three selection rules separately on the multimedia data. The factors that were investigated include (1) the performance measures and criteria for selecting classifiers, which are represented by the three rules R0, R1 and R2, (2) the size of ensemble (set at 3, 5, 7 or 9), and (3) the salience of multimedia data, i.e. if the combined multimedia data MDM can produce better results, compared

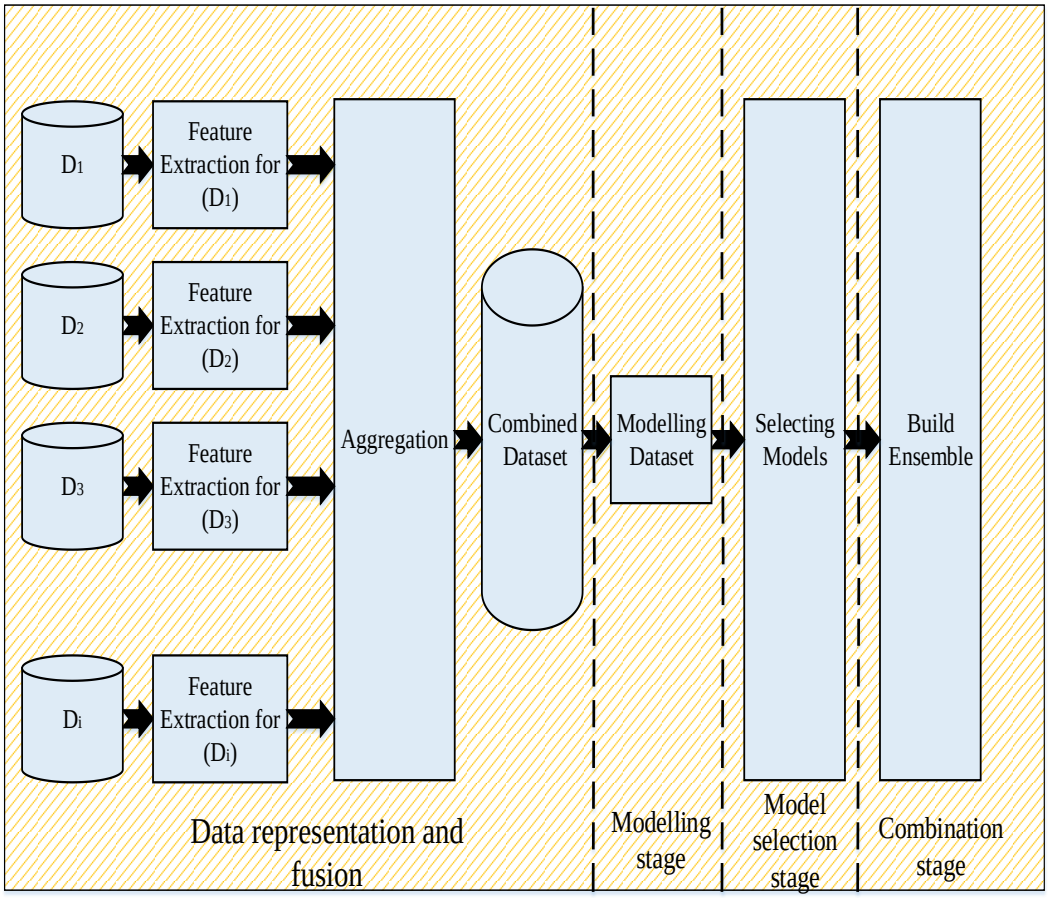

Figure 5.1: A general framework for the feature-level ensemble method (FLEM).

with each of the single-media data subsets: D_t and D_g. For each specific set-up, the experiment was repeated 5 times with different data partitions to check consistency.

5.3.1.2 HOMOFLEM Experiments

In addition, for comparison with heterogeneous ensemble , homogeneous ensembles (**HOMOFLEMs**) were built with the classifiers selected only from the same type. As ten different types of base learning algorithms were used for generating classifiers, ten homogeneous ensembles were constructed for each set-up of factors listed above. The homogeneous models for each base learning algorithm were generated by manipulating the training dataset. Each training dataset for the homogeneous ensemble models was generated by randomly sampling 75% of the original training dataset. This was repeated ten times to give ten different samplings, which were used to generate ten 10 homogeneous models.

Therefore, for all possible combinations of these parameters, 600 sets of experiments were conducted in total.

5.3.1.3 Results from FLEMs built with three rules and variable sizes:

The 10 base learning algorithms used in all conducted experiments in HEST and HESG are trained and examined in the test part and the validation part. Table 5.1 shows the result for the results of the accuracy for the 10 base learning algorithms in the testing set and the validating set for the five different runs.

The results of FLEM, obtained in these experiments, are shown in Appendix C Figure C.3.

Table 5.1: The results for all single models used in FLEMfor all five runs.

	Run1			Run2			Run3			Run4			Run5		
	Test	Time	Val	Time	Test	Time	Val	Time	Test	Time	Val	Time	Test	Time	Val	Time
J48	83.48	0.011	83.53	0.005	83.48	0.01	80.36	0.003	84.52	0.002	84.97	0.003	84.67	0.002	82.74	0.001	84.97	0.005	87.30	0.001
BayesNet	75.89	0.982	79.56	0.764	75.45	0.869	76.79	0.637	76.39	0.872	75.60	0.638	77.98	0.872	77.18	0.641	77.53	0.861	77.58	0.637
NaiveBayes	72.77	0.068	76.59	0.038	74.85	0.049	73.61	0.036	74.21	0.047	72.32	0.037	74.55	0.049	73.02	0.037	73.51	0.048	74.40	0.035
SMO	93.75	0.332	94.25	0.22	94.20	0.303	95.24	0.246	91.87	0.293	92.26	0.221	93.90	0.295	93.65	0.221	92.71	0.286	93.85	0.216
IBk	87.35	4.966	85.32	3.659	82.89	4.868	84.52	3.718	83.13	4.882	82.59	3.706	84.97	4.793	82.74	3.714	84.23	4.798	86.31	3.612
JRip	79.46	0.003	79.37	0.002	79.17	0.002	78.97	0.001	80.16	0.002	78.42	0.001	79.17	0.002	81.35	0.002	81.99	0.002	78.17	0.002
RandomTree	50.60	0.004	58.13	0.001	49.40	0.003	50.99	0.002	47.62	0.002	49.11	0.002	49.55	0.002	50.40	0.003	54.46	0.002	51.39	0.001
PART	86.01	0.004	84.92	0.002	87.95	0.002	84.92	0.002	86.71	0.002	84.97	0.003	86.76	0.003	85.91	0.002	86.31	0.002	87.50	0.002
REPTree	77.53	0.919	79.37	0.002	73.66	0.808	73.81	0.001	74.60	0.817	74.85	0.001	75.30	0.806	75.40	0.001	73.51	0.777	76.98	0.001
LWL	63.84	345.514	62.30	257.157	63.10	347.975	64.29	261.334	61.31	343.889	64.58	258.943	61.76	341.581	63.10	255.663	63.10	339.778	60.12	254.322

Figure 5.2 compares the results of FLEMs built with the three rules and variable sizes from 3 to 9 on the test data. As can be seen, there is not much difference in overall classification accuracy between the three rules when the ensemble sizes are equal to 5 and more. But when the size of ensembles is small, i.e. 3, R0 did much better than the other two rules, producing the highest accuracy (93%). This indicates that it is very important to choose the classifiers that are the most accurate ones in the model pool, PM, as the core models in an ensemble, which is what R0 does. So, that those best individual classifiers can dominate the performance of the ensemble to produce the overall best classification. When the size of an ensemble increases the three rules appear to produce similar accuracy consistently. However, R2 possesses the largest mean accuracies with smallest standard deviations, which means the ensembles built with R2 are more consistent or reliable, as well as more accurate. So, we conclude that the ensemble with model selection criteria using the CFD combined with DF and accuracy measures (R2), provides a superior result to that of either pair-wise diversity (R1) or accuracy (R0) alone. Diversity and accuracy must both be taken into consideration when constructing large ensembles for classification.

5.3.1.4 Results from HOMOFLEMs built with three rules and variable sizes:

The results for all the homogeneous ensembles HOMOFLEM generated by the ten base learning algorithms using the three rules are shown in Appendix A.

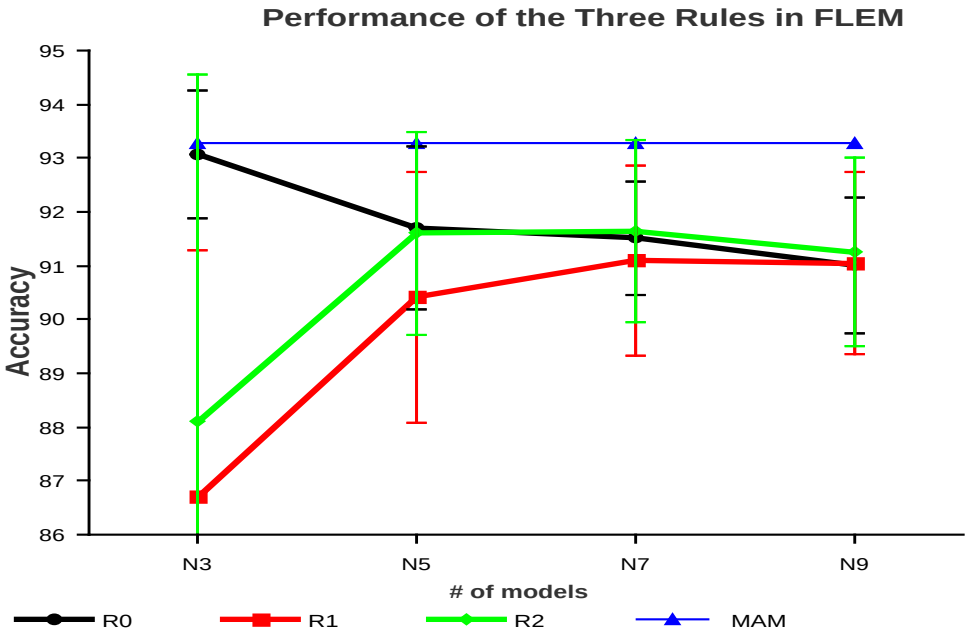

Figure 5.2: Comparison of three rules as the size of FLEM varies.

5.3.1.5 Results using text, images and combined datasets:

As designed, further experiments were conducted by separately using three sets of data: text, imagery and combined, in order to investigate if the aggregation of subsets of multimedia data gives better results. The experiments on the textual dataset D_t alone were conducted with our Heterogeneous Ensemble System, called HEST, and their results were described in Chapter 4 and reported in our paper (Alyahyan et al., 2016). The experiments on the image dataset D_g, called HESG, were conducted in Chapter 4 in the same way as the one used for the text experiments. The results of HESG, obtained in these experiments, are shown in Appendix C Figure C.2.

A further observation from these results is that, using FLEM, the accuracy of the combined text and imagery datasets was lower than that of using the text dataset alone. Furthermore, the accuracy of the image dataset alone was lower

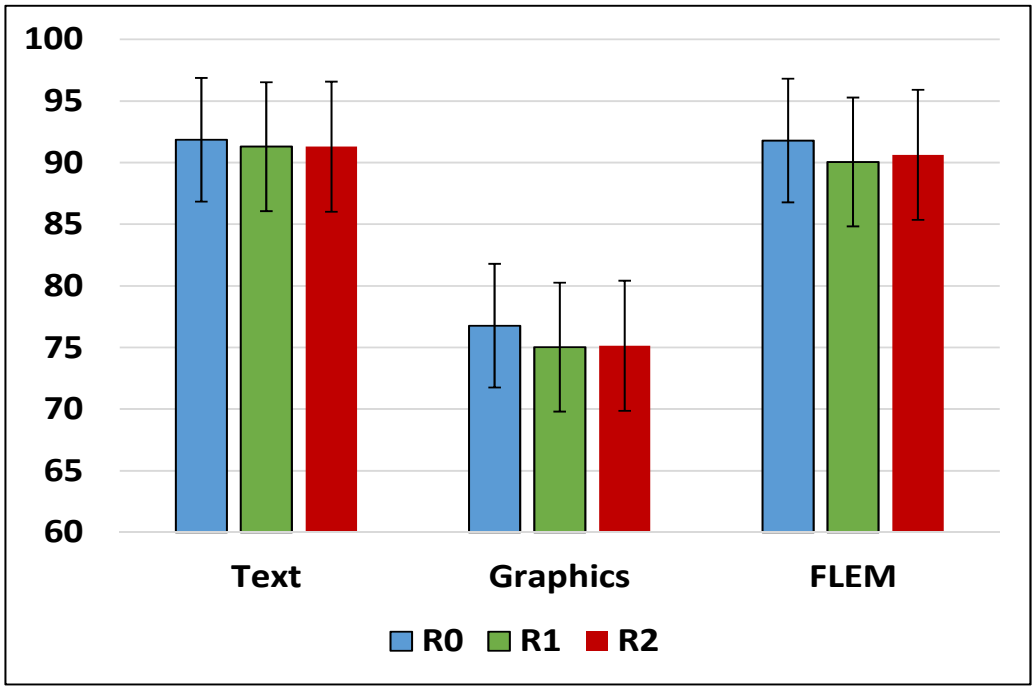

Figure 5.3: Comparison of all the ensembles built with three rules for text dataset, image dataset and the combined multimedia dataset respectively.

than that of the text dataset, as shown in Figure 5.3. A plausible explanation for these differences is that the features extracted from the imagery dataset did not represent the information associated with the underlying classification knowledge of the problem very well, or even worse brought in some noise, and hence confused the learning algorithms resulting in quite weak or bad models, which in turn resulted in weak ensembles. On the other hand, the text data, or more precisely, the features extracted from the text data, are more representative or salient as the ensembles built with the models trained with the text data are more accurate, about 15% higher than those of ensembles built with the image data.

With the combined dataset, the ensembles produced good results, which are comparable to those obtained by the text data only. But on the whole, in this application, the aggregation of multimedia datasets did not offer much additional

benefit in terms of improving classification accuracy.

5.3.1.6 Comparison with homogeneous ensembles:

For the comparison with homogeneous ensembles we compared FLEM with HOMOFLEM and AdaBoostM1 for each base learning algorithm.

A- Comparison between FLEM and AdaBoostM1

As we compared HEST and HESG with AdaBoostM1 in Chapter 4, we did the same thing with FLEM. Table 5.2 shows the results for homogeneous ensemble over all the five runs conducted.

Table 5.2: The accuracy for five runs using AdaBoostM1 method for each base classifier in FLEM.

	Run 1	Run 2	Run 3	Run 4	Run 5	Mean	SD
BayesNet	79.46	80.36	79.91	81.85	81.10	80.80	0.95
NaiveBayes	76.49	75.74	74.40	75.89	74.70	75.19	0.87
SMO	**93.75**	**94.20**	92.26	**93.90**	**92.71**	93.27	0.83
IBk	86.76	82.44	82.44	85.42	82.89	83.30	1.98
Jrip	85.12	89.14	87.80	87.80	88.84	88.39	1.58
PART	91.37	91.37	**93.30**	91.67	91.82	92.04	0.81
J48	90.63	89.58	90.33	90.63	90.33	90.22	0.43
RandomTree	50.74	51.34	52.83	48.36	52.53	51.26	1.78
REPTree	87.20	84.97	86.46	84.97	86.61	85.75	1.02
LWL	82.89	82.14	81.25	80.36	84.08	82.14	1.44
Mean	82.44	82.13	82.10	82.08	82.56	82.26	
SD	12.35	12.18	11.88	13.06	11.89	12.27	

B- Comparison between FLEM and HOMOFLEEM

Another set of experiments was conducted to compare the performance between FLEM and various HOMOFLEM, built using all three model selection rules that have been implemented in FLEMs. To generate the homogeneous models for FLEM for each algorithm we manipulated the dataset by randomly reordering

them and performing the partitioning 10 times in order to produce 10 models in the PM.

The comparison of all HOMOFLEM results for each base learning algorithm used in FLEM is given in Figure 5.4. Each statistical difference diagram compares the results for five different runs. The comparison includes: single model, AdaBoostM1, FLEM results; and HOMOFLEM results using R0, R1 and R2.

In one case (SMO) heterogenous FLEMs were worse performing than homogenous FLEMs. This suggests that the type of base learner is significant in this approach, and that SMO is not a suitable base learner to give good performance with heterogenous FLEMs.

Heterogeneous FLEMs gave better performance than HOMOFLEMs, single models or AdaBoostM1, in most cases. Overall heterogeneous FLEM-R0 was best performing of all the methods, having the best accuracy in 8 out of the 10 cases, and second best in 1, however, these results were not statistically significant in many cases, as can be seen from the critical difference diagrams.

In one case (SMO) heterogeneous FLEMs were worse performing than homogeneous FLEMs. This suggests that the type of base learner is significant in this approach, and that combining other weak base learners with SMO is not suitable to give good performance with heterogeneous FLEMs.

C- Comparison between FLEM and the best of AdaBoostM1

The statistical test was carried out over the five different runs. For R0, R1 and R2 we counted the mean for the different four sizes which are 3, 5, 7 and 9. For the best AdaBoostM1 we chose the highest ensemble each run, no matter which

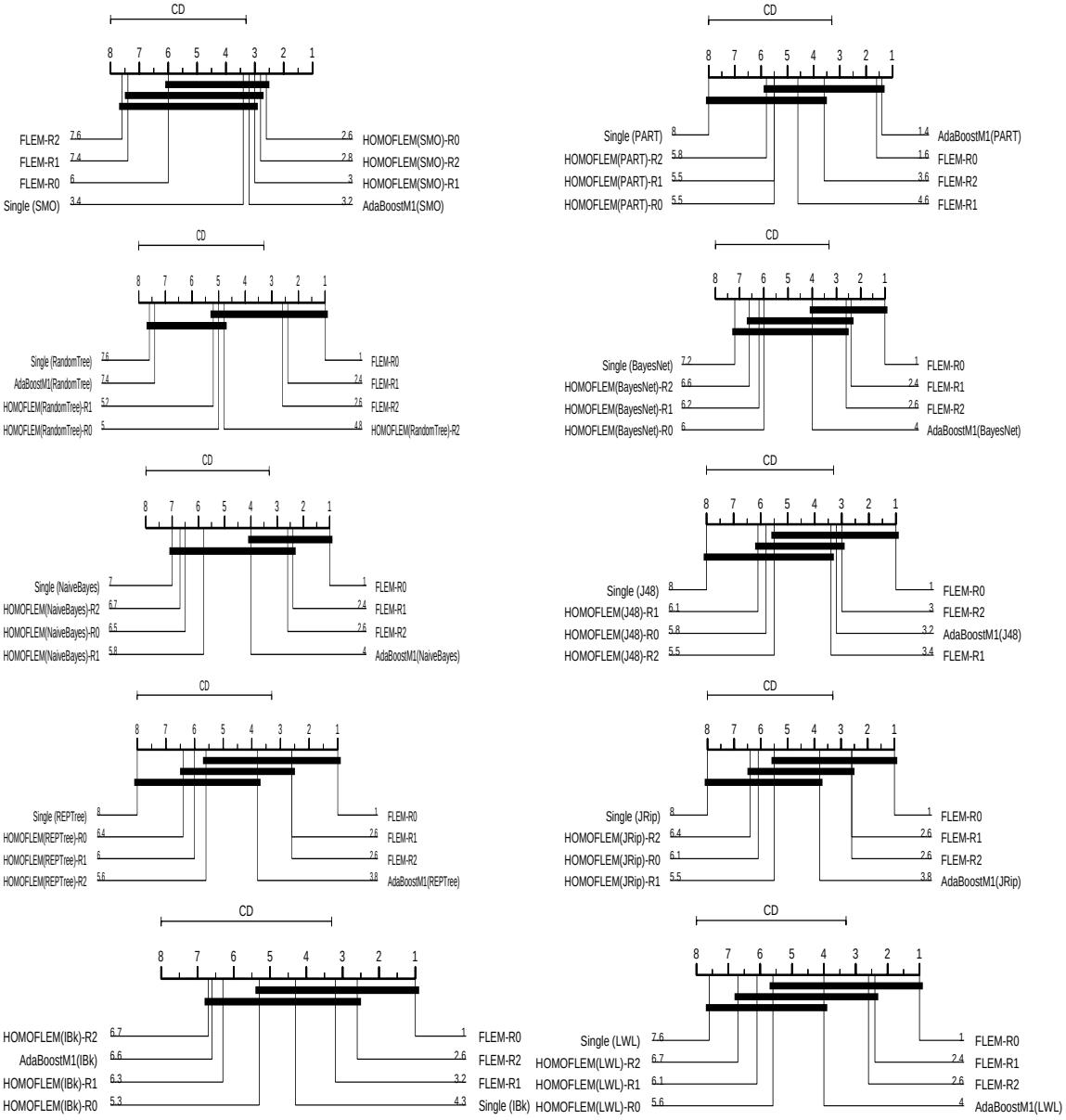

Figure 5.4: The comparison of results for homogeneous ensembles generated with each base learning algorithm used in FLEM. Each sub-figure compares the results for five runs. The comparison includes: single model, AdaBoostM1 and FLEM results; and homogeneous ensemble results using R0, R1 and R2.

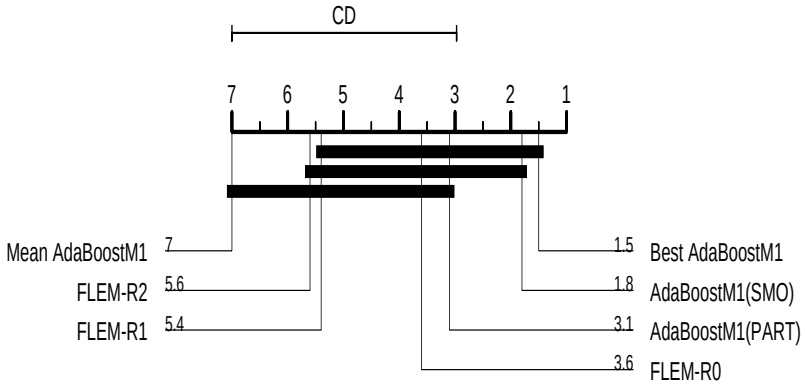

Figure 5.5: The critical difference diagram shows the differences between the results obtained by R0, R1, R2, Mean of AdaBoostM1, Best AdaBoostM1, AdaBoostM1(SMO) and AdaBoostM1(PART).

base learning algorithm was used. The highest two AdaBoostM1 were obtained using SMO and PART, so we included them on the statical test.

Figure 5.5 shows the critical difference diagram for the purpose of comparison between FLEM and the best of AdaBoostM1. It shows that the best AdaBoostM1, AdaBoostM1(SMO) and AdaBoostM1(PART) are ranked higher than our results in FLEM–R0, FLEM–R1 and FLEM–R2. A possible explanation for the weakness of our methods is that combining multi-media data at the feature level affects some base classifiers negatively due to the high dimensionality of the combined data. This effect will extend to the heterogeneous ensemble created by these classifiers. SMO classifier is performing well on both sides of the data D_t and D_g so that it performs well on the homogeneous ensemble. Furthermore, when we tested SMO with HOMOFLEM, we obtained good results as shown in the top left diagram of Figure 5.4.

5.4 Summary

Aggregating and mining multi-media datasets effectively is a challenge task in machine learning and data mining fields. In this work, we developed a feature-level ensemble method (FLEM) with an aim of achieving better classification of multimedia data. Our FLEM consists of four stages: extracting features from multimedia subsets and aggregating them into a single dataset, modelling the combined dataset, selecting models with different rules based on various criteria, and building heterogeneous ensembles. The experimental results have demonstrated that our FLEM is capable of handling multimedia datasets—unstructured text data and imagery data, simultaneously and builds the best ensembles with appropriate datasets, with either combined multi-media data or single-media data. In general, the heterogeneous ensembles are much better than homogeneous ensembles in terms of accuracy and consistency.

Another point drawn from these results is that it is necessary to be cautious when combining multiple data subsets because the aggregated data may not produce a better result than that of using data subsets of single media. Possible reasons include poor features extracted from each subset, which capture more noise rather than useful information; and/or inappropriate aggregation, which may introduce some inconsistency or even contradictions into the final dataset and therefore cause a great deal of difficulty and/or confusion in learning.

In this chapter we have investigated heterogeneous ensemble systems that combine each individual type of medium at the feature level prior to data aggregation and perform ensemble generation using the combined dataset. In Chapter 6

we will extend the work presented in Chapter 4 and in this chapter by performing modelling on each data type separately and incorporating a new general model selection rule that enables the use of multiple critera for selecting each model.

Chapter 6

Generalised Decision Level Ensemble Method

In Chapter 5 we investigated ensemble systems that combine all the individual types of media at the feature level prior to data aggregation and ensemble generation using the combined dataset, we called that FLEM. In this chapter we will investigate ensemble systems where features are extracted from data for individual types of medium. Data for each type of medium are then used separately to generate models. These models will then be combined into an ensemble. This system will be called Generalised Decision Level Ensemble Method (GDLEM), because it will extend the work presented in Chapters 4 and 5 by performing modelling on each data type separately and incorporating a new general model selection rule that enables the use of multiple critera for selecting each model. This research work and results have been published in Wireless Networks Journal (Alyahyan and Wang, 2018a) and the Thirty-eighth SGAI International Conference on Artificial Intelligence (Alyahyan and Wang, 2018b).

6.1 Introduction

In this chapter we conduct Generalised Decision Level Ensemble Method (GDLEM) experiments. We first use the heterogeneous ensemble method to model two different multimedia datasets and then aggregate the outcome from the selected models. The same three rules (R0, R1 and R2, described in Chapters 4 and 5,) that were implemented for model selection were applied as in earlier experiments but one extra, generalised rule was added, which we call R3, which has the ability to combine accuracy and diversity to select a single model.

In the first stage, the DLEM extracts features from each subset of media data to create a series of datasets, D_is, for $1 < i < n$ such that each D_i represents the unique type of media features, i, for each instance.

In the second stage, the DLEM employs some heterogeneous machine learning algorithms to generate individual models for each dataset D_i. The total number of the generated individual models for the MMD, D, is determined by $m*n$. This modelling stage produces a pool of models, PM, with members PM_{ij} representing the individual model fitted using D_i with the base classifier method B_j, for $1 < j < m$.

6.2 The GDLEM

6.2.1 The Generalised Decision Level Ensemble Method Framework

Our Generalised Decision-Level Ensemble Method (GDLEM), as shown in Fig. 6.1, consists of four modules, (1) the multimedia data representation and feature extraction module, (2) the modelling module, (3) the model selection module, and, (4) the combination module using *majority voting*.

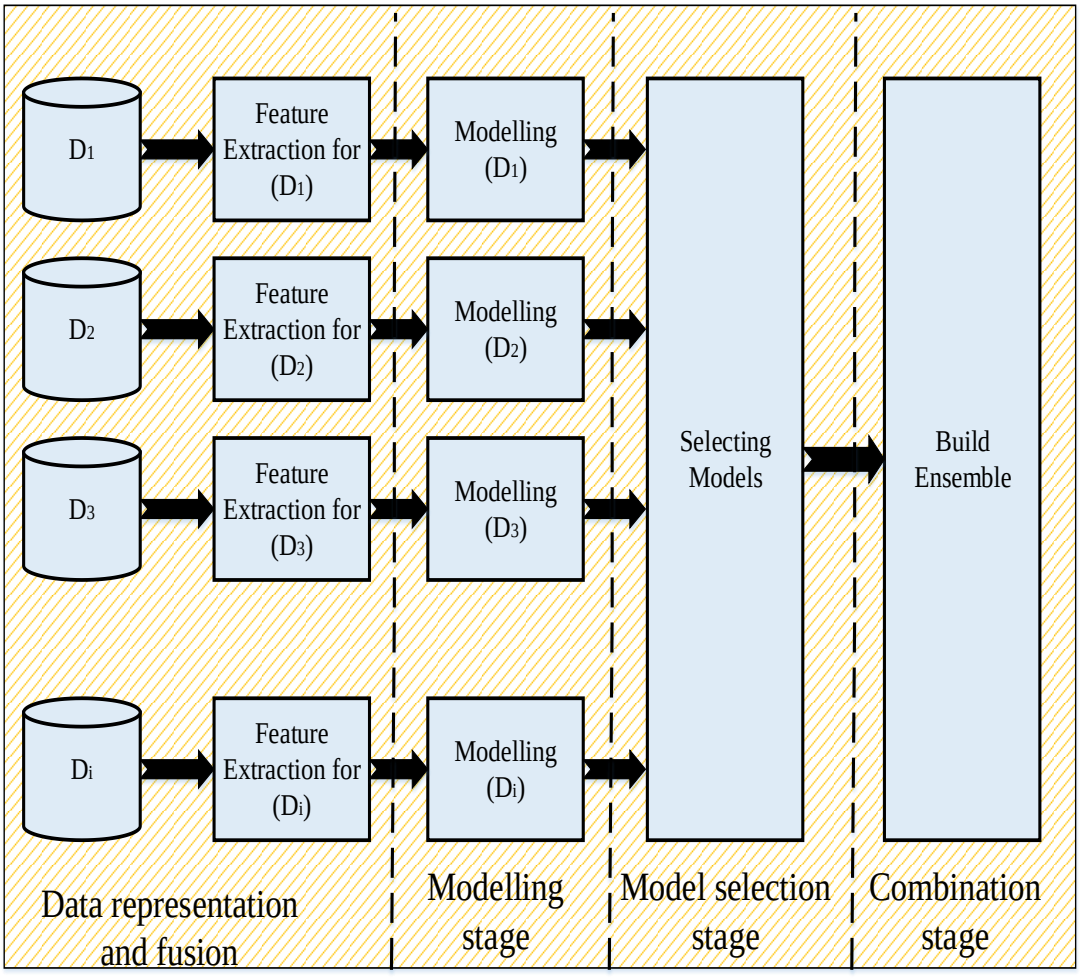

Figure 6.1: The general framework for DLEM

The third stage selects models from the model pool PM using accuracy and diversity as selection criteria, either individually or jointly in some predefined

rules. Using these criteria, three different rules: $R0$, $R1$, $R2$, were derived. After some intensive experiments in our earlier studies, we then devised a new rule that uses a function to combine accuracy and diversity in a more generalised manner to select the models.

Rules R0, R1 and R2 were described in Chapters 4 and 5.

The new generalized rule, R3, is described below.

R3: This new rule uses a combination of accuracy (Acc) and diversity(Div) as a generalised criterion for selecting models to build an ensemble. The combined measure is defined in Equation 6.2.1:

$$\gamma_i = \alpha(Acc)_i + \beta(Div)_i \ . \tag{6.2.1}$$

Here α and β are the weights for accuracy, Acc, and diversity, Div, respectively, of model m_i ($1 \leq \mathrm{i} \leq n-1$) in the PM. The type of diversity measure Div in this rule is flexible and can be pairwise or non-pairwise as long as it is considered appropriate. In this study, we use the CFD.

After taking the best model out from the model pool PM, the combined score, γ_i, is calculated for the remaining $n-2$ models in PM. The model with $max(\gamma_i)$ is selected from PM and added to Φ.

R3 is considered as a generalised rule because all other three rules R0, R1 and R2 are just its special cases with specific values for the weights and the diversity measure. When set $\alpha = 1$ and $\beta = 0$, R3 becomes R0. If we use the DF as the diversity measure and set $\alpha = 0$ and $\beta = 1$, then R3 becomes R1. If we a use non-pairwise diversity measure such as the CFD and set $\alpha = 0$ and $\beta = 1$, R3

becomes R2.

Based on this new rule, a corresponding algorithm for building a decision level ensemble was derived and named as Generalised Decision-level Ensemble Method (GDLEM) because it is flexible, employing R3 to apply various rules for selecting models by manipulating the weights or changing the measures used in the relationship γ in equation 6.2.1. The GDELM is as follows. The first step is the same as that of the other three rules, i.e. choosing the MAM from PM as the first member of Φ. The key difference starts from the second step where the selection of candidate models uses the newly defined γ_i. This second step is repeated until N models with $max(\gamma_i)$ completely fill Φ.

6.2.2 Implementation of the GDLEM

The experiment was carried out on a standard PC, with an Intel I7 processor and 16 GB RAM. As the GDLEM is flexible for selecting candidate classifiers, we have selected 10 efferent base classifiers that are provided in the WEKA library (Witten et al., 2016). These base classifiers are: trees (*J48, RandomTree, REPTree*), bayes (*NaiveBayes, BayesNet*), function (*SMO*), rules (*JRip, PART*) and Lazy (*IBk, LWL*).

6.3 Experiment Design and Results

6.3.1 Experiment Design and Results

We carried out a series of experiments to investigate the performance of the GDLEM, using three selection rules separately, on the multimedia data. The issues investigated included (1) the performance measures and classifier selection

criteria represented by the rules: R0, R1, R2 and R3, and (2) the ensemble size. A total of 135 experiments were conducted. This involved running all possible combination of these parameters. Each experiment was repeated five times with different samplings of the datasets.

In parallel, we conducted experiments to investigate the influence of CFD values on the accuracy of all the ensembles built with the first three rules, although the CFD is not used by R0 and R1.

With R3, through varying the values of the weights α and β from 0 to 1 with an increment of 0.1, such that $\alpha+\beta = 1$, and using the above experiment settings, 850 experiments were carried out in total.

6.3.1.1 Results of R0, R1 and R2

Some results are summarised in Figures **??**–**??**. They clearly shows that the DLEMs built with the three rules are generally superior to individual classifiers, because the mean accuracies (shown in red lines on the figures) of the DLEMs are approximately 10% higher than the mean accuracies (illustrated by blue lines) of the individual classifiers in the DLEMs. In addition, it was further demonstrated that our ensemble results have a higher level of accuracy overall than the best individual models, the MAMs. Hence, our DLEM had the best reliability overall because the reliability of an MAM was not consistent over a succession of experiments, while the ensembles built with our method, the DLEM, are more consistent and reliable as well as more accurate. Moreover, the selected models and its diversity CFD for each experiment are shown in Appendix B, Tables B.13 – B.21.

Figure 6.2 compares the results of GDLEMs built with the three rules and variable sizes of odd numbers from 3–19 on the test data. This shows the weakness of R1. Our previous studies indicated that there were accuracy issues with this rule. However, these became much more apparent in the current work when the high numbers of models were used. The increase in model numbers highlighted very clearly the disadvantages of R1. As can be seen, its accuracy levels varied inconsistently, starting low and going lower. It only improved when N = 11. All the way up to N = 19, it is still worse than two other rules.

R0 performed reasonably well because it combines the models which have the best accuracies in the PM. R2 is same as R0 when N=3, but improved while R0 went down when the size increased, although they are similar after N = 11. But R2 is more favourable as it performed better when the size of ensembles was smaller, which means it is more efficient.

Figure 6.3 shows the average values of the CFD in the ensembles built with R0, R1 and R2, although the CFD is not used in R0 and R1. The purpose is to see if the CFD can be used to explain why some ensembles are better than others. These results show that in R0 the CFD is increasing to give the best results at N=11. When we link this result with the accuracy level for R0 shown in Figures **??**–**??**, we can see that the best ensemble results were gained when we combined models that have the best accuracy and CFD when N = 11 and 19.

6.3.1.2 Results of R3

As R3 is a generalised and flexible rule, it enables us to do more investigation into the influences of the CFD in the ensemble. Figures 6.4–6.7 show improvement

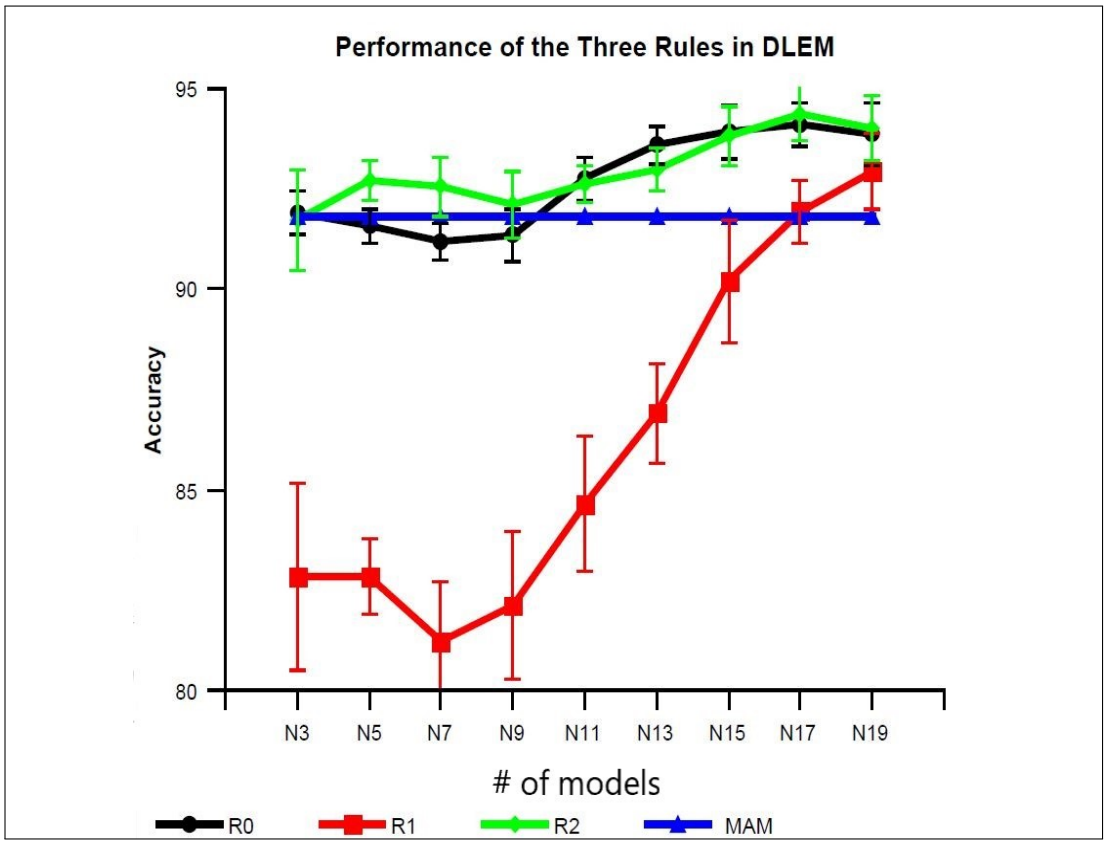

Figure 6.2: Comparing the results produced by all three rules in nine different sizes of the GDLEM.

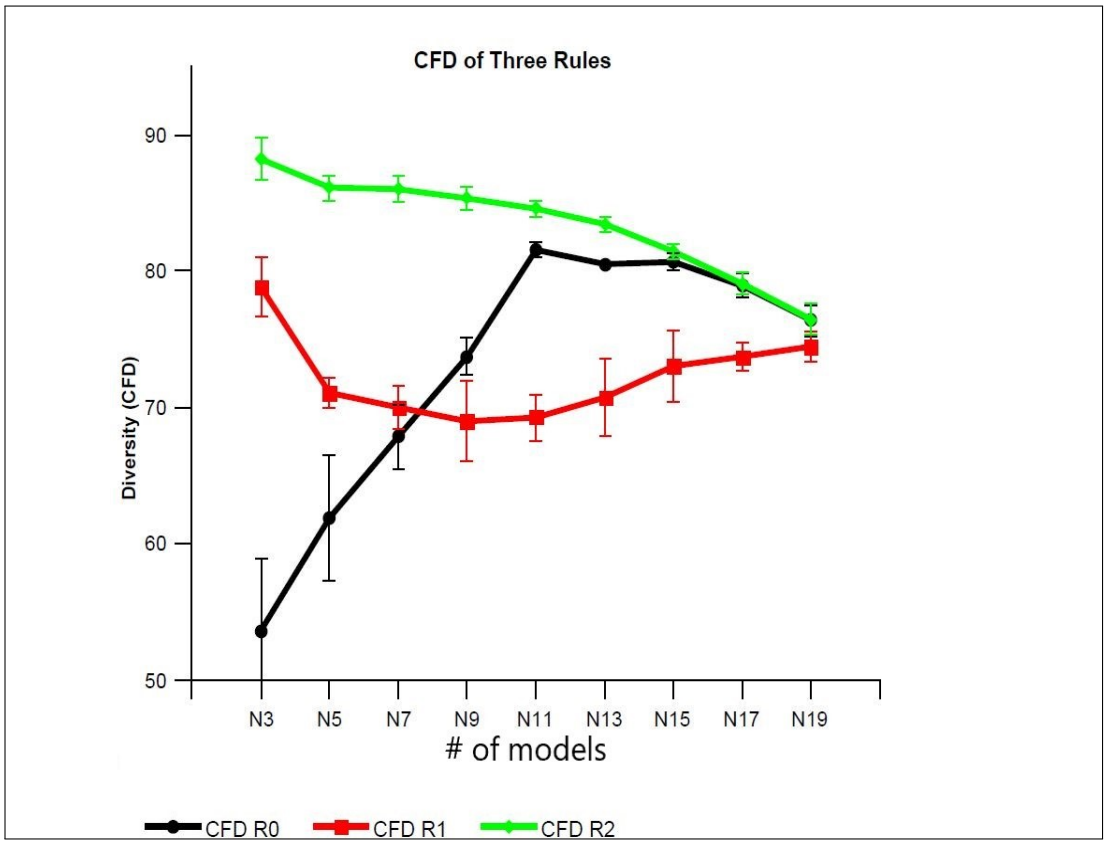

Figure 6.3: Comparing the CFDs for all three rules in nine different sizes of the ensembles.

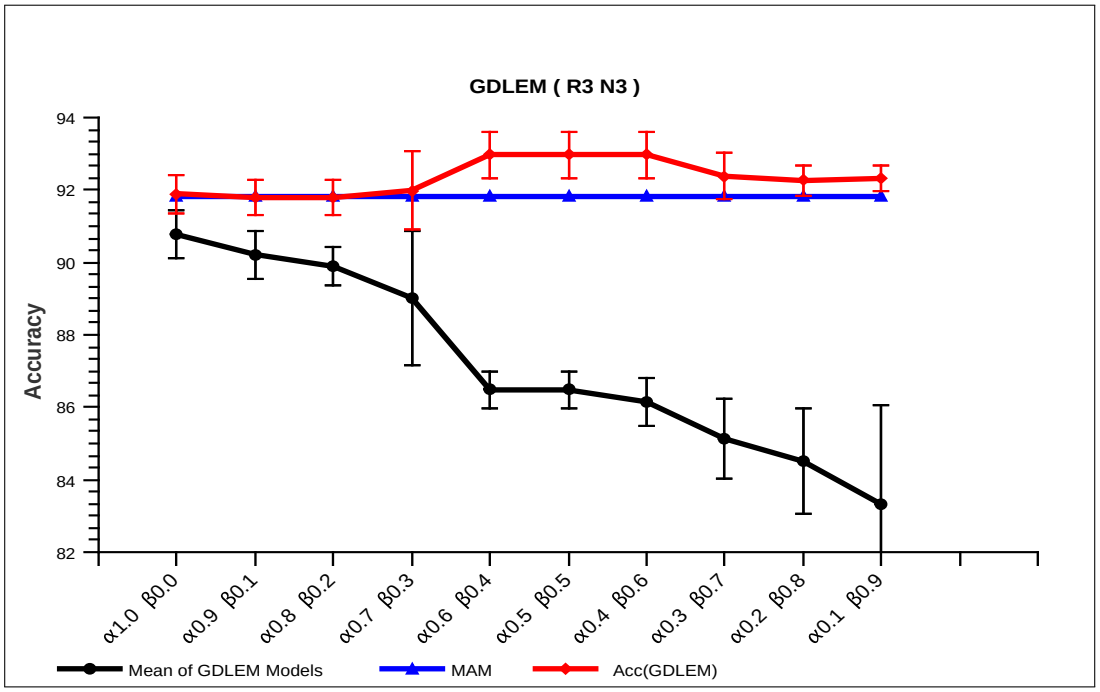

Figure 6.4: Sample of GDLEM results for the generalised rule R3 with ensemble size 3. Three lines: The red line represents the accuracy of GDLEM, black the mean accuracy for models that are chosen for the GDLEM and blue for the MAM.

in accuracy for model selection for some ensembles of size 3 to 9, given α is set between 0.5 and 0.6. For ensembles of size greater than 9, varying α and β does not have much impact on the accuracy level for the ensemble (see Figures 6.8–6.11) and that is because the size of the model pool is too small. When the size of the ensemble reaches and exceeds 50% of the model pool, there is not much space for selecting models and hence the ensembles could be more or less the same regardless of whatsoever models are chosen.

The best results are produced by R3 when the weight of the accuracy α is equal to 0.4 as it is shown in the critical difference diagram in Figure 6.12 and the weight for diversity is 0.6. This means that when more weight, about 20%, is put on the diversity than on the accuracy, the ensembles with less accurate but more diverse modules achieved the best results. Moreover, the diagram shows

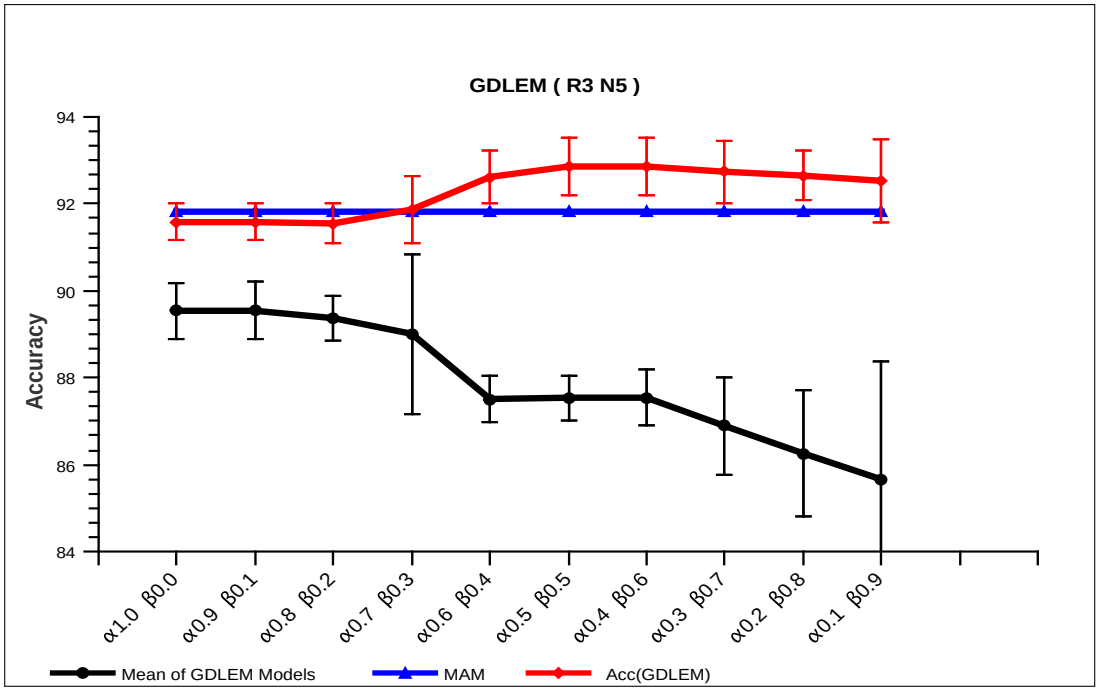

Figure 6.5: Sample of GDLEM results for the generalised rule R3 with ensemble size 5. Three lines: The red line represents the accuracy of GDLEM, black the mean accuracy for models that are chosen for the GDLEM and blue for the MAM.

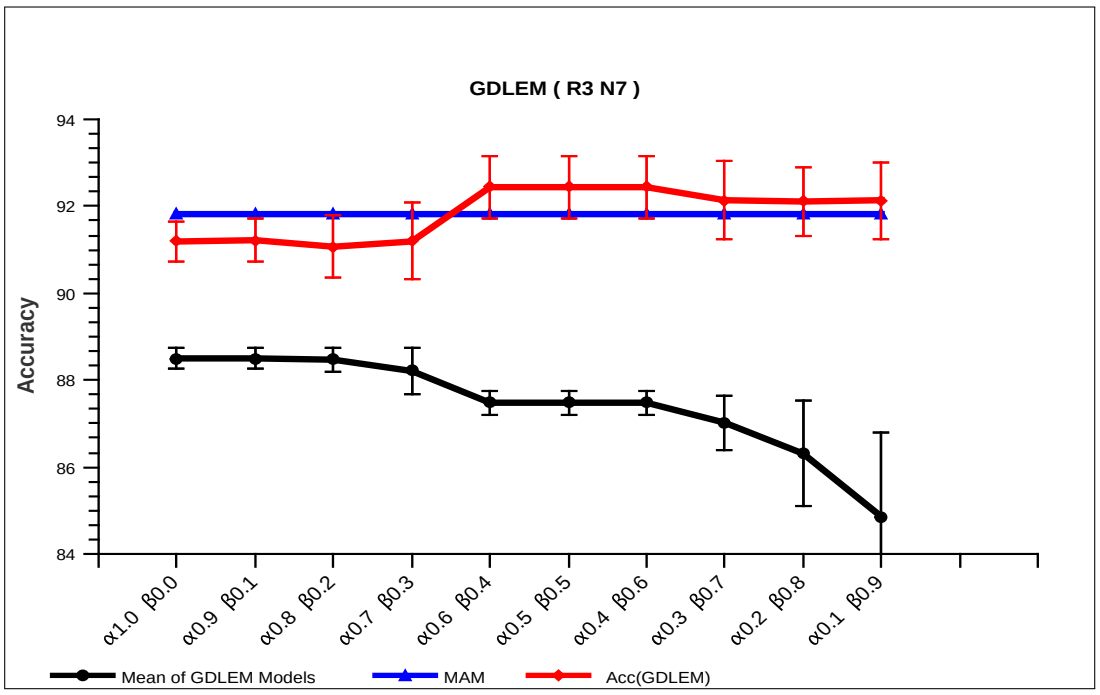

Figure 6.6: Sample of GDLEM results for the generalised rule R3 with ensemble size 7. Three lines: The red line represents the accuracy of GDLEM, black the mean accuracy for models that are chosen for the GDLEM and blue for the MAM.

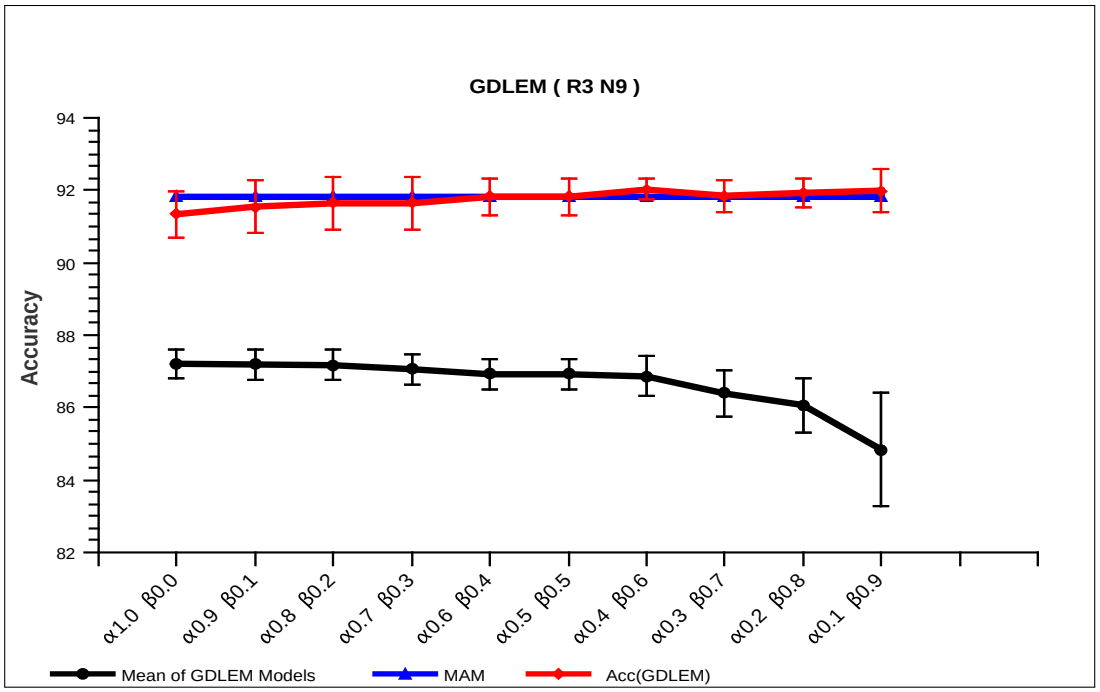

Figure 6.7: Sample of GDLEM results for the generalised rule R3 with ensemble size 9. Three lines: The red line represents the accuracy of GDLEM, black the mean accuracy for models that are chosen for the GDLEM and blue for the MAM.

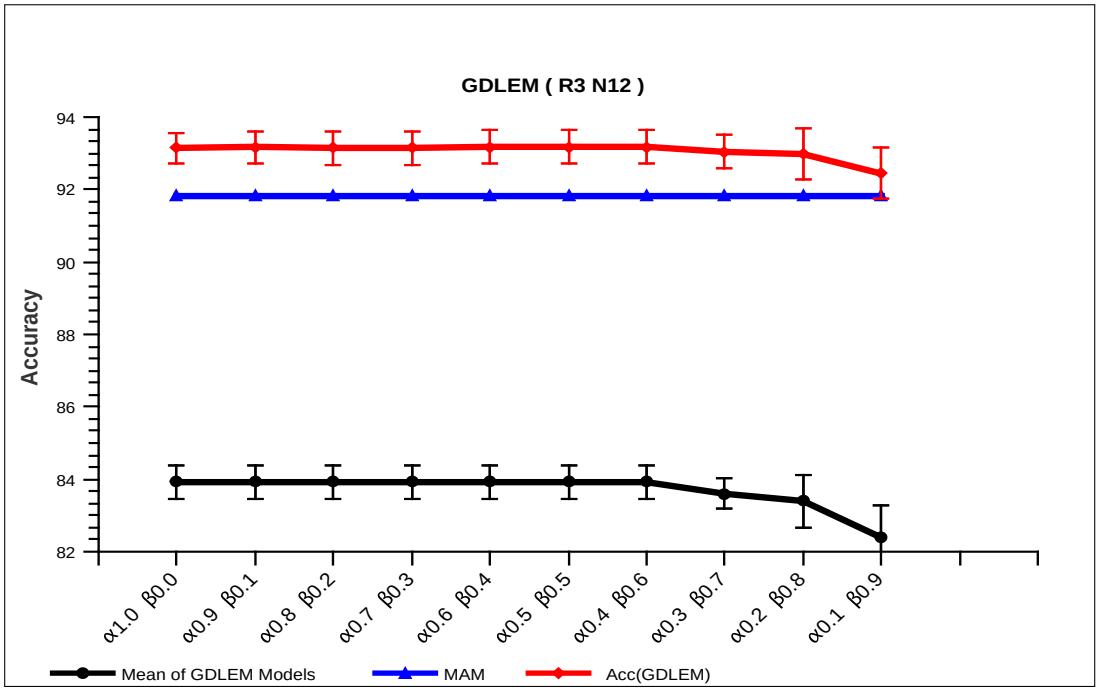

Figure 6.8: Sample of GDLEM results for the generalised rule R3 with ensemble size 12. Three lines: The red line represents the accuracy of GDLEM, black the mean accuracy for models that are chosen for the GDLEM and blue for the MAM.

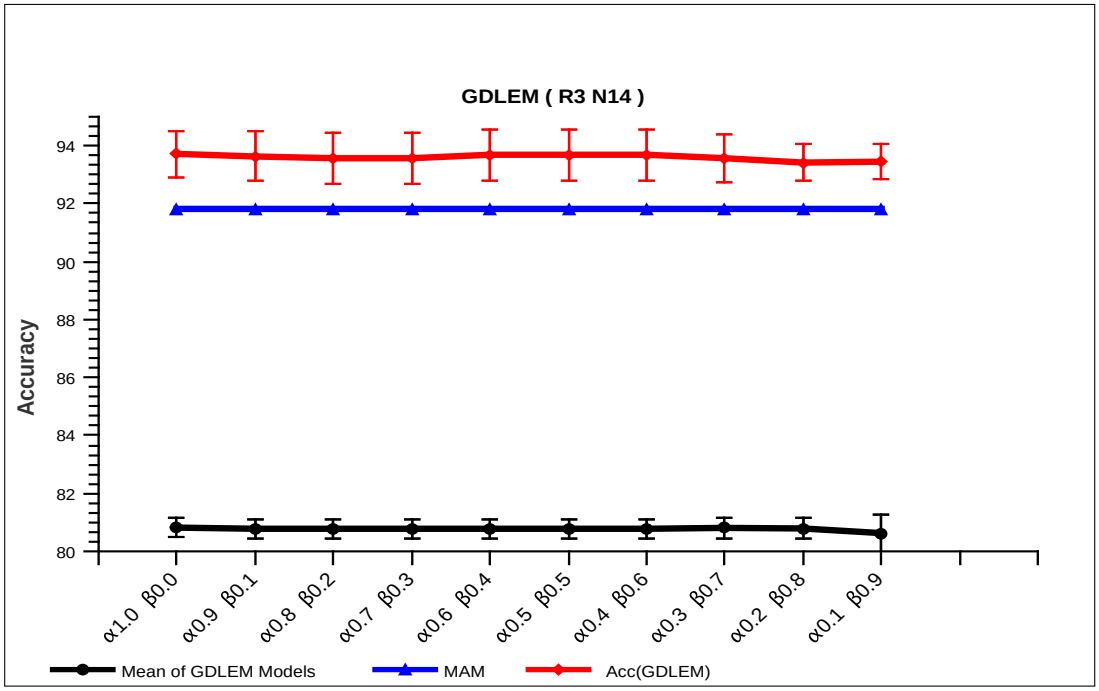

Figure 6.9: Sample of GDLEM results for the generalised rule R3 with ensemble size 14. Three lines: The red line represents the accuracy of GDLEM, black the mean accuracy for models that are chosen for the GDLEM and blue for the MAM.

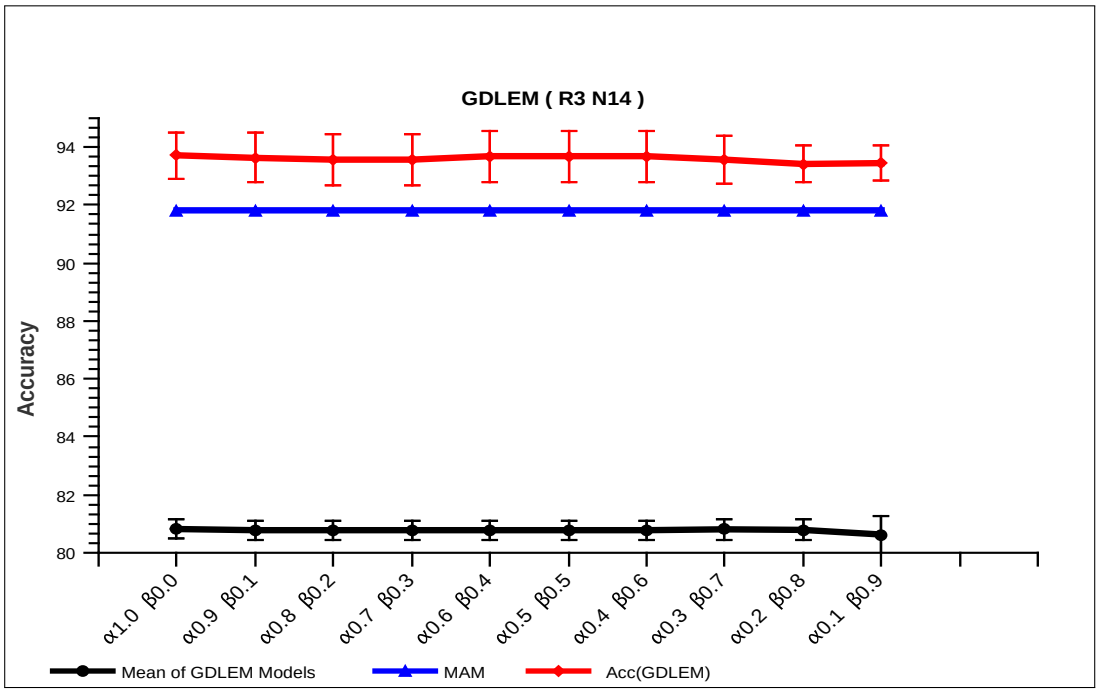

Figure 6.10: Sample of GDLEM results for the generalised rule R3 with ensemble size 16. Three lines: The red line represents the accuracy of GDLEM, black the mean accuracy for models that are chosen for the GDLEM and blue for the MAM.

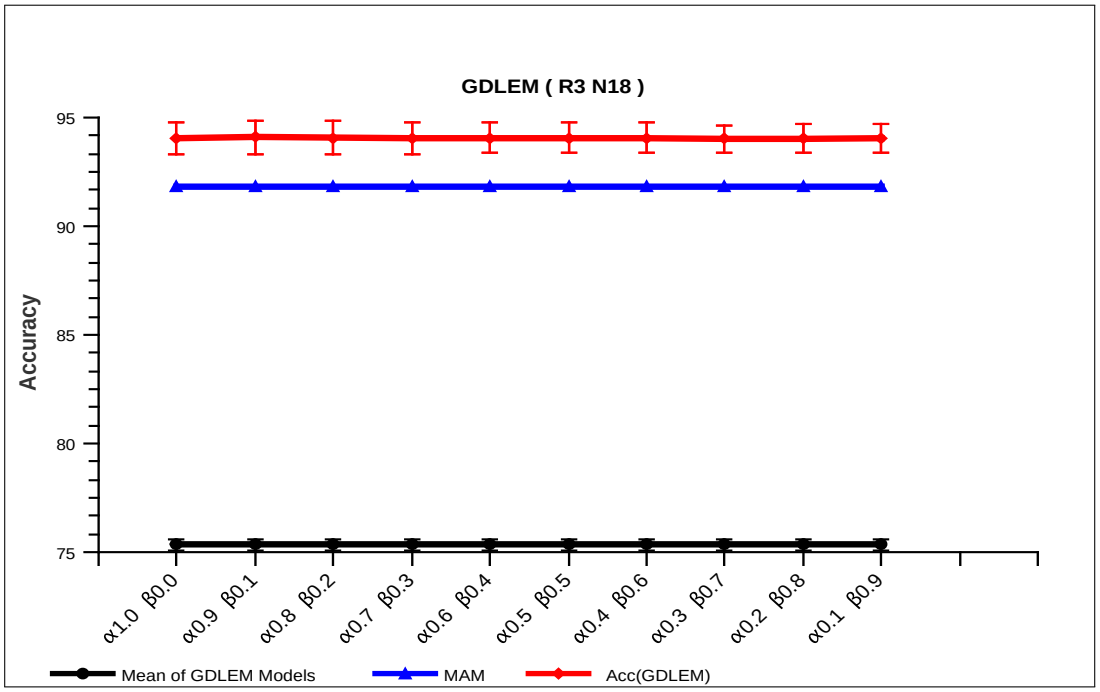

Figure 6.11: Sample of GDLEM results for the generalised rule R3 with ensemble size 18. Three lines: The red line represents the accuracy of GDLEM, black the mean accuracy for models that are chosen for the GDLEM and blue for the MAM.

that the range between 0.4 and 0.6 for α performs better than others.

Thus, it can be seen that the generalised selection rule R3 is a combination of accuracy and CFD measures, gives chances to the GDLEM to select the models that can help improve the accuracy of heterogeneous ensembles. The systematic empirical investigations found that the best ensembles are produced when the weights for accuracy and diversity are split at 0.4 to 0.6 respectively. That effect is clearer when there is a large pool of models and we select less than half the number of models. In summary, the ensembles built with model selection criteria that use a combination of CFD, DF diversity, and accuracy measures, give good results. They are superior to those results obtained using either pair-wise diversity (R1) or just accuracy (R0).

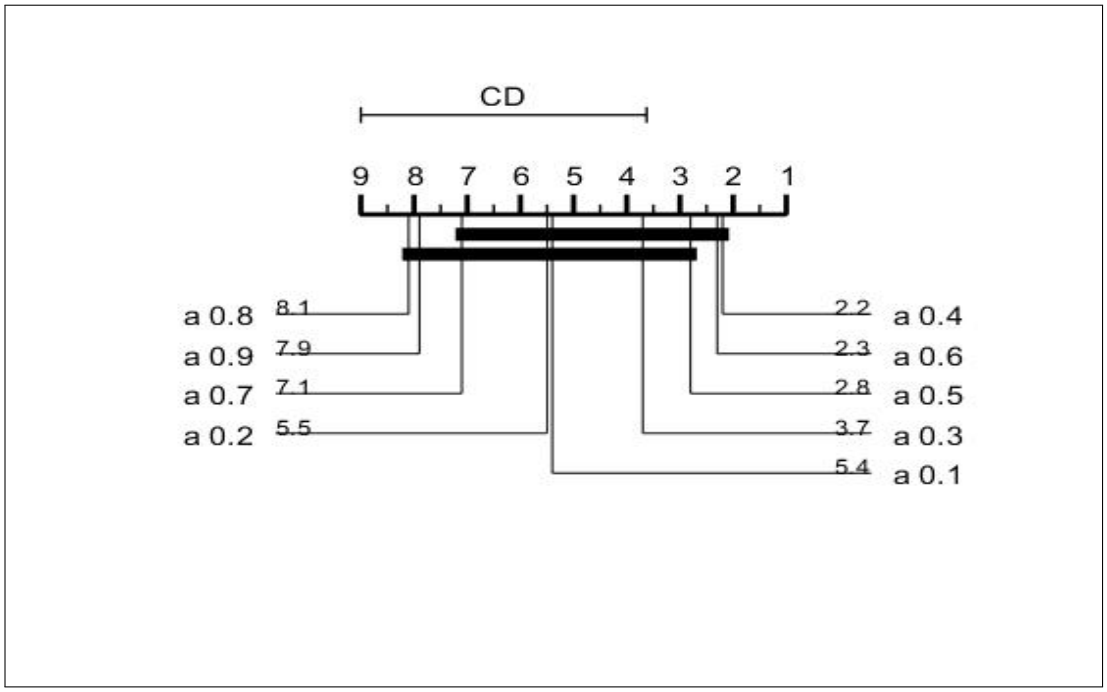

Figure 6.12: Diagram showing critical differences of the average results of ensembles with different sizes from 3 to 19, when the accuracy weight α varied from 0.1 to 1.0 with a step size of 0.1.

6.3.2 Critical Comparison With Other Ensembles

The results of the GDLEM were compared with the FLEM and various heterogeneous ensembles based on the single media data, text (HEST) and image data (HESG). The full comparative results between the FLEM and the HESG were published in (Alyahyan and Wang, 2017) and the full results for the HEST were published in (Alyahyan et al., 2016). Figure 6.13 shows the critical difference diagram for the GDLEM, DLEM, FLEM, HEST and HESG, with all rules R0, R1, R2 and R3. The GDLEM-R3 is the best on average and a credible explanation is that R3 with appropriate weights can produce the optimal combination of model accuracy and CFD to improve ensemble accuracy.

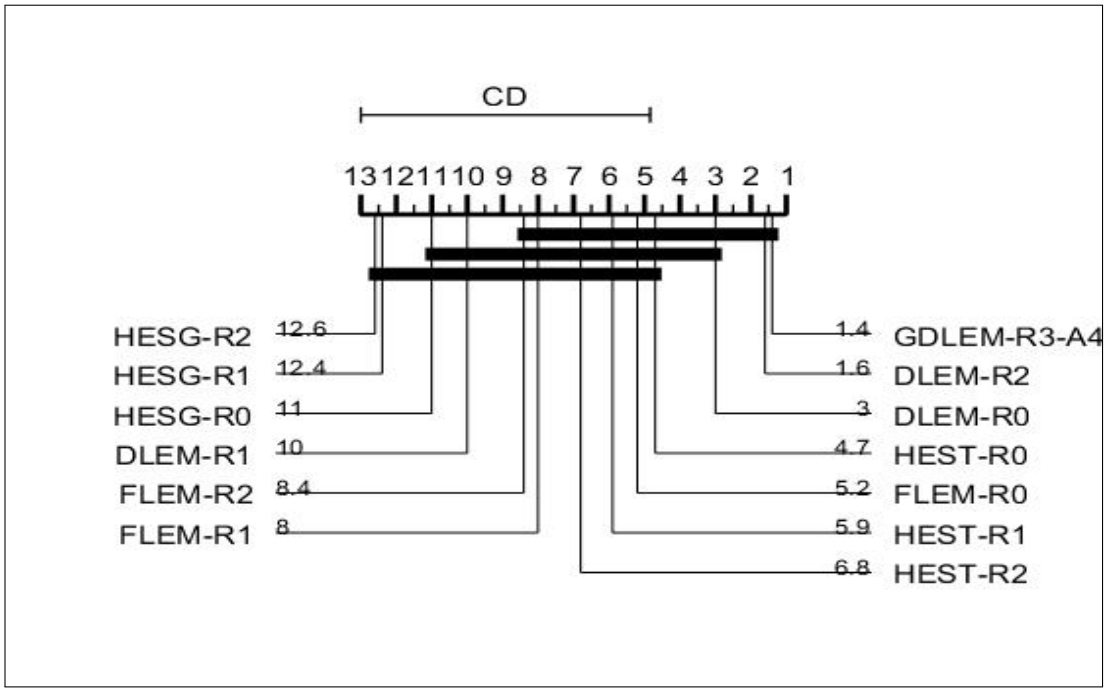

Figure 6.13: Critical difference diagram for the ensembles built with GDELM, DLEM, Feature-Level Ensemble Method(FLEM), Hybrid Ensembles Built with Textual Data(HEST) and with Imagery Data(HESG) for all rules R0, R1, R2 and R3. It shows that the GDLEM with R3 is the best.

6.4 Summary

In this chapter, we developed a generalised heterogeneous ensemble method, GDLEM, to classify multi-media datasets at the decision level, with the aim of achieving the best and most consistently accurate results. Our GDLEM consists of four stages: extracting features from multi-media subsets, modelling the subsets datasets, selecting models with different rules based on various criteria, and building heterogeneous ensembles. The new model selection rule, R3, was demonstrated to have a capability to select the individual models that are less accurate but more diverse. Hence, in some situations, e.g. accuracy weight from 40% to 60%, it achieved the best level of ensemble accuracy, beating those obtained by other ensembles, including DLEM, FLEM, HEST and HESG, using

the same dataset. Another obvious observation from this study is that heterogeneous ensembles give better results when we combine accuracy and diversity measurements for model selection.

Chapter 7

Model Comparison and Evaluation

7.1 Introduction

Having presented our work on HES, FLEM and GDLEM in Chapters 4–6 we present in this chapter an overall discussion of all the methods developed in our research and an evaluation of them.

7.2 Overview of the Research

This research investigated the problem of classifying multi-media data in regard to two main aspects. The first aspect was the construction of heterogeneous ensemble classification methods. The second aspect was applying these methods to multimedia data.

We considered the heterogeneous ensemble for classification problems because it gives the opportunity to analyse the data using different base learning classification algorithms. There are some fundamental issues for selecting models to construct a heterogeneous ensemble, which we addressed in our methods. These issues include: (1) the accuracy of the individual model, (2) the diversity among

the models and (3) the number of models. Moreover, instead of using one criterion to select a single model as in rules (R0, R1 and R2), we apply more than one criterion for selecting one model and this is implemented in rule R3.

A multimedia dataset that has more than one type of media, as defined in Chapter 1, can be dealt with it in two different ways. The first way, is dealing with a single type of media from this data as implemented in HES. The second way, is combining the different types of media data and that was dealt with in two different methods (1) combining the data at the feature level as it applied in FLEM, and (2) combining the data at the decision level as it is applied in GDLEM.

In addition, this section will evaluate our methods by comparing them with two well known established methods, specifically Random Forest and Deep Learning, and using an additional dataset of multimedia data published by Oramas et al. (2018). This dataset was published after the completion of our method development, and therefore enables an independent comparison of our methods with those of (Oramas et al., 2018).

7.3 The development of the research methodology.

In the development of the frameworks, we started by applying the HES framework (see Fig 4.1). In this framework we implemented a heterogeneous ensemble system that was able to classify a single data subset. The components of this framework are (1) the extraction of features from a multimedia subset which are stored in a dataset D, (2) the generation of models, (3) the selection of models using

different rules based on various criteria, and (4) the construction of heterogeneous ensembles.

Since we combined the data at different levels, we extended the HES framework to be able to deal with the combined data. To do this we developed two more frameworks that can deal with the combined data. The first framework, FLEM, combines the data at the feature level as shown in Fig 5.1. It gives the opportunity to combine more than one type of data and aggregate them in a single dataset that allows us to apply the machine learning algorithms to all the types of data together.

The second framework, GDLEM, combines the data at the decision level, as shown in Fig 6.1. It gives the capability to model each individual data subset independently using all available base leaning algorithms. This gives more models than HES or FLEM.

Regarding the development of the model selection rules, two points should be noted. Firstly, R0 uses accuracy measurement, R1 added the pairwise diversity measurement and R2 added the non-pairwise diversity measurement. Hence, in R1 and R2 the measures are combined at the rule level. On other words, applying a single measurement criterion to select a single model.

Secondly, R3 uses a combination of accuracy (Acc) and diversity (Div) as a generalised criterion for selecting a single model to be added to the ensemble. Hence, in R3 the measures are combined at the model level, which means applying more than one measure to select a single model. Fig 7.1 shows the development of the frameworks and the rules for model selection in our research.

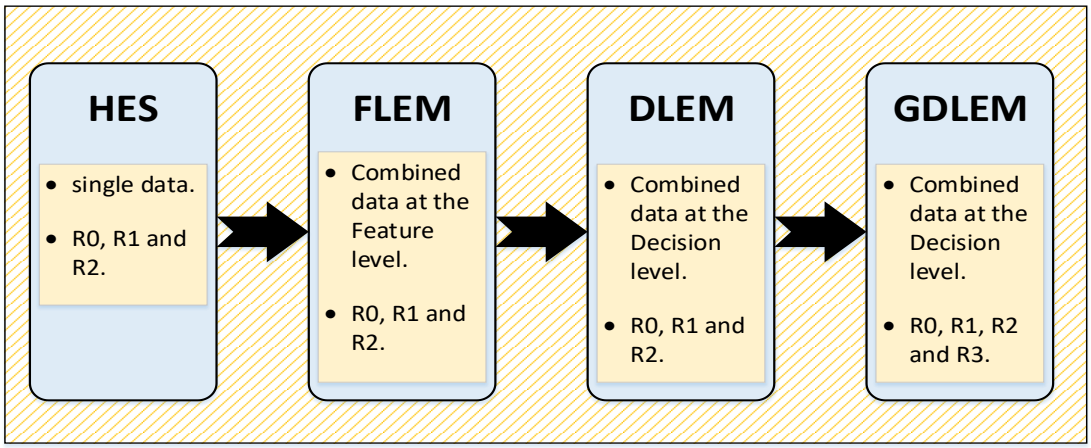

Figure 7.1: The development of the framework

7.4 Examination of the Results

In this section, we will do more investigation of our results and see where our methods make mistakes. The investigation will involve two levels of the results. The first level is the class level and the second level is the instance level.

7.4.1 Class Level

Our aim here is to look more deeply into our results in order to see where our methods are performing well and where are they performing badly. In order to do this, we need to examine the performance of our methods for each target class. Table 7.1 shows the mean accuracies for predicting each class using our methods with each rule. For HEST, HESG and FLEM the ensemble sizes were 3, 5, 7 and 9; and for DLEM the sizes were 3, 5, 7, 9, 11, 13, 15, 17 and 19. Each value given is the mean obtained from five different runs.

From Table 7.1 we can see that the easiest class for our methods was "tall-building" for which the overall mean accuracy obtained was 91.00%. The most difficult class for our methods was "opencountry" for which the overall mean

Table 7.1: The mean accuracies for predicting each class in HEST, HESG, FLEM and DLEM for five different runs.

Method	Rule	cost	forest	highway	insidecity	mountain	opencountry	street	tallbuilding	MEAN	SD
HEST	R0	**95.87**	91.84	90.43	96.01	85.52	87.99	90.81	94.75	91.65	3.77
	R1	95.21	91.61	88.53	95.89	84.46	87.72	90.82	94.04	91.03	3.98
	R2	95.31	**93.40**	87.38	**96.32**	82.08	88.76	90.31	**95.98**	91.19	5.01
HESG	R0	78.25	80.51	76.93	74.92	77.47	65.48	83.38	81.52	77.31	5.50
	R1	76.89	79.31	78.05	73.26	74.17	65.61	83.14	78.75	76.15	5.25
	R2	74.18	73.51	79.92	75.36	76.79	64.33	81.47	80.42	75.75	5.49
FlEM	R0	95.26	89.64	**92.58**	94.69	**90.96**	83.06	**95.36**	95.85	92.17	4.32
	R1	94.73	87.98	89.65	94.61	89.00	79.29	93.02	94.03	90.29	5.19
	R2	94.67	88.04	91.52	94.79	90.62	80.41	93.64	95.00	91.09	4.96
DLEM	R0	95.79	92.92	92.32	96.01	87.14	89.99	93.14	95.52	92.86	3.09
	R1	86.29	83.86	89.67	87.82	87.52	76.61	92.10	90.59	86.81	4.86
	R2	95.53	93.21	92.26	96.18	86.61	**91.00**	93.53	95.57	**92.99**	3.14
	MEAN	89.83	87.15	87.44	89.65	84.36	80.02	90.06	91.00		
	SD	8.52	6.47	5.76	9.42	5.58	10.03	4.70	6.67		

Table 7.2: Confusion matrix summarises all confusion matrices for HEST, HESG, FLEM and DLEM.

		predicted							
		cost	forest	highway	insidecity	mountain	opencountry	street	tallbuilding
actual	cost	**25974**	192	401	86	217	1408	4	68
	forest	5	**23970**	101	7	764	830	131	22
	highway	547	91	**18282**	253	174	719	339	70
	insidecity	71	236	298	**21878**	30	137	668	937
	mountain	424	1008	219	92	**24822**	2578	89	189
	opencountry	1564	1532	768	43	2606	**25566**	67	173
	street	7	44	531	601	134	138	**20741**	799
	tallbuilding	84	328	71	1183	411	315	809	**24834**

accuracy obtained was 80.02%.

Moreover, to see where the methods were confused, we generated a confusion matrix that summarises all 315 confusion matrices from HEST, HESG, FLEM and DLEM. Table 7.2 shows the confusion matrix, and the heat map is shown in Figure 7.2.

From Table 7.2 and its heat-map in Figure 7.2, we can identify the classes where our methods made the most mistakes. Our methods misclassified the "opencountry" class for 19.33% of its instances, most often as "mountain" (8.13% of instances) and "cost" (4.44% of instances).

Another class our methods make mistakes with was "mountain" which was

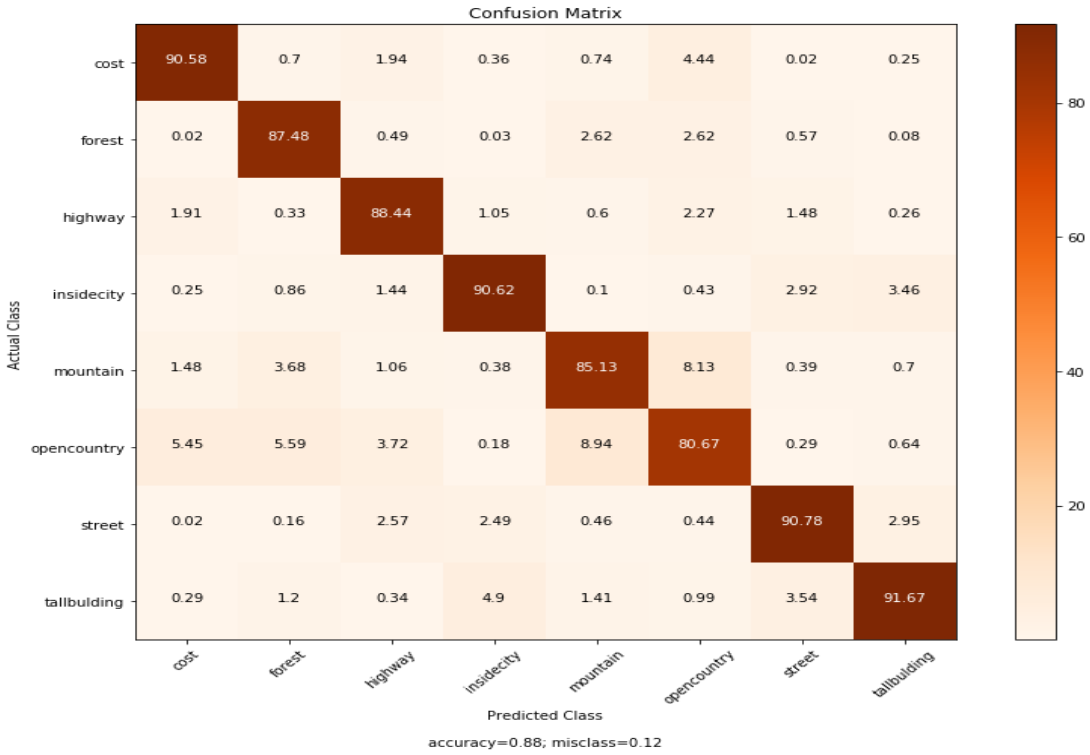

Figure 7.2: Heat-map confusion matrix showing the percentage of the predictions for each class for HEST, HESG, FLEM and DLEM.

misclassified in 8.49% of its instances as "opencountry". Thus, our methods had the greatest difficulty distinguishing between the "opencountry" and "mountain" classes. The most likely explanation for this is the high correlation between them in the features used for prediction.

7.4.2 Instance Level

We analysed all the tested instances in all the five runs in all HEST, HESG, FLEM and DLEM experiments, and we found that the number of misclassified instances for each method were, 412, 148 and 33 respectively. There were only 20 instances that were misclassified in all of our methods. The distribution for them is shown in Table 7.3. There were no misclassified instances in the class insidecity.

Table 7.3: The number of misclassified instances for each class.

Class	# of instances
coast	2
forest	2
highway	2
mountain	1
opencountry	6
street	4
tallbuilding	3

Figure 7.3: Sample of misclassified images for each class. The first row is for coast, forest, highway and mountain; the second row is for opencountry, street and tallbuilding

Figure 7.3 shows samples for misclassified images for each class. The first row is for coast, forest, highway and mountain; the second row is for opencountry, street and tallbuilding. Table 7.4 shows the attributes or (annotations) for each figure in the annotation files.

7.5 Evaluation

In this section, we describe how we evaluated our research methods and how we compared the results of our methods with each other, and with those obtained with existing, well established methods.

Table 7.4: The annotations for the sup-figures included on Fig7.3.

Image	Annotations on the image
coast	mountain, trees, sand and lake water
forest	path, ground grass, sky, trees and tree trunk
highway	sky, hill, field, tree, brushes and road
mountain	sky, mountain, trees, building occluded and ground
opencountry	sky, mountain, trees, seawater and sand beach
street	sky, skyscraper, occluded, building, buildings occluded, river water, dock, car, crane occluded, hedge, palm tree and tree
tallbuilding	sky, building, ground grass and road

7.5.1 Evaluation of the Research Methods

There are two main aspects that should be considered by researchers evaluating ensemble methods. These aspects are the *accuracy* and the *reliability*. The accuracy was calculated using the confusion matrix as shown in Table 3.2. The reliability was measured by running each experiment five times and examining the differences between them.

7.5.2 Comparison of the Results

In HES the comparison was carried out using homogeneous ensembles built using the AdaBoost algorithm. This algorithm is one of the most commonly used for classification and is recognised as having good performance. It has the capability to use any type of classification algorithm as its base learner. Thus it is ideally suited as a baseline method against which to compare ours. In FLEM comparison was carried out with homogeneous ensembles built with the AdaBoost algorithm, and comparing with HES. The results of GDLEM were compared with FLEM, HEST and HESG and the results were shown in the critical difference diagram (Figure 6.13). Table 7.5 shows the summary of the comparisons that were used for our research methods.

Table 7.5: Evaluation of methods used in this research.

Research Method	Evaluation
HES	Compared with AdaBoostM1
FLEM	Compared with AdaBoostM1 and HES
GDLEM	Compared with HES and FLEM

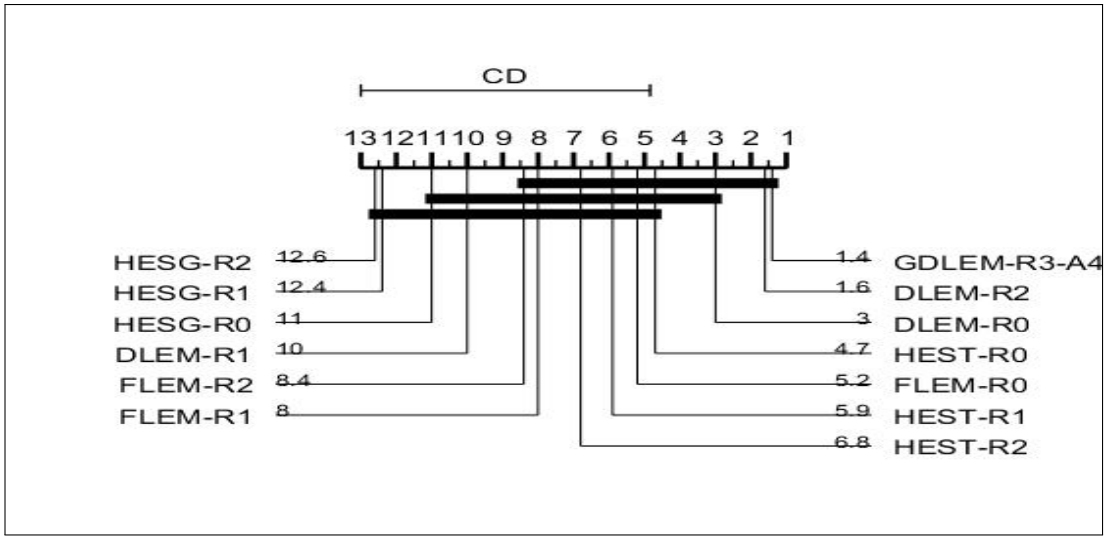

Figure 7.4: Critical difference diagram for the ensembles built with GDELM, DLEM, Feature-Level Ensemble Method(FLEM), Hybrid Ensembles Built with Textual Data(HEST) and with Imagery Data(HESG) for all rules R0, R1, R2 and R3. It shows that the GDLEM with R3 is the best.

7.6 Comparisons Between our Methods

The results of all our experiments generated by HEST, HESG, FLEM and GDLEM were statistically compared. Figure 7.4 shows the critical difference diagram for the GDLEM, DLEM, FLEM, HEST and HESG, with all rules R0, R1, R2 and R3. The GDLEM-R3 is the best on average and a credible explanation is that R3 with appropriate weights can produce the optimal combination of model accuracy and CFD to improve ensemble accuracy.

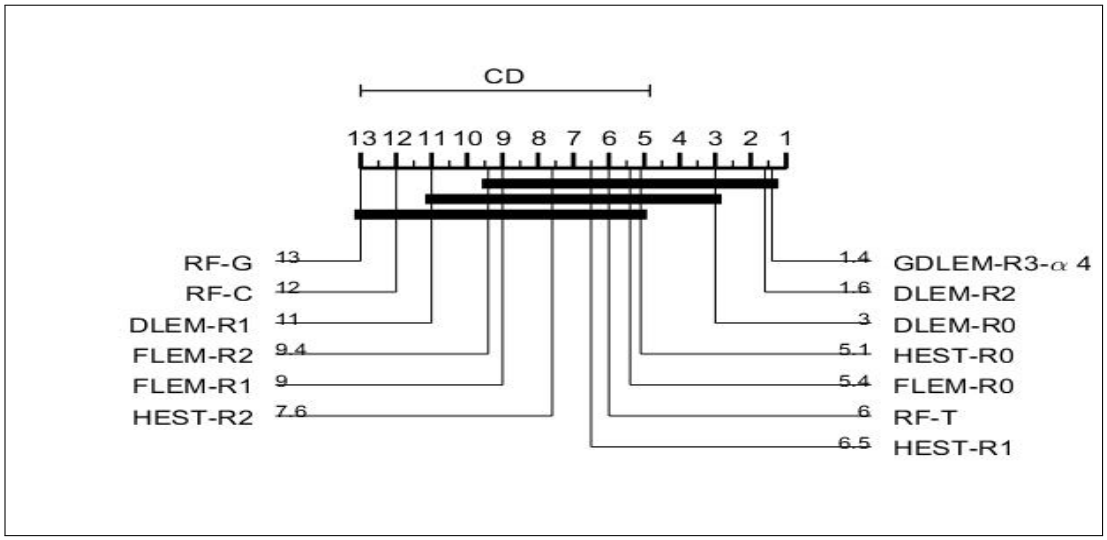

Figure 7.5: Critical difference diagram for the ensembles built with GDELM, DLEM, Feature-Level Ensemble Method(FLEM), Hybrid Ensembles Built with Textual Data(HEST) for all rules R0, R1, R2 and R3;and Random Forest for text (RF-T), image (RF-G) and combined (RF-C). It shows that the GDLEM with R3 is the best.

7.7 Comparison of results with Random Forest

Further comparison was carried out between our methods and the homogeneous ensemble Random Forest. Figure 7.5 shows the critical difference diagram for the GDLEM, DLEM, FLEM and HEST, with all rules R0, R1, R2 and R3; and Random Forest for text (RF-T), image (RF-G) and combined (RF-C). It can be clearly seen that Random Forest with image or combined data gives very poor performance. With text it is still well below the performance of our methods.

7.8 Comparison of results with External Methods and Dataset

We compared our methods with the work presented by Oramas et al. (2018). They used a multimedia dataset for music genre classification using two different types of media: audio and video. The approach they used to extract features

Table 7.6: Description of MSD-1 Dataset attributes for each representation.

Data Representation	# Attributes	Range of Attribute Values
CNN_Audio	2048	0.0 to 6.53
MM_Audio	200	-3.9 to 3.82
CNN_Visual	2048	0.0 to 9.73
MM_Visual	200	-5.18 to 5.17

and perform the classification was Deep Learning (DL). By testing our methods on the same dataset that they used we have been able to perform a completely independent validation of our work.

7.8.1 Dataset Used

Oramas et al. (2018) released their dataset they used to make it easier for comparison and they called it MSD-1.[1] The released dataset was two different representations for each of two different media data types, as shown in Table 7.6.

Table 7.7 reports the number of instances of each genre in the three subsets, and also the genre distribution as percentages of the entire dataset.

7.8.2 Our Experimental Set-up and Results from the Comparison

Our framework is general and allows us to deal with multimedia using different feature extraction methods and different base learning algorithms. Therefore, we used the extracted features exactly as Oramas et al. (2018) released it and a set of base learning algorithms with the default WEKA parameters. The results for each base learning algorithm using the F1 measure as it used in their work are shown in Table 7.8.

As we used the FLEM in our contributions and Oramas et al combined the

[1]https://doi.org/10.5281/zenodo.1240484

Table 7.7: The number of instances for each genre on the train, validation and test subsets. The percentage of elements for each genre is also shown.

Genre	Train	Val	Test	%
Blues	518	120	190	2.68
Country	1351	243	194	5.78
Electronic	3434	725	733	15.81
Folk	858	164	136	3.74
Jazz	1844	373	462	8.66
Latin	390	83	83	1.80
Metal	1749	512	375	8.52
New Age	158	71	38	0.86
Pop	2333	644	466	11.13
Punk	487	132	96	2.31
Rap	1932	380	381	8.71
Reggae	1249	190	266	5.51
RnB	1223	222	396	5.95
Rock	3694	709	829	16.91
World	331	123	46	1.62

Table 7.8: The results of F1 measure on test and validation for each single base learning algorithm used in our experiment. It shows the result for different representations.

	CNN_Audio		MM_Audio		CNN_Visual		MM_Visual	
Model Name	Test	Val	Test	Val	Test	Val	Test	Val
BayesNet	0.337	0.329	0.366	0.347	0.270	0.232	0.253	0.221
SMO	0.331	0.303	0.335	0.307	0.253	0.229	0.254	0.220
RandomForest	0.319	0.306	0.339	0.314	0.205	0.206	0.241	0.216
NaiveBayes	0.285	0.294	0.359	0.341	0.250	0.225	0.257	0.223
PART	0.256	0.254	0.256	0.257	0.177	0.141	0.176	0.177
JRip	0.278	0.267	0.298	0.271	0.192	0.179	0.203	0.207
RandomTree	0.245	0.236	0.261	0.246	0.143	0.135	0.163	0.157
REPTree	0.291	0.269	0.303	0.283	0.167	0.147	0.195	0.175
J48	0.278	0.257	0.272	0.261	0.162	0.142	0.179	0.162

Table 7.9: The results of F1 on test and validation for each single base learning algorithm used in our experiment where we combined the data at the feature level.

	CNN_Audio + MM_Audio		CNN_Visual + MM_Visual		CNN_Audio + CNN_Visual		MM_Audio + MM_Visual		ALL	
Model Name	Test	Val	Test	Val	Test	Val	Test	Val	Test	Val
BayesNet	0.357	0.342	0.276	0.238	0.414	0.367	0.416	0.399	0.434	0.395
SMO	0.335	0.309	0.258	0.226	0.389	0.376	0.411	0.392	0.388	0.369
RandomForest	0.332	0.319	0.234	0.217	0.321	0.299	0.373	0.357	0.355	0.339
NaiveBayes	0.299	0.308	0.252	0.222	0.388	0.375	0.410	0.392	0.399	0.389
PART	0.278	0.247	0.141	0.181	0.286	0.266	0.314	0.288	0.292	0.261
JRip	0.294	0.281	0.206	0.183	0.300	0.302	0.325	0.315	0.324	0.309
RandomTree	0.260	0.238	0.154	0.140	0.208	0.210	0.272	0.272	0.237	0.231
REPTree	0.304	0.284	0.196	0.174	0.287	0.277	0.341	0.308	0.329	0.313
J48	0.279	0.261	0.172	0.144	0.262	0.249	0.294	0.271	0.269	0.260

data at the feature level we also calculated the F1 results for all the combined data as shown in Table 7.9.

Table 7.10 shows the comparison of results between our methods and the work presented by Oramas et al. (2018). The F1 measure was used to calculate the performance as they did. The mean for FLEM was calculated by averaging the results for ensemble sizes 3, 5, 7 and 9 with alpha varied from 0.10 to 1.0 by increasing 0.10 each time, in total there were 80 values. The mean for DLEM was calculated as same as the mean for FLEM but with ensemble sizes 3, 5, 7, 9, 11, 13, 15 and 17 and there were 80 values.

In Audio and Video datasets representation, the mean of our results obtained with FLEM and DLEM are better than the results obtained by Oramas et al. (2018) for each representation. In the A + V representation they obtained slightly better results than the mean of our methods FLEM and DLEM, but on the other hand our best results are better than theirs.

Table 7.10: Comparison of our results with the results obtained by Oramas et al. (2018).

		(Oramas et al., 2018) Results	Best FLEM Ensemble	Worst FLEM	Mean FLEM	Best DLEM Ensemble	Worst DLEM	Mean DLEM
Audio	CNN_Audio	0.336±0.002	0.358	0.333	0.343±0.001			
	MM_Audio	0.334 ±0.003	0.369	0.337	0.356 ±0.003			
	CNN_Audio + MM_Audio	0.346 ±0.002	0.369	0.341	0.353 ±0.002	0.370	0.344	0.359 ±0.002
Video	CNN_Visual	0.255±0.003	0.291	0.267	0.276±0.003			
	MM_Visual	0.239 ±0.002	0.273	0.245	0.263 ±0.002			
	CNN_Visual + MM_Visual	0.245±0.003	0.296	0.267	0.276 ±0.002	0.294	0.250	0.278±0.002
A + V	CNN_Audio + CNN_Visual	0.425 ±0.005	0.423	0.380	0.408 ±0.006			
	MM_Audio + MM_Visual	0.400±0.004	0.427	0.403	0.413±0.001			
	ALL	0.427 ±0.000	0.437	0.408	0.420 ±0.003	0.442	0.401	0.423 ±0.004

7.9 Summary

In this chapter we have presented a comprehensive overview of our methodology and the results of our three main experiments in this research. We have shown how our methods have developed, and the accompanying improvement in performance, starting from using a single media classification to multi-media classification; and from model selection rules that apply one criterion to select a single model, moving to a generalized model selection rule that allows not just to combine more than one criterion to select a single model but determines the weight for each specified criterion. Finally, we have performed a comparison with the work of other researchers using their own dataset, and we have shown that, overall, our methods achieved better results than they did.

Chapter 8

Conclusion and Further Work

In this chapter, we give the conclusion and discuss the contributions of our research. In addition, we will list some recommended further work.

8.1 Conclusion

In this research, we investigated the problem of classifying multi-media data. We noted in the introduction that most existing work on developing or applying methods in this topic has only used one type of multimedia data, rather than several types. In contrast, in our research we have developed machine learning heterogeneous ensemble methods for analyzing datasets containing multiple types of multimedia data.

We address the problem by classifying the data at two different levels. Using different types of multi-media data proved advantageous in this respect as we derived benefits from their characteristics, which enabled us to use two different approaches to classifying the data: feature level and decision level. Different model selection rules were used, which included both dynamic and static rules. We also used a generalised rule to combine complex measures to select a single

model for addition to the ensemble. Our work outperformed other state-of-the-art homogeneous ensemble methods in terms of both accuracy and reliability.

In Section 1.3 we stated that the aim of this research was to investigate ensemble techniques for classifying MMDs, and we set the following objectives:

1. To identify the best procedure for transforming and/or combining several multi-media data sets into a form suitable for use by ensemble classification methods.

2. To develop a methodology for building an effective ensemble classifier for MMDs at two levels, feature level and decision level.

3. To test and critically evaluate our new developed methods.

We consider that we have met these objectives. The first objective was met by the Heterogeneous Ensemble System described in Chapter 4 and illustrated in Figure 4.1.

The second objective was met by the work presented in Chapters 5 and 6, where we described the development of two different levels of ensemble for classifying multimedia data. In addition we investigated model selection rules. Furthermore, in Chapter 6 we developed an advanced generalised rule that uses weighting on multiple criteria for selecting each individual model.

The third objective has been met by extensive testing, as discussed in Section 7.5.1. We compared our methods with the established AdaBoostM1 and with each other as listed in Table 7.5. In addition, all of our methods were compared with Random Forest as shown in Figure 7.5.

Thus, all of our objectives have been met, and we therefore consider that the aim of this research has been accomplished. We discuss the contributions of our research in the next section.

8.2 Contribution

The main contribution of our research was described in Chapters 4, 5 and 6.

In Chapter 4 we developed the main framework for our research in order to solve the problem of imaginary scene classification. Heterogeneous ensemble classification methods were used in conjunction with model selection rules which consider two criteria. These are the accuracy of individual models and the diversity among these models. These criteria were used both individually and in combination, and were employed as a test case to study the capability of heterogeneous ensembles which had been constructed using rules that consider either the accuracy of individual models or their diversity, or both. Three rules were specifically devised using the accuracy of individual models and the diversity measurements among these models to create an ensemble. Our results proved superior to those of previous studies which had used individual models for imaginary scene classification. We found that there are advantages to increasing diversity among the models selected for the ensemble, and that these produced more stable and reliable results. We also discovered that diversity is more effective when used with a larger number of models selected for the ensemble. We therefore concluded that combining models provides considerable benefits in terms of the ensemble's accuracy.

In Chapter 5 we developed a feature level ensemble method (FLEM). This method can aggregate more than one type of data into one big data set which enabled us to apply to it machine learning algorithms using the model selection rules which were developed in the previous chapter. FLEM consists of four stages: extracting features from multimedia subsets and aggregating them into a single dataset, modelling the combined dataset, selecting models with different rules based on various criteria, and building heterogeneous ensembles. Our results demonstrated that FLEM is capable of dealing with multimedia datasets (unstructured text data and imagery data), simultaneously. Furthermore, it builds the best ensembles with appropriate datasets, with either combined multi-media data or single-media data. The heterogeneous ensembles were generally far superior to homogeneous ensembles, both in terms of accuracy and consistency. Our results also showed that there is a need for caution when combining multiple data subsets because the aggregated data may not produce a result which is better than that given by using data subsets of single-media. Possible reasons for this include poor features extracted from each subset, which capture more noise instead of useful information; and/or inappropriate aggregation, which may introduce some inconsistency or even contradictions into the final dataset. This has the potential to cause considerable difficulty and/or confusion in learning.

In Chapter 6 a generalised heterogeneous ensemble method, GDLEM, was developed, for the purpose of classifying multi-media datasets at the decision level. The aim of this was to achieve the best and most consistently accurate results.

Our GDLEM consists of four stages: extracting features from multi-media subsets, modelling the subsets datasets, selecting models with different rules based on various criteria, and building heterogeneous ensembles. The new model selection rule which we developed was demonstrated to be capable of selecting individual models which are less accurate but more diverse. This achieved the best level of ensemble accuracy, which was superior to those obtained by other ensembles, including DLEM, FLEM, HEST and HESG, using the same dataset. We also discovered that heterogeneous ensembles give better results when accuracy and diversity measurements are combined for model selection.

We can conclude by saying that our work is unique in this area and can therefore be considered to be ground-breaking in the field of multi-media data mining for classification problems. Thus our work is a significant advance on what previously has been achieved.

8.3 Limitation

Our work has a number of limitations that impact on how well it might generalise to other data, as listed below, however it should be possible for future work to effectively address them:

1. The lack of multimedia datasets in general meant that we had to develop our methods using only one dataset. However, the test results using the new dataset of (Oramas et al., 2018) show that this is not a significant limitation. Also, we only used two types of data. This was due to the lack of suitable datasets being available

2. We did not use any form of feature selection, which could impact on the analysis of large datasets. The development of effective feature selection methods for this type of analysis would facilitate the analysis of larger datasets.

3. While there are a number of measures that could be used for model selection, our methods are limited to accuracy and CFD. This would particularly impact on the analysis of unbalanced data, but this could be addressed in future work as discussed below.

8.4 Further Work

The achievements of this study point to other areas which could be the subject for future work. These include:

- Creating other complex selection rules by adding more measures to those used in the generalised R3.
- It would prove useful to analyse multi-media datasets which contain other, different types of media, which have not yet been the subject of this research. More experiments could be conducted by using more multi-media datasets.
- It could be useful to apply some feature selection methods on each of the data subsets, to eliminate irrelevant or redundant features, which in turn can reduce the dimensionality of the data and simplify learning.
- Applying this approach on different classification problems like time series classification.
- Increasing model pool size so that there are more choices for model selection.

Bibliography

(2016). An ensemble learning method for scene classification based on hidden markov model image representation. *CoRR*, abs/1607.06794.

Abboute, A., Boudjeriou, Y., Entringer, G., Azé, J., Bringay, S., and Poncelet, P. (2014). Mining twitter for suicide prevention. In *International Conference on Applications of Natural Language to Data Bases/Information Systems*, pages 250–253. Springer.

Abidin, S., Xia, X., Togneri, R., and Sohel, F. (2018). Local binary pattern with random forest for acoustic scene classification. In *2018 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6.

Aburomman, A. A. and Reaz, M. B. I. (2017). A survey of intrusion detection systems based on ensemble and hybrid classifiers. *Computers and Security*, 65:135 – 152.

Alyahyan, S., Farrash, M., and Wang, W. (2016). Heterogeneous ensemble for imaginary scene classification. In *Proceedings of the 8th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2016) - Volume 1: KDIR, Porto - Portugal, November 9 - 11, 2016.*, pages 197–204.

Alyahyan, S. and Wang, W. (2017). Feature level ensemble method for classifying multi-media data. In *International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pages 235–249. Springer.

Alyahyan, S. and Wang, W. (2018a). Decision level ensemble method for classifying multi-media data. *Wireless Networks*, pages 1–9.

Alyahyan, S. and Wang, W. (2018b). Generalised decision level ensemble method for classifying multi-media data. In Bramer, M. and Petridis, M., editors, *Artificial Intelligence XXXV*, pages 326–339, Cham. Springer International Publishing.

Atrey, P. K., Hossain, M. A., El Saddik, A., and Kankanhalli, M. S. (2010). Multimodal fusion for multimedia analysis: a survey. *Multimedia systems*, 16(6):345–379.

Audebert, N., Saux, B. L., and Lefevrey, S. (2017). Fusion of heterogeneous data in convolutional networks for urban semantic labeling. In *2017 Joint Urban Remote Sensing Event (JURSE)*, pages 1–4.

Bagnall, A., Davis, L., Hills, J., and Lines, J. (2012). Transformation based ensembles for time series classification. In *Proceedings of the 2012 SIAM international conference on data mining*, pages 307–318. SIAM.

Bagnall, A., Lines, J., Hills, J., and Bostrom, A. (2015). Time-series classification with cote: the collective of transformation-based ensembles. *IEEE Transactions on Knowledge and Data Engineering*, 27(9):2522–2535.

Bai, Y., Guo, L., Jin, L., and Huang, Q. (2009). A novel feature extraction method using pyramid histogram of orientation gradients for smile recognition. In *2009 16th IEEE International Conference on Image Processing (ICIP)*, pages 3305–3308. IEEE.

Ballard, C. and Wang, W. (2016). Dynamic ensemble selection methods for heterogeneous data mining. In *Intelligent Control and Automation (WCICA), 2016 12th World Congress on*, pages 1021–1026. IEEE.

Baradwaj, B. K. and Pal, S. (2011). Mining educational data to analyze students' performance. *International Journal of Advanced Computer Science and Applications*, 2(6):63–69.

Bhatt, C. A. and Kankanhalli, M. S. (2011). Multimedia data mining: state of the art and challenges. *Multimedia Tools and Applications*, 51(1):35–76.

Bian, S. (2006). *Data mining ensemble hierarchy, diversity and accuracy.* PhD thesis, University of East Anglia.

Bifet, A., Holmes, G., Pfahringer, B., Kirkby, R., and Gavaldà, R. (2009). New ensemble methods for evolving data streams. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 139–148, New York, NY, USA. ACM.

Borji, A. (2007). Combining heterogeneous classifiers for network intrusion detection. In *Annual Asian Computing Science Conference*, pages 254–260. Springer.

Bosch, A., Zisserman, A., and Muñoz, X. (2006). Scene classification via plsa. In *Computer Vision–ECCV 2006*, pages 517–530. Springer.

Bouckaert, R. R., Frank, E., Hall, M., Kirkby, R., Reutemann, P., Seewald, A., and Scuse, D. (2016). Weka manual for version 3-9-1. *The University of Waikato, Hamilton, New Zealand.*

Bramer, M. (2007). *Principles of data mining*, volume 180. Springer.

Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123–140.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Chen, S.-C., Shyu, M.-L., Zhang, C., and Strickrott, J. (2001). Multimedia data mining for traffic video sequences. In *Proceedings of the Second International Conference on Multimedia Data Mining*, MDMKDD'01, pages 78–86, Berlin, Heidelberg. Springer-Verlag.

Chen, Z.-Y., Fan, Z.-P., and Sun, M. (2015). Behavior-aware user response modeling in social media: Learning from diverse heterogeneous data. *European Journal of Operational Research*, 241(2):422–434.

Cover, T. M. and Hart, P. E. (1967). Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21–27.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE.

De Stefano, C., Fontanella, F., Folino, G., and Di Freca, A. S. (2011). A bayesian approach for combining ensembles of gp classifiers. In *International Workshop on Multiple Classifier Systems*, pages 26–35. Springer.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, 7(Jan):1–30.

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *Multiple Classifier Systems. MCS 2000. Lecture Notes in Computer Science*, volume 1857, pages 1–15, Berlin, Heidelberg. Springer.

Do, C.-T., Douzal-Chouakria, A., Marié, S., Rombaut, M., and Varasteh, S. (2017). Multi-modal and multi-scale temporal metric learning for a robust time series nearest neighbors classification. *Information Sciences*, 418:272–285.

Dong, Y.-S. and Han, K.-S. (2004). A comparison of several ensemble methods for text categorization. In *Services Computing, 2004.(SCC 2004). Proceedings. 2004 IEEE International Conference on*, pages 419–422. IEEE.

Garcia-Ceja, E., Tejada, C. E. G., and Brena, R. (2018). Multi-view stacking for activity recognition with sound and accelerometer data. *Information Fusion*, 40:45 – 56.

Gashler, M., Giraud-Carrier, C., and Martinez, T. (2008). Decision tree ensemble: Small heterogeneous is better than large homogeneous. In *2008 Seventh International Conference on Machine Learning and Applications*, pages 900–905.

Ghose, A. and Ipeirotis, P. G. (2011). Estimating the helpfulness and economic impact of product reviews: Mining text and reviewer characteristics. *IEEE Transactions on Knowledge and Data Engineering*, 23(10):1498–1512.

Giacinto, G. and Roli, F. (2001). Design of effective neural network ensembles for image classification purposes. *Image and Vision Computing*, 19(9):699–707.

Gomes, H. M., Barddal, J. P., Enembreck, F., and Bifet, A. (2017). A survey on ensemble learning for data stream classification. *ACM Comput. Surv.*, 50(2):23:1–23:36.

Gosain, A. and Bhugra, M. (2013). A comprehensive survey of association rules on quantitative data in data mining. In *Information and Communication Technologies (ICT), 2013 IEEE Conference on*, pages 1003–1008. IEEE.

Gour, N. and Khanna, P. (2019). Automated glaucoma detection using gist and pyramid histogram of oriented gradients (phog) descriptors. *Pattern Recognition Letters*.

Grauman, K. and Darrell, T. (2005). The pyramid match kernel: Discriminative classification with sets of image features. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1458–1465. IEEE.

Grove, A. J. and Schuurmans, D. (1998). Boosting in the limit: Maximizing the margin of learned ensembles. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pages 692–699.

Han, J., Pei, J., and Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.

Haq, A. and Wilk, S. (2017). Fusion of clinical data: A case study to predict the type of treatment of bone fractures. In *European Conference on Advances in Databases and Information Systems*, pages 294–301. Springer.

Haque, M. N., Noman, N., Berretta, R., and Moscato, P. (2016). Heterogeneous ensemble combination search using genetic algorithm for class imbalanced data classification. *PLOS ONE*, 11(1):1–28.

Hofmann, T. (2001). Unsupervised learning by probabilistic latent semantic analysis. *Machine learning*, 42(1-2):177–196.

Huang, D.-S. and Zheng, C.-H. (2006). Independent component analysis-based penalized discriminant method for tumor classification using gene expression data. *Bioinformatics*, 22(15):1855–1862.

Ikonomakis, M., Kotsiantis, S., and Tampakas, V. (2005). Text classification using machine learning techniques. *WSEAS transactions on computers*, 4(8):966–974.

Imani, M. and Ghassemian, H. (2020). An overview on spectral and spatial information fusion for hyperspectral image classification: Current trends and challenges. *Information fusion*, 59:59–83.

Jurek, A., Bi, Y., Wu, S., and Nugent, C. (2014). A survey of commonly used ensemble-based classification techniques. *The Knowledge Engineering Review*, 29(05):551–581.

Kamavisdar, P., Saluja, S., and Agrawal, S. (2013). A survey on image classification approaches and techniques. *International Journal of Advanced Research in Computer and Communication Engineering*, 2(1):1005–1009.

Kang, S., Cho, S., and Kang, P. (2015). Multi-class classification via heterogeneous ensemble of one-class classifiers. *Engineering Applications of Artificial Intelligence*, 43:35–43.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., and Fei-Fei, L. (2014). Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732.

Ke, Y. and Sukthankar, R. (2004). Pca-sift: A more distinctive representation for local image descriptors. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–506. IEEE.

Keedwell, E. and Narayanan, A. (2009). Gene expression classification using

multi-objective ensembles. In *AISB 2009 Convention– Proceedings of the Symposium Evolutionary Systems*, Edinburgh, UK. SSAISB.

Koh, J. E. W., Ng, E. Y. K., Bhandary, S. V., Laude, A., and Acharya, U. R. (2018). Automated detection of retinal health using phog and surf features extracted from fundus images. *Applied Intelligence*, 48(5):1379–1393.

Kolter, J. Z. and Maloof, M. A. (2003). Dynamic weighted majority: a new ensemble method for tracking concept drift. In *ICDM 2003: Proceedings of the Third IEEE International Conference on Data Mining*, pages 123–130.

Kotsiantis, S. B., Zaharakis, I., and Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160:3–24.

Kulkarni, V. Y. and Sinha, P. K. (2012). Pruning of random forest classifiers: A survey and future directions. In *Data Science & Engineering (ICDSE), 2012 International Conference on*, pages 64–68. IEEE.

Kuncheva, L. I. and Whitaker, C. J. (2003). Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine learning*, 51(2):181–207.

Leopold, E. and Kindermann, J. (2002). Text categorization with support vector machines. how to represent texts in input space? *Machine Learning*, 46(1-3):423–444.

Lertampaiporn, S., Thammarongtham, C., Nukoolkit, C., Kaewkamnerdpong, B., and Ruengjitchatchawalya, M. (2013). Heterogeneous ensemble approach with discriminative features and modified-smotebagging for pre-mirna classification. *Nucleic acids research*, 41(1):e21–e21.

Li, H., Jiang, T., and Zhang, K. (2004). Efficient and robust feature extraction by maximum margin criterion. In *Advances in neural information processing systems*, pages 97–104.

Lines, J., Davis, L. M., Hills, J., and Bagnall, A. (2012). A shapelet transform for time series classification. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 289–297. ACM.

Lines, J. A. (2015). *Time Series Classi*

cation through Transformation and Ensembles. PhD thesis, University of East Anglia.

Liu, H., Wei, Z., Chen, Y., Pan, J., Lin, L., and Ren, Y. (2017). Drone detection based on an audio-assisted camera array. In *Third IEEE International Conference on Multimedia Big Data, Laguna Hills, CA, USA, April 19-21, 2017*, pages 402–406. IEEE Computer Society.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110.

Lu, D. and Weng, Q. (2007). A survey of image classification methods and techniques for improving classification performance. *International Journal of Remote Sensing*, 28(5):823–870.

Madsen, R. E., Sigurdsson, S., Hansen, L. K., and Larsen, J. (2004). Pruning the vocabulary for better context recognition. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 2, pages 483–488. IEEE.

Maimon, O. and Rokach, L. (2005). *Decomposition methodology for knowledge discovery and data mining*, pages 981–1003. Springer.

Manjunath, R. and Balaji, S. (2014). Review and analysis of multimedia data mining tasks and models. *International Journal of Innovative Research in Computer and Communication Engineering*, 2.

Manjunath, T., Hegadi, R. S., and Ravikumar, G. (2010). A survey on multimedia data mining and its relevance today. *IJCSNS*, 10(11):165–170.

Matikainen, P., Sukthankar, R., and Hebert, M. (2012). Classifier ensemble recommendation. In *European Conference on Computer Vision*, pages 209–218. Springer.

Medjahed, S. A. (2015). A comparative study of feature extraction methods in images classification. *International journal of image, graphics and signal processing*, 7(3):16.

Mehmood, T. and Rasheed, Z. (2015). Multivariate procedure for variable selection and classification of high dimensional heterogeneous data. *Communications for Statistical Applications and Methods*, 22(6):575–587.

Mendes-Moreira, J., Soares, C., Jorge, A. M., and Sousa, J. F. D. (2012). Ensemble approaches for regression: A survey. *ACM Computing Surveys (CSUR)*, 45(1):10.

Meng, X.-H., Huang, Y.-X., Rao, D.-P., Zhang, Q., and Liu, Q. (2013). Comparison of three data mining models for predicting diabetes or prediabetes by risk factors. *The Kaohsiung journal of medical sciences*, 29(2):93–99.

Michie, D., Spiegelhalter, D. J., and Taylor, C. C. (1994). Machine learning, neural and statistical classification.

Mikolajczyk, K. and Schmid, C. (2004). Scale & affine invariant interest point detectors. *International journal of computer vision*, 60(1):63–86.

Miner, G. (2012). *Practical text mining and statistical analysis for non-structured text data applications*. Academic Press.

Mishra, S. K. (2013). A review of ensemble technique for improving majority voting for classifier. *International Journal of Advanced Research in Computer Science and Software Engineering*, 3(1).

Mojahed, A., Bettencourt-Silva, J. H., Wang, W., and de la Iglesia, B. (2015). Applying clustering analysis to heterogeneous data using similarity matrix fusion (smf). In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 251–265. Springer.

Mojahed, A. and de la Iglesia, B. (2017). An adaptive version of k-medoids to deal with the uncertainty in clustering heterogeneous data using an intermediary fusion approach. *Knowledge and Information Systems*, 50(1):27–52.

Naaman, M. (2012). Social multimedia: highlighting opportunities for search and mining of multimedia data in social media applications. *Multimedia Tools and Applications*, 56(1):9–34.

Oh, J. H. and Bandi, B. (2002). Multimedia data mining framework for raw video sequences. In *Proceedings of the Third International Conference on Multimedia Data Mining*, MDMKDD'02, pages 1–10, London, UK, UK. Springer-Verlag.

Oliva, A. and Torralba, A. (2001). Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3):145–175.

Onan, A. (2018). An ensemble scheme based on language function analysis and feature engineering for text genre classification. *Journal of Information Science*, 44(1):28–47.

Oramas, S., Barbieri, F., Nieto, O., and Serra, X. (2018). Multimodal deep learning for music genre classification. *Transactions of the International Society for Music Information Retrieval. 2018; 1 (1): 4-21.*

Orlando, J. I., Prokofyeva, E., del Fresno, M., and Blaschko, M. B. (2017). Convolutional neural network transfer for automated glaucoma identification. In Romero, E., Lepore, N., Brieva, J., Brieva, J., and and, I. L., editors, *12th International Symposium on Medical Information Processing and Analysis*, volume 10160, pages 241 – 250. International Society for Optics and Photonics, SPIE.

Partalas, I., Tsoumakas, G., and Vlahavas, I. (2009). Pruning an ensemble of classifiers via reinforcement learning. *Neurocomputing*, 72(7-9):1900–1909.

Partridge, D. and Krzanowski, W. (1997). Software diversity: practical statistics for its measurement and exploitation. *Information and software technology*, 39(10):707–717.

Phyu, T. N. (2009). Survey of classification techniques in data mining. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, volume 1, pages 18–20.

Pooja, S. R. (2013). A comparative study of instance reduction techniques. In *Proceedings of 2nd International Conference on Emerging Trends in Engineering and Management, International Journal of Advances in Engineering Sciences*, volume 3, pages 7–13.

Prasad, R. and Sebastian, M. P. (2014). A survey on phrase structure learning methods for text classification. *International Journal on Natural Language Computing*, 3(2):33–46.

Rätsch, G., Onoda, T., and Müller, K.-R. (2001). Soft margins for adaboost. *Machine learning*, 42(3):287–320.

Rogati, M. and Yang, Y. (2002). High-performing feature selection for text classification. In *Proceedings of the eleventh international conference on Information and knowledge management*, pages 659–661. ACM.

Rokach, L. and Maimon, O. (2014). *Data mining with decision trees: theory and applications*. World scientific.

Salloum, S. A., Al-Emran, M., Monem, A. A., and Shaalan, K. (2017). A survey of text mining in social media: facebook and twitter perspectives. *Adv. Sci. Technol. Eng. Syst. J*, 2(1):127–133.

Seeger, C., Müller, A., Schwarz, L., and Manz, M. (2016). Towards road type classification with occupancy grids. In *IEEE Intelligent Vehicles Symposium (IV) Workshop: DeepDriving-Learning Representations for Intelligent Vehicles*, pages 1–4.

Seera, M. and Lim, C. P. (2014). A hybrid intelligent system for medical data classification. *Expert Systems with Applications*, 41(5):2239–2249.

Seijo-Pardo, B., Porto-Diaz, I., Bolon-Canedo, V., and Alonso-Betanzos, A. (2017). Ensemble feature selection: Homogeneous and heterogeneous approaches. *Knowledge-Based Systems*, 118:124 – 139.

Shah, D. and Limbad, N. (2015). A literature survey on contrast data mining. *International Journal of Science and Research, pages: 954*, 958.

Shannon, C. E. (2001). A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55.

Shoeb, M. and Ahmed, J. (2017). Sentiment analysis and classification of tweets using data mining. *International Research Journal of Engineering and Technology (IRJET)*, 4(12):1471–1474.

Smetek, M. and Trawiński, B. (2011). Selection of heterogeneous fuzzy model ensembles using self-adaptive genetic algorithms. *New Generation Computing*, 29(3):309.

Sobolevsky, S., Sitko, I., Grauwin, S., des Combes, R. T., Hawelka, B., Arias, J. M., and Ratti, C. (2014). Mining urban performance: Scale-independent classification of cities based on individual economic transactions.

Tan, P.-N., Steinbach, M., and Kumar, V. (2006). Introduction to data mining.

Tomar, D. and Agarwal, S. (2013). A survey on data mining approaches for healthcare. *International Journal of Bio-Science and Bio-Technology*, 5(5):241–266.

Tsoumakas, G., Katakis, I., and Vlahavas, I. (2004). Effective voting of heterogeneous classifiers. In *European Conference on Machine Learning*, pages 465–476. Springer.

Tuarob, S., Tucker, C. S., Salathe, M., and Ram, N. (2014). An ensemble heterogeneous classification methodology for discovering health-related knowledge in social media messages. *Journal of biomedical informatics*, 49:255–268.

Van Erp, M., Vuurpijl, L., and Schomaker, L. (2002). An overview and comparison of voting methods for pattern recognition. In *Frontiers in Handwriting Recognition, 2002. Proceedings. Eighth International Workshop on*, pages 195–200. IEEE.

van Loon, W., Fokkema, M., Szabo, B., and de Rooij, M. (2020). Stacked penalized logistic regression for selecting views in multi-view learning. *Information Fusion*, 61:113 – 123.

Veni, C. V. K. and Rani, T. S. (2014). Ensemble based classification using small training sets : A novel approach. In *2014 IEEE Symposium on Computational Intelligence in Ensemble Learning (CIEL)*, pages 1–8.

Vijayarani, S. and Sakila, A. (2015). Multimedia mining research-an overview. *International Journal of Computer Graphics and Animation*, 5(1):69.

Wang, W. (2008). Some fundamental issues in ensemble methods. In *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*, pages 2243–2250. IEEE.

Williams, T. P. and Gong, J. (2014). Predicting construction cost overruns using text mining, numerical data and ensemble classifiers. *Automation in Construction*, 43:23–29.

Witten, I. H. and Frank, E. (2005). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

Witten, I. H., Frank, E., Hall, M. A., and Pal, C. J. (2016). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

Wlodarczak, P., Soar, J., and Ally, M. (2015). Multimedia data mining using deep learning. In *Digital Information Processing and Communications (ICDIPC), 2015 Fifth International Conference on*, pages 190–196. IEEE.

Woloszynski, T. and Kurzynski, M. (2011). A probabilistic model of classifier competence for dynamic ensemble selection. *Pattern Recognition*, 44(10-11):2656–2668.

Woźniak, M., Graña, M., and Corchado, E. (2014a). A survey of multiple classifier systems as hybrid systems. *Information Fusion*, 16:3–17.

Woźniak, M., Graña, M., and Corchado, E. (2014b). A survey of multiple classifier systems as hybrid systems. *Information Fusion*, 16:3–17.

Xiao, X.-Y., Hu, R., Zhang, S.-W., and Wang, X.-F. (2010). Hog-based approach for leaf classification. In Huang, D.-S., Zhang, X., Reyes García, C. A., and Zhang, L., editors, *Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence*, pages 149–155, Berlin, Heidelberg. Springer Berlin Heidelberg.

Xue, Z., You, D., Candemir, S., Jaeger, S., Antani, S., Long, L. R., and Thoma, G. R. (2015). Chest x-ray image view classification. In *2015 IEEE 28th International Symposium on Computer-Based Medical Systems*, pages 66–71. IEEE.

Yamanishi, K. (1999). Distributed cooperative bayesian learning strategies. *Information and Computation*, 150(1):22–56.

Yan, R., Liu, Y., Jin, R., and Hauptmann, A. (2003). On predicting rare classes with svm ensembles in scene classification. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 3, pages III–21. IEEE.

Yan, Y., Chen, M., Shyu, M., and Chen, S. (2015). Deep learning for imbalanced multimedia data classification. In *2015 IEEE International Symposium on Multimedia (ISM)*, pages 483–488.

Yang, J., Jiang, Y.-G., Hauptmann, A. G., and Ngo, C.-W. (2007). Evaluating bag-of-visual-words representations in scene classification. In *Proceedings*

of the international workshop on Workshop on multimedia information retrieval, pages 197–206. ACM.

Yang, Y. and Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. In *ICML*, volume 97, pages 412–420.

Zhang, J., Wang, L., and Wen, X. (2019). Combination of gist and phog features for calligraphy styles classification. In *Proceedings of the 2019 4th International Conference on Multimedia Systems and Signal Processing*, ICMSSP 2019, pages 21–24, New York, NY, USA. ACM.

Zhang, L. and Zhou, W.-D. (2011). Sparse ensembles using weighted combination methods based on linear programming. *Pattern Recognition*, 44(1):97–106.

Appendix A

The results for Homogeneous Ensembles

Table A.1: The homogeneous ensemble results for J48

	N3			N5			N7			N9			Mean			AdaBoost	Single
	R0	R1	R2	R0	R1	R2	R0	R1	R2	R0	R1	R2	R0	R1	R2	AdaBoost	Single
Run 1	85.12	86.01	84.82	87.65	86.16	87.35	88.84	88.24	88.54	88.54	88.84	88.54	87.54	87.31	87.31	90.63	83.48
Run 2	85.71	85.42	86.90	86.90	86.76	86.90	87.65	87.65	87.80	88.54	88.39	88.39	87.20	87.05	87.50	89.58	83.48
Run 3	87.05	87.05	87.35	87.95	88.10	89.29	88.10	89.73	89.88	88.84	89.73	88.99	87.98	88.65	88.88	90.33	84.52
Run 4	87.35	87.05	87.05	89.58	89.29	89.14	90.18	90.03	90.03	90.33	90.48	90.48	89.36	89.21	89.17	90.63	84.67
Run 5	85.12	85.12	86.16	87.50	87.20	87.65	87.95	88.84	87.80	88.24	88.10	88.99	87.20	87.31	87.65	90.33	84.97
Mean	86.07	86.13	86.46	87.92	87.50	88.07	88.54	88.90	88.81	88.90	89.11	89.08	87.86	87.91	88.10	90.30	84.23
SD	1.07	0.90	1.01	1.01	1.22	1.08	1.01	1.00	1.09	0.83	0.98	0.83	0.90	0.96	0.86	0.43	0.70

Table A.2: The homogeneous ensemble results for BayesNet

	N3			N5			N7			N9			Mean			AdaBoost	Single
	R0	R1	R2	R0	R1	R2	R0	R1	R2	R0	R1	R2	R0	R1	R2		
Run 1	76.04	75.74	75.89	76.19	75.89	75.89	76.64	76.49	76.04	76.64	76.64	76.04	76.38	76.19	75.97	79.46	75.89
Run 2	76.34	76.19	76.04	75.60	75.74	75.74	75.45	75.45	75.60	75.30	75.30	75.45	75.67	75.67	75.71	80.36	75.45
Run 3	75.45	75.30	75.74	75.45	75.30	75.60	75.00	75.30	75.45	75.30	75.30	75.30	75.30	75.30	75.52	79.91	76.39
Run 4	78.27	77.68	77.68	77.98	78.13	77.83	78.27	78.27	77.98	78.57	78.42	78.27	78.27	78.13	77.94	81.85	77.98
Run 5	78.57	78.57	78.13	78.13	78.42	78.13	78.27	78.57	78.42	78.27	78.42	78.42	78.31	78.50	78.27	81.10	77.53
Mean	76.93	76.70	76.70	76.67	76.70	76.64	76.73	76.82	76.70	76.82	76.82	76.70	76.79	76.76	76.68	80.54	76.65
SD	1.40	1.38	1.12	1.29	1.46	1.23	1.53	1.54	1.40	1.57	1.57	1.53	1.43	1.46	1.32	0.95	1.08

Table A.3: The homogeneous ensemble results for NaiveBayes

	N3			N5			N7			N9			Mean			AdaBoost	Single
	R0	R1	R2	R0	R1	R2	R0	R1	R2	R0	R1	R2	R0	R1	R2		
Run1	72.17	72.77	72.77	72.92	72.92	72.77	73.07	73.07	73.07	73.21	73.07	73.07	72.84	72.95	72.92	76.49	72.77
Run2	75.00	74.26	74.26	75.00	74.85	74.11	75.00	74.85	74.26	74.55	74.70	74.55	74.89	74.67	74.29	75.74	74.85
Run3	72.02	72.17	72.02	72.17	72.17	71.28	72.02	72.17	71.28	71.58	71.88	71.28	71.95	72.10	71.47	74.40	74.21
Run4	75.15	75.00	75.00	74.85	75.00	74.85	74.40	74.55	75.00	74.85	74.85	74.85	74.81	74.85	74.93	75.89	74.55
Run5	74.26	74.26	74.26	74.55	74.70	74.26	74.11	74.85	74.40	74.40	74.26	74.40	74.33	74.52	74.33	74.70	73.51
Mean	73.72	73.69	73.66	73.90	73.93	73.45	73.72	73.90	73.60	73.72	73.75	73.63	73.76	73.82	73.59	75.45	73.98
SD	1.52	1.17	1.22	1.27	1.29	1.43	1.18	1.22	1.48	1.35	1.26	1.48	1.31	1.22	1.40	0.87	0.84

Table A.4: The homogeneous ensemble results for IBk

IBk	N3			N5			N7			N9			Mean			AdaBoost	Single
	R0	R1	R2	R0	R1	R2	R0	R1	R2	R0	R1	R2	R0	R1	R2		
Run1	87.05	87.20	86.76	87.80	87.20	86.46	87.05	87.05	87.35	87.35	87.20	87.35	87.31	87.17	86.98	86.76	87.35
Run2	83.48	82.14	82.14	83.48	83.04	82.74	83.48	82.59	82.89	82.59	82.59	82.59	83.26	82.59	82.59	82.44	82.89
Run3	81.99	81.55	82.29	82.29	82.44	81.55	82.14	82.59	81.85	82.29	82.29	82.29	82.18	82.22	81.99	82.44	83.13
Run4	85.27	83.93	84.38	84.97	84.97	84.97	84.67	84.97	84.67	84.97	84.97	84.52	84.97	84.71	84.64	85.42	84.97
Run5	84.67	83.78	83.78	83.63	83.63	83.33	83.48	83.48	83.93	83.33	83.93	84.23	83.78	83.71	83.82	82.89	84.23
Mean	84.49	83.72	83.87	84.43	84.26	83.81	84.17	84.14	84.14	84.11	84.20	84.20	84.30	84.08	84.00	83.99	84.51
SD	1.90	2.20	1.88	2.11	1.90	1.93	1.85	1.90	2.09	2.09	1.99	2.02	1.96	1.98	1.96	1.98	1.80

Table A.5: The homogeneous ensemble results for JRip

	N3			N5			N7			N9			Mean			AdaBoost	Single
	R0	R1	R2	R0	R1	R2	R0	R1	R2	R0	R1	R2	R0	R1	R2		
Run1	82.44	83.04	79.91	83.63	82.89	81.85	83.78	84.38	82.74	83.63	83.78	83.63	83.37	83.52	82.03	85.12	79.46
Run2	83.18	84.52	82.44	85.12	84.97	85.12	85.42	86.61	87.20	86.16	86.31	87.05	84.97	85.60	85.45	89.14	79.17
Run3	84.23	84.38	83.04	84.52	85.71	83.93	85.86	85.57	86.01	86.61	86.61	86.61	85.31	85.57	84.90	87.80	80.16
Run4	83.04	81.99	81.99	84.23	84.52	84.67	85.71	86.01	86.61	86.16	86.61	86.46	84.78	84.78	84.93	87.80	79.17
Run5	84.97	84.38	83.33	86.61	86.01	86.01	86.16	86.46	86.46	86.46	86.90	87.35	86.05	85.94	85.79	88.84	81.99
Mean	83.57	83.66	82.14	84.82	84.82	84.32	85.39	85.80	85.80	85.80	86.04	86.22	84.90	85.08	84.62	87.74	79.99
SD	1.01	1.11	1.35	1.13	1.23	1.57	0.94	0.90	1.77	1.23	1.28	1.49	0.98	0.97	1.49	1.58	1.19

Table A.6: The homogeneous ensemble results for PART

	N3			N5			N7			N9						AdaBoost	Single
	R0	R1	R2	R0	R1	R2	R0	R1	R2	R0	R1	R2	R0	R1	R2		
Run1	87.20	87.50	87.50	89.88	90.63	89.58	90.33	90.63	90.03	90.63	90.33	90.77	89.51	89.77	89.47	91.37	86.01
Run2	88.39	89.58	89.58	89.43	89.43	89.88	90.63	90.18	90.33	90.18	89.43	90.03	89.66	89.66	89.96	91.37	87.95
Run3	86.90	87.35	88.10	88.99	88.69	88.84	89.29	89.58	89.88	90.33	90.33	90.48	88.88	88.99	89.32	93.30	86.71
Run4	88.84	88.10	87.80	90.63	91.07	90.03	91.37	90.92	90.92	91.07	91.07	91.67	90.48	90.29	90.10	91.67	86.76
Run5	89.73	88.84	87.95	91.22	89.43	88.84	90.48	89.43	90.03	90.18	90.03	89.88	90.40	89.43	89.17	91.82	86.31
Mean	88.21	88.27	88.18	90.03	89.85	89.43	90.42	90.15	90.24	90.48	90.24	90.57	89.78	89.63	89.61	91.90	86.75
SD	1.17	0.94	0.81	0.90	0.97	0.57	0.75	0.64	0.42	0.38	0.59	0.71	0.67	0.48	0.40	0.81	0.74

Table A.7: The homogeneous ensemble results for RandomTree

	N3			N5			N7			N9			Mean			AdaBoost	Single
	R0	R1	R2	R0	R1	R2	R0	R1	R2	R0	R1	R2	R0	R1	R2		
Run1	62.80	63.84	63.84	67.86	66.07	66.37	67.56	68.30	69.49	72.02	70.68	72.17	67.56	67.22	67.97	50.74	50.60
Run2	58.48	56.10	58.18	61.76	65.77	64.43	64.14	66.22	66.07	66.82	66.96	66.22	62.80	63.76	63.73	51.34	49.40
Run3	58.04	58.04	61.61	63.69	66.22	65.33	68.60	68.60	67.56	69.94	69.94	70.09	65.07	65.70	66.15	52.83	47.62
Run4	59.82	61.46	57.89	68.30	67.11	69.35	70.68	70.98	69.94	73.07	71.13	70.98	67.97	67.67	67.04	48.36	49.55
Run5	64.14	57.14	60.42	66.37	65.48	65.33	68.01	70.24	69.64	71.58	69.64	71.28	67.52	65.63	66.67	52.53	54.46
Mean	60.65	59.32	60.39	65.60	66.13	66.16	67.80	68.87	68.54	70.68	69.67	70.15	66.18	66.00	66.31	51.16	50.33
SD	2.69	3.23	2.48	2.80	0.62	1.91	2.37	1.85	1.67	2.44	1.62	2.32	2.21	1.54	1.59	1.78	2.55

Table A.8: The homogeneous ensemble results for REPTree

	N3			N5			N7			N9			Mean			AdaBoost	Single
	R0	R1	R2	R0	R1	R2	R0	R1	R2	R0	R1	R2	R0	R1	R2		
Run1	77.53	77.53	77.53	78.42	80.95	79.76	79.61	80.51	80.80	79.17	79.61	79.61	78.68	79.65	79.43	87.20	77.53
Run2	77.38	76.19	77.23	80.65	80.51	80.95	80.95	81.40	81.10	80.95	81.10	80.36	79.99	79.80	79.91	84.97	73.66
Run3	77.98	77.68	77.83	80.51	79.02	81.99	80.80	79.91	79.91	79.91	81.25	81.55	79.80	79.46	80.32	86.46	74.60
Run4	78.87	77.98	78.72	81.55	81.55	80.95	81.85	83.63	83.18	82.14	83.63	83.04	81.10	81.70	81.47	84.97	75.30
Run5	76.34	76.49	75.74	77.53	77.83	80.06	77.08	78.13	77.83	78.87	79.46	78.72	77.46	77.98	78.09	86.61	73.51
Mean	77.62	77.17	77.41	79.73	79.97	80.74	80.06	80.71	80.57	80.21	81.01	80.65	79.40	79.72	79.84	86.04	74.92
SD	0.92	0.78	1.09	1.68	1.52	0.88	1.84	2.02	1.94	1.35	1.68	1.69	1.39	1.33	1.24	1.02	1.63

Table A.9: The homogeneous ensemble results for SMO

	N3			N5			N7			N9			Mean			AdaBoost	Single
	R0	R1	R2	R0	R1	R2	R0	R1	R2	R0	R1	R2	R0	R1	R2		
Run1	94.49	93.90	93.90	94.05	94.05	94.05	94.35	94.05	94.64	94.49	94.49	94.49	94.35	94.12	94.27	93.75	93.75
Run2	93.60	94.35	94.35	93.90	93.75	93.90	94.20	94.05	94.05	94.35	94.35	94.35	94.01	94.12	94.16	94.20	94.20
Run3	93.30	94.05	93.60	92.71	92.71	93.75	93.01	93.15	93.60	93.15	93.01	93.15	93.04	93.23	93.53	92.26	91.87
Run4	93.90	93.60	93.60	93.75	93.60	93.60	93.60	93.75	93.45	93.60	93.75	93.60	93.71	93.68	93.56	93.90	93.90
Run5	93.45	93.15	93.15	93.60	93.60	93.30	93.15	93.01	93.01	93.45	93.45	93.30	93.42	93.30	93.19	92.71	92.71
Mean	93.75	93.81	93.72	93.60	93.54	93.72	93.66	93.60	93.75	93.81	93.81	93.78	93.71	93.69	93.74	93.36	93.28
SD	0.47	0.45	0.44	0.53	0.50	0.29	0.60	0.49	0.62	0.58	0.62	0.61	0.51	0.43	0.46	0.83	0.97

Table A.10: The homogeneous ensemble results for LWL

	N3			N5			N7			N9			Mean			AdaBoost	Single
	R0	R1	R2	R0	R1	R2	R0	R1	R2	R0	R1	R2	R0	R1	R2		
Run1	65.03	64.73	63.84	64.88	65.18	64.88	63.69	63.99	63.24	62.95	62.95	62.80	64.14	64.21	63.69	82.89	63.84
Run2	64.73	64.14	63.24	63.69	63.69	63.99	63.39	62.95	63.10	62.95	62.80	63.24	63.69	63.39	63.39	82.14	63.10
Run3	65.33	64.73	65.18	64.14	64.73	63.99	64.43	64.73	63.84	64.58	64.14	64.14	64.62	64.58	64.29	81.25	61.31
Run4	63.84	63.39	63.39	63.24	63.99	63.39	62.95	64.14	63.24	62.95	63.39	63.24	63.24	63.73	63.32	80.36	61.76
Run5	64.14	62.35	63.69	63.39	63.54	63.54	63.24	63.10	63.24	63.10	62.80	63.10	63.47	62.95	63.39	84.08	63.10
Mean	64.61	63.87	63.87	63.87	64.23	63.96	63.54	63.78	63.33	63.30	63.21	63.30	63.83	63.77	63.62	82.14	62.62
SD	0.62	1.01	0.77	0.66	0.70	0.58	0.57	0.75	0.29	0.72	0.57	0.50	0.64	0.76	0.53	1.44	1.05

Appendix B

Selected Models and CFD

Table B.1: The selected models and its CFD for HEST when ensemble size is 3.

	R0		R1		R2	
	Models	CFD	Models	CFD	Models	CFD
Run1	SMO, BayesNet, IBk	58.99	PART, JRip, SMO	66.67	LWL, SMO, JRip	76.97
Run2	SMO, BayesNet, NaiveBayes	50.00	PART, BayesNet, SMO	69.51	LWL, SMO, BayesNet	88.09
Run3	BayesNet, SMO, NaiveBayes	50.00	IBk, PART, BayesNet	66.67	LWL, BayesNet, PART	84.35
Run4	SMO, NaiveBayes, PART	60.00	BayesNet, NaiveBayes, SMO	53.79	LWL, SMO, NaiveBayes	85.40
Run5	NaiveBayes, BayesNet, SMO	49.18	IBk, J48, NaiveBayes	67.42	LWL, NaiveBayes, J48	85.11

Table B.2: The selected models and its CFD for HEST when ensemble size is 5

	R0		R1		R2	
	Models	CFD	Models	CFD	Models	CFD
Run1	SMO, BayesNet, IBk, PART, NaiveBayes	57.62	J48, IBk, PART, JRip, SMO	65.84	BayesNet, LWL, RandomTree, SMO, JRip	79.07
Run2	SMO, BayesNet, NaiveBayes, PART, IBk	69.63	LWL, JRip, PART, BayesNet, SMO	84.49	PART, LWL, NaiveBayes, SMO, BayesNet	85.54
Run3	BayesNet, SMO, NaiveBayes, PART, J48	60.06	SMO, J48, IBk, PART, BayesNet	64.75	RandomTree, SMO, LWL, BayesNet, PART	83.33
Run4	SMO, NaiveBayes, PART, BayesNet, J48	60.06	PART, JRip, BayesNet, NaiveBayes, SMO	67.29	PART, LWL, BayesNet, SMO, NaiveBayes	84.39
Run5	NaiveBayes, BayesNet, SMO, PART, IBk	62.22	PART, REPTree, IBk, J48, NaiveBayes	66.81	LWL, SMO, BayesNet, NaiveBayes, J48	84.24

Table B.3: The selected models and its CFD for HEST when ensemble size is 7

	R0		R1		R2	
	Models	CFD	Models	CFD	Models	CFD
Run1	SMO, BayesNet, IBk, PART, NaiveBayes, J48, JRip	64.20	BayesNet, REPTree, J48, IBk, PART, JRip, SMO	67.34	PART, BayesNet, LWL, IBk, RandomTree, SMO, JRip	79.57
Run2	SMO, BayesNet, NaiveBayes, PART, IBk, J48, REPTree	68.97	IBk, REPTree, LWL, JRip, PART, BayesNet, SMO	83.15	PART, IBk, LWL, NaiveBayes, RandomTree, SMO, BayesNet	84.67
Run3	BayesNet, SMO, NaiveBayes, PART, J48, IBk, JRip	68.00	RandomTree, REPTree, SMO, J48, IBk, PART, BayesNet	74.70	RandomTree, SMO, LWL, J48, NaiveBayes, BayesNet, PART	83.01
Run4	SMO, NaiveBayes, PART, BayesNet, J48, IBk, JRip	70.62	J48, RandomTree, PART, JRip, BayesNet, NaiveBayes, SMO	74.74	PART, LWL, J48, BayesNet, REPTree, SMO, NaiveBayes	83.33
Run5	NaiveBayes, BayesNet, SMO, PART, IBk, J48, REPTree	67.64	LWL, SMO, PART, REPTree, IBk, J48, NaiveBayes	83.01	PART, IBk, LWL, SMO, BayesNet, NaiveBayes, J48	83.67

Table B.4: The selected models and its CFD for HEST when ensemble size is 9

	R0		R1		R2	
	Models	CFD	Models	CFD	Models	CFD
Run1	SMO, BayesNet, IBk, PART, NaiveBayes, J48, JRip, REPTree, RandomTree	64.20	LWL, RandomTree, BayesNet, REPTree, J48, IBk, PART, JRip, SMO	67.34	PART, BayesNet, LWL, IBk, NaiveBayes, RandomTree, REPTree, SMO, JRip	79.57
Run2	SMO, BayesNet, NaiveBayes, PART, IBk, J48, REPTree, JRip, RandomTree	68.97	NaiveBayes, RandomTree, IBk, REPTree, LWL, JRip, PART, BayesNet, SMO	83.15	PART, IBk, LWL, J48, NaiveBayes, RandomTree, REPTree, SMO, BayesNet	84.67
Run3	BayesNet, SMO, NaiveBayes, PART, J48, IBk, JRip, REPTree, RandomTree	68.00	NaiveBayes, LWL, RandomTree, REPTree, SMO, J48, IBk, PART, BayesNet	74.70	RandomTree, SMO, LWL, IBk, J48, NaiveBayes, REPTree, BayesNet, PART	83.01
Run4	SMO, NaiveBayes, PART, BayesNet, J48, IBk, JRip, REPTree, RandomTree	70.62	LWL, REPTree, J48, RandomTree, PART, JRip, BayesNet, NaiveBayes, SMO	74.74	PART, IBk, LWL, J48, BayesNet, RandomTree, REPTree, SMO, NaiveBayes	83.33
Run5	NaiveBayes, BayesNet, SMO, PART, IBk, J48, REPTree, JRip, RandomTree	67.64	JRip, BayesNet, LWL, SMO, PART, REPTree, IBk, J48, NaiveBayes	83.01	PART, IBk, LWL, SMO, BayesNet, RandomTree, REPTree, NaiveBayes, J48	83.67

Table B.5: The selected models and its CFD for HESG when ensemble size is 3

	R0		R1		R2	
	Models	CFD	Models	CFD	Models	CFD
Run1	SMO, BayesNet, NaiveBayes	57.55	REPTree, BayesNet, SMO	71.38	J48, SMO, BayesNet	72.14
Run2	SMO, BayesNet, NaiveBayes	50.78	IBk, BayesNet, SMO	62.55	LWL, SMO, BayesNet	69.55
Run3	SMO, BayesNet, NaiveBayes	51.46	RandomTree, REPTree, SMO	66.81	LWL, SMO, REPTree	66.97
Run4	SMO, BayesNet, NaiveBayes	52.82	BayesNet, RandomTree, SMO	66.99	NaiveBayes, SMO, RandomTree	67.71
Run5	SMO, BayesNet, NaiveBayes	50.27	PART, JRip, SMO	69.62	J48, SMO, JRip	70.40

Table B.6: The selected models and its CFD for HESG when ensemble size is 5

	R0		R1		R2	
	Models	CFD	Models	CFD	Models	CFD
Run1	SMO, BayesNet, NaiveBayes, IBk, PART	63.39	RandomTree, J48, REPTree, BayesNet, SMO	66.09	JRip, J48, NaiveBayes, SMO, BayesNet	68.75
Run2	SMO, BayesNet, NaiveBayes, IBk, J48	64.26	JRip, RandomTree, IBk, BayesNet, SMO	65.26	LWL, J48, NaiveBayes, SMO, BayesNet	66.25
Run3	SMO, BayesNet, NaiveBayes, IBk, PART	61.27	BayesNet, JRip, RandomTree, REPTree, SMO	63.36	BayesNet, LWL, NaiveBayes, SMO, REPTree	65.99
Run4	SMO, BayesNet, NaiveBayes, IBk, PART	63.55	IBk, PART, BayesNet, RandomTree, SMO	63.68	BayesNet, NaiveBayes, JRip, SMO, RandomTree	65.56
Run5	SMO, BayesNet, NaiveBayes, IBk, JRip	66.07	BayesNet, IBk, PART, JRip, SMO	68.12	BayesNet, IBk, J48, SMO, JRip	68.47

Table B.7: The selected models and its CFD for HESG when ensemble size is 7

	R0		R1		R2	
	Models	CFD	Models	CFD	Models	CFD
Run1	SMO, BayesNet, NaiveBayes, IBk, PART, JRip, J48	65.93	JRip, NaiveBayes, RandomTree, J48, REPTree, BayesNet, SMO	65.56	IBk, JRip, J48, NaiveBayes, RandomTree, SMO, BayesNet	66.45
Run2	SMO, BayesNet, NaiveBayes, IBk, J48, JRip, PART	62.45	NaiveBayes, J48, JRip, RandomTree, IBk, BayesNet, SMO	63.76	IBk, LWL, JRip, J48, NaiveBayes, SMO, BayesNet	64.15
Run3	SMO, BayesNet, NaiveBayes, IBk, PART, JRip, REPTree	62.02	J48, PART, BayesNet, JRip, RandomTree, REPTree, SMO	61.17	BayesNet, LWL, IBk, NaiveBayes, JRip, SMO, REPTree	63.76
Run4	SMO, BayesNet, NaiveBayes, IBk, PART, REPTree, JRip	63.20	JRip, REPTree, IBk, PART, BayesNet, RandomTree, SMO	61.88	BayesNet, IBk, NaiveBayes, JRip, REPTree, SMO, RandomTree	64.12
Run5	SMO, BayesNet, NaiveBayes, IBk, JRip, REPTree, PART	66.49	J48, REPTree, BayesNet, IBk, PART, JRip, SMO	65.70	BayesNet, IBk, J48, NaiveBayes, RandomTree, SMO, JRip	67.05

Table B.8: The selected models and its CFD for HESG when ensemble size is 9

	R0		R1		R2	
	Models	CFD	Models	CFD	Models	CFD
Run1	SMO, BayesNet, NaiveBayes, IBk, PART, JRip, J48, REPTree, RandomTree	64.76	IBk, PART, JRip, NaiveBayes, RandomTree, J48, REPTree, BayesNet, SMO	64.76	PART, IBk, JRip, J48, NaiveBayes, RandomTree, REPTree, SMO, BayesNet	64.76
Run2	SMO, BayesNet, NaiveBayes, IBk, J48, JRip, PART, RandomTree, REPTree	60.95	REPTree, PART, NaiveBayes, J48, JRip, RandomTree, IBk, BayesNet, SMO	60.95	PART, IBk, LWL, JRip, J48, NaiveBayes, RandomTree, SMO, BayesNet	61.11
Run3	SMO, BayesNet, NaiveBayes, IBk, PART, JRip, REPTree, J48, RandomTree	61.62	IBk, NaiveBayes, J48, PART, BayesNet, JRip, RandomTree, REPTree, SMO	61.62	RandomTree, BayesNet, LWL, IBk, NaiveBayes, JRip, PART, SMO, REPTree	62.24
Run4	SMO, BayesNet, NaiveBayes, IBk, PART, REPTree, JRip, J48, RandomTree	61.40	J48, NaiveBayes, JRip, REPTree, IBk, PART, BayesNet, RandomTree, SMO	61.40	PART, BayesNet, IBk, J48, NaiveBayes, JRip, REPTree, SMO, RandomTree	61.40
Run5	SMO, BayesNet, NaiveBayes, IBk, JRip, REPTree, PART, J48, RandomTree	65.15	NaiveBayes, RandomTree, J48, REPTree, BayesNet, IBk, PART, JRip, SMO	65.15	PART, BayesNet, IBk, J48, NaiveBayes, RandomTree, REPTree, SMO, JRip	65.15

Table B.9: The selected models and its CFD for FLEM when ensemble size is 3

	R0		R1		R2	
	Models	CFD	Models	CFD	Models	CFD
Run1	SMO, IBk, PART	81.06	NaiveBayes, BayesNet, SMO	72.84	LWL, SMO, BayesNet	85.08
Run2	SMO, PART, IBk	80.95	REPTree, LWL, SMO	80.83	PART, SMO, LWL	86.99
Run3	SMO, PART, J48	69.13	JRip, PART, SMO	78.28	RandomTree, SMO, PART	85.86
Run4	SMO, PART, J48	68.80	J48, LWL, SMO	87.03	J48, SMO, LWL	87.03
Run5	SMO, PART, J48	69.39	J48, PART, SMO	69.39	RandomTree, SMO, PART	87.45

Table B.10: The selected models and its CFD for FLEM when ensemble size is 5

	R0		R1		R2	
	Models	CFD	Models	CFD	Models	CFD
Run1	SMO, IBk, PART, J48, BayesNet	80.77	JRip, RandomTree, NaiveBayes, BayesNet, SMO	79.29	PART, IBk, RandomTree, SMO, BayesNet	83.39
Run2	SMO, PART, IBk, J48, JRip	76.90	BayesNet, PART, REPTree, LWL, SMO	79.58	RandomTree, PART, IBk, SMO, LWL	81.74
Run3	SMO, PART, J48, IBk, JRip	74.17	J48, BayesNet, JRip, PART, SMO	78.98	J48, RandomTree, JRip, SMO, PART	82.82
Run4	SMO, PART, J48, IBk, JRip	76.32	BayesNet, PART, J48, LWL, SMO	81.83	J48, PART, JRip, SMO, LWL	82.64
Run5	SMO, PART, J48, IBk, JRip	77.97	LWL, BayesNet, J48, PART, SMO	82.64	J48, RandomTree, IBk, SMO, PART	84.52

Table B.11: The selected models and its CFD for FLEM when ensemble size is 7

	R0		R1		R2	
	Models	CFD	Models	CFD	Models	CFD
Run1	SMO, IBk, PART, J48, BayesNet, JRip, REPTree	79.02	LWL, J48, JRip, RandomTree, NaiveBayes, BayesNet, SMO	77.27	J48, PART, IBk, REPTree, RandomTree, SMO, BayesNet	81.75
Run2	SMO, PART, IBk, J48, JRip, BayesNet, REPTree	78.47	NaiveBayes, IBk, BayesNet, PART, REPTree, LWL, SMO	76.93	J48, RandomTree, PART, JRip, IBk, SMO, LWL	79.95
Run3	SMO, PART, J48, IBk, JRip, BayesNet, REPTree	77.82	NaiveBayes, REPTree, J48, BayesNet, JRip, PART, SMO	76.63	J48, RandomTree, BayesNet, JRip, IBk, SMO, PART	80.80
Run4	SMO, PART, J48, IBk, JRip, BayesNet, REPTree	78.78	IBk, NaiveBayes, BayesNet, PART, J48, LWL, SMO	78.53	J48, NaiveBayes, PART, JRip, IBk, SMO, LWL	80.83
Run5	SMO, PART, J48, IBk, JRip, BayesNet, REPTree	80.08	IBk, REPTree, LWL, BayesNet, J48, PART, SMO	80.72	J48, RandomTree, REPTree, JRip, IBk, SMO, PART	81.51

Table B.12: The selected models and its CFD for FLEM when ensemble size is 9

	R0		R1		R2	
	Models	CFD	Models	CFD	Models	CFD
Run1	SMO, IBk, PART, J48, BayesNet, JRip, REPTree, NaiveBayes, LWL	78.58	PART, REPTree, LWL, J48, JRip, RandomTree, NaiveBayes, BayesNet, SMO	77.70	J48, PART, IBk, REPTree, NaiveBayes, RandomTree, JRip, SMO, BayesNet	79.61
Run2	SMO, PART, IBk, J48, JRip, BayesNet, REPTree, NaiveBayes, LWL	76.90	RandomTree, J48, NaiveBayes, IBk, BayesNet, PART, REPTree, LWL, SMO	77.13	J48, REPTree, RandomTree, PART, BayesNet, JRip, IBk, SMO, LWL	77.93
Run3	SMO, PART, J48, IBk, JRip, BayesNet, REPTree, NaiveBayes, LWL	76.40	LWL, IBk, NaiveBayes, REPTree, J48, BayesNet, JRip, PART, SMO	76.40	J48, RandomTree, NaiveBayes, REPTree, BayesNet, JRip, IBk, SMO, PART	78.57
Run4	SMO, PART, J48, IBk, JRip, BayesNet, REPTree, NaiveBayes, LWL	78.49	RandomTree, REPTree, IBk, NaiveBayes, BayesNet, PART, J48, LWL, SMO	77.85	J48, REPTree, RandomTree, PART, BayesNet, JRip, IBk, SMO, LWL	79.03
Run5	SMO, PART, J48, IBk, JRip, BayesNet, REPTree, NaiveBayes, LWL	77.67	RandomTree, NaiveBayes, IBk, REPTree, LWL, BayesNet, J48, PART, SMO	76.95	J48, RandomTree, NaiveBayes, REPTree, BayesNet, JRip, IBk, SMO, PART	78.68

Table B.13: The selected models and its CFD for DLEM when ensemble size is 3

	R0		R1		R2	
	Models	CFD	Models	CFD	Models	CFD
Run1	tSMO, tBayesNet, tIBk,	58.99	gSMO, gBayesNet, tSMO,	81.62	tJ48, tSMO, gBayesNet,	86.99
Run2	tSMO, tBayesNet, tNaiveBayes,	50.00	gBayesNet, gIBk, tSMO,	80.21	tBayesNet, tSMO, gIBk,	90.55
Run3	tBayesNet, tSMO, tNaiveBayes,	50.00	gIBk, gSMO, tBayesNet,	76.14	tRandomTree, tBayesNet, gSMO,	86.55
Run4	tSMO, tNaiveBayes, tPART,	60.00	gIBk, gSMO, tSMO,	78.99	tNaiveBayes, tSMO, gSMO,	88.62
Run5	tNaiveBayes, tBayesNet, tSMO,	49.18	gSMO, gBayesNet, tNaiveBayes,	77.42	tJ48, tNaiveBayes, gBayesNet,	88.40

Table B.14: The selected models and its CFD for DLEM when ensemble size is 5

	R0		R1		R2	
	Models	CFD	Models	CFD	Models	CFD
Run1	tSMO, tBayesNet, tIBk, tPART, tNaiveBayes,	57.62	gNaiveBayes, gIBk, gSMO, gBayesNet, tSMO,	70.20	tLWL, tBayesNet, tIBk, tSMO, gBayesNet,	84.74
Run2	tSMO, tBayesNet, tNaiveBayes, tPART, tIBk,	69.63	gSMO, gJRip, gBayesNet, gIBk, tSMO,	72.53	tLWL, tNaiveBayes, tBayesNet, tSMO, gIBk,	86.54
Run3	tBayesNet, tSMO, tNaiveBayes, tPART, tJ48,	60.06	gBayesNet, gREPTree, gIBk, gSMO, tBayesNet,	70.85	tLWL, tNaiveBayes, tSMO, tBayesNet, gSMO,	86.07
Run4	tSMO, tNaiveBayes, tPART, tBayesNet, tJ48,	60.06	gJ48, gBayesNet, gIBk, gSMO, tSMO,	71.79	tLWL, tNaiveBayes, tPART, tSMO, gSMO,	86.03
Run5	tNaiveBayes, tBayesNet, tSMO, tPART, tIBk,	62.22	gNaiveBayes, gIBk, gSMO, gBayesNet, tNaiveBayes,	70.08	tLWL, tSMO, tBayesNet, tNaiveBayes, gBayesNet,	87.34

Table B.15: The selected models and its CFD for DLEM when ensemble size is 7

	R0		R1		R2	
	Models	CFD	Models	CFD	Models	CFD
Run1	tSMO, tBayesNet, tIBk, tPART, tNaiveBayes, tJ48, tJRip,	64.20	gJ48, gJRip, gNaiveBayes, gIBk, gSMO, gBayesNet, tSMO,	72.22	gLWL, tPART, tBayesNet, tJ48, tIBk, tSMO, gBayesNet,	84.23
Run2	tSMO, tBayesNet, tNaiveBayes, tPART, tIBk, tJ48, tREPTree,	68.97	gJ48, gNaiveBayes, gSMO, gJRip, gBayesNet, gIBk, tSMO,	70.16	tLWL, tNaiveBayes, tPART, tBayesNet, tIBk, tSMO, gIBk,	86.43
Run3	tBayesNet, tSMO, tNaiveBayes, tPART, tJ48, tIBk, tJRip,	68.00	gPART, gNaiveBayes, gBayesNet, gREPTree, gIBk, gSMO, tBayesNet,	67.81	tNaiveBayes, gLWL, tPART, tSMO, tJ48, tBayesNet, gSMO,	86.33
Run4	tSMO, tNaiveBayes, tPART, tBayesNet, tJ48, tIBk, tJRip,	70.62	gPART, gNaiveBayes, gJ48, gBayesNet, gIBk, gSMO, tSMO,	69.43	tNaiveBayes, gLWL, tPART, tBayesNet, tJ48, tSMO, gSMO,	86.40
Run5	tNaiveBayes, tBayesNet, tSMO, tPART, tIBk, tJ48, tREPTree,	67.64	gREPTree, gPART, gNaiveBayes, gIBk, gSMO, gBayesNet, tNaiveBayes,	70.48	tLWL, tSMO, tPART, tBayesNet, tIBk, tNaiveBayes, gBayesNet,	86.63

Table B.16: The selected models and its CFD for DLEM when ensemble size is 9

	R0		R1		R2	
	Models	CFD	Models	CFD	Models	CFD
Run1	tSMO, tBayesNet, tIBk, tPART, tNaiveBayes, tJ48, tJRip, tREPTree, tRandomTree,	72.38	gRandomTree, gPART, gJ48, gJRip, gNaiveBayes, gIBk, gSMO, gBayesNet, tSMO,	69.16	tJRip, tNaiveBayes, gLWL, tPART, tBayesNet, tJ48, tIBk, tSMO, gBayesNet,	83.88
Run2	tSMO, tBayesNet, tNaiveBayes, tPART, tIBk, tJ48, tREPTree, tJRip, tRandomTree,	73.88	tBayesNet, gREPTree, gJ48, gNaiveBayes, gSMO, gJRip, gBayesNet, gIBk, tSMO,	71.33	tLWL, tNaiveBayes, tPART, tBayesNet, tJ48, tIBk, tREPTree, tSMO, gIBk,	85.48
Run3	tBayesNet, tSMO, tNaiveBayes, tPART, tJ48, tIBk, tJRip, tREPTree, tRandomTree,	75.27	gJRip, gJ48, gPART, gNaiveBayes, gBayesNet, gREPTree, gIBk, gSMO, tBayesNet,	65.86	tNaiveBayes, gLWL, tPART, tSMO, tJ48, tIBk, tREPTree, tBayesNet, gSMO,	85.79
Run4	tSMO, tNaiveBayes, tPART, tBayesNet, tJ48, tIBk, tJRip, tREPTree, tRandomTree,	74.86	gJRip, gRandomTree, gPART, gNaiveBayes, gJ48, gBayesNet, gIBk, gSMO, tSMO,	66.31	tNaiveBayes, gLWL, tPART, tBayesNet, tJ48, tIBk, tREPTree, tSMO, gSMO,	85.60
Run5	tNaiveBayes, tBayesNet, tSMO, tPART, tIBk, tJ48, tREPTree, tJRip, tRandomTree,	72.31	tJ48, gJ48, gREPTree, gPART, gNaiveBayes, gIBk, gSMO, gBayesNet, tNaiveBayes,	72.35	tSMO, gLWL, tPART, tBayesNet, tJ48, tIBk, tREPTree, tNaiveBayes, gBayesNet,	85.95

Table B.17: The selected models and its CFD for DLEM when ensemble size is 11

	R0		R1		R2	
	Models	CFD	Models	CFD	Models	CFD
Run1	tSMO, tBayesNet, tIBk, tPART, tNaiveBayes, tJ48, tJRip, tREPTree, tRandomTree, gSMO, gBayesNet,	81.78	tJRip, gREPTree, gRandomTree, gPART, gJ48, gJRip, gNaiveBayes, gIBk, gSMO, gBayesNet, tSMO,	70.10	tJRip, tNaiveBayes, gLWL, tPART, tBayesNet, tJ48, tIBk, tRandomTree, tREPTree, tSMO, gBayesNet,	83.56
Run2	tSMO, tBayesNet, tNaiveBayes, tPART, tIBk, tJ48, tREPTree, tJRip, tRandomTree, gSMO, gBayesNet,	81.13	gPART, gLWL, tBayesNet, gREPTree, gJ48, gNaiveBayes, gSMO, gJRip, gBayesNet, gIBk, tSMO,	67.35	tLWL, tNaiveBayes, tPART, tBayesNet, gSMO, tJ48, tIBk, tRandomTree, tREPTree, tSMO, gIBk,	84.96
Run3	tBayesNet, tSMO, tNaiveBayes, tPART, tJ48, tIBk, tJRip, tREPTree, tRandomTree, gSMO, gBayesNet,	82.32	tIBk, tPART, gJRip, gJ48, gPART, gNaiveBayes, gBayesNet, gREPTree, gIBk, gSMO, tBayesNet,	70.78	tJRip, tNaiveBayes, gLWL, tPART, tSMO, tJ48, tIBk, tRandomTree, tREPTree, tBayesNet, gSMO,	84.86
Run4	tSMO, tNaiveBayes, tPART, tBayesNet, tJ48, tIBk, tJRip, tREPTree, tRandomTree, gSMO, gBayesNet,	81.61	gLWL, tNaiveBayes, gJRip, gRandomTree, gPART, gNaiveBayes, gJ48, gBayesNet, gIBk, gSMO, tSMO,	67.48	tJRip, tNaiveBayes, gLWL, tPART, tBayesNet, tJ48, tIBk, tRandomTree, tREPTree, tSMO, gSMO,	84.44
Run5	tNaiveBayes, tBayesNet, tSMO, tPART, tIBk, tJ48, tREPTree, tJRip, tRandomTree, gSMO, gBayesNet,	80.95	gJRip, gRandomTree, tJ48, gJ48, gREPTree, gPART, gNaiveBayes, gIBk, gSMO, gBayesNet, tNaiveBayes,	70.70	tLWL, tSMO, tPART, tBayesNet, tJ48, tIBk, tRandomTree, tREPTree, gIBk, tNaiveBayes, gBayesNet,	85.07

Table B.18: The selected models and its CFD for DLEM when ensemble size is 13

	R0		R1		R2	
	Models	CFD	Models	CFD	Models	CFD
Run1	tSMO, tBayesNet, tIBk, tPART, tNaiveBayes, tJ48, tJRip, tREPTree, tRandomTree, gSMO, gBayesNet, gNaiveBayes, gIBk,	80.45	tPART, gLWL, tJRip, gREPTree, gRandomTree, gPART, gJ48, gJRip, gNaiveBayes, gIBk, gSMO, gBayesNet, tSMO,	70.03	tLWL, tJRip, tNaiveBayes, tPART, tBayesNet, gSMO, gJ48, tJ48, tIBk, tRandomTree, tREPTree, tSMO, gBayesNet,	82.74
Run2	tSMO, tBayesNet, tNaiveBayes, tPART, tIBk, tJ48, tREPTree, tJRip, tRandomTree, gSMO, gBayesNet, gNaiveBayes, gIBk,	80.63	gRandomTree, tPART, gPART, gLWL, tBayesNet, gREPTree, gJ48, gNaiveBayes, gSMO, gJRip, gBayesNet, gIBk, tSMO,	67.89	tLWL, tJRip, tNaiveBayes, tPART, tBayesNet, gSMO, tJ48, tIBk, tRandomTree, tREPTree, gBayesNet, tSMO, gIBk,	83.79
Run3	tBayesNet, tSMO, tNaiveBayes, tPART, tJ48, tIBk, tJRip, tREPTree, tRandomTree, gSMO, gBayesNet, gNaiveBayes, gIBk,	80.46	tSMO, tJ48, tIBk, tPART, gJRip, gJ48, gPART, gNaiveBayes, gBayesNet, gREPTree, gIBk, gSMO, tBayesNet,	73.99	tLWL, tJRip, gNaiveBayes, tNaiveBayes, gLWL, tPART, tSMO, tJ48, tIBk, tRandomTree, tREPTree, tBayesNet, gSMO,	83.56
Run4	tSMO, tNaiveBayes, tPART, tBayesNet, tJ48, tIBk, tJRip, tREPTree, tRandomTree, gSMO, gBayesNet, gNaiveBayes, gIBk,	80.50	tBayesNet, gREPTree, gLWL, tNaiveBayes, gJRip, gRandomTree, gPART, gNaiveBayes, gJ48, gBayesNet, gIBk, gSMO, tSMO,	68.26	tLWL, tJRip, tNaiveBayes, tPART, tBayesNet, gBayesNet, tJ48, tIBk, tRandomTree, tREPTree, gIBk, tSMO, gSMO,	83.06
Run5	tNaiveBayes, tBayesNet, tSMO, tPART, tIBk, tJ48, tREPTree, tJRip, tRandomTree, gSMO, gBayesNet, gNaiveBayes, gIBk,	80.57	tREPTree, tIBk, gJRip, gRandomTree, tJ48, gJ48, gREPTree, gPART, gNaiveBayes, gIBk, gSMO, gBayesNet, tNaiveBayes,	73.58	tLWL, tJRip, tSMO, gLWL, tPART, tBayesNet, gSMO, tJ48, tIBk, tRandomTree, tREPTree, tNaiveBayes, gBayesNet,	84.08

Table B.19: The selected models and its CFD for DLEM when ensemble size is 15

	R0		R1		R2	
	Models	CFD	Models	CFD	Models	CFD
Run1	tSMO, tBayesNet, tIBk, tPART, tNaiveBayes, tJ48, tJRip, tREPTree, tRandomTree, gSMO, gBayesNet, gNaiveBayes, gIBk, gPART, gJRip,	79.75	tJ48, tIBk, tPART, gLWL, tJRip, gREPTree, gRandomTree, gPART, gJ48, gJRip, gNaiveBayes, gIBk, gSMO, gBayesNet, tSMO,	72.37	tPART, tBayesNet, gJ48, tJ48, tLWL, tJRip, gNaiveBayes, tNaiveBayes, gSMO, tIBk, tRandomTree, tREPTree, gIBk, tSMO, gBayesNet,	81.15
Run2	tSMO, tBayesNet, tNaiveBayes, tPART, tIBk, tJ48, tREPTree, tJRip, tRandomTree, gSMO, gBayesNet, gNaiveBayes, gIBk, gJ48, tLWL,	81.37	tLWL, tJRip, gRandomTree, tPART, gPART, gLWL, tBayesNet, gREPTree, gJ48, gNaiveBayes, gSMO, gJRip, gBayesNet, gIBk, tSMO,	70.29	tPART, tBayesNet, tJ48, tLWL, tJRip, gNaiveBayes, tNaiveBayes, gSMO, gREPTree, tIBk, tRandomTree, tREPTree, gBayesNet, tSMO, gIBk,	81.61
Run3	tBayesNet, tSMO, tNaiveBayes, tPART, tJ48, tIBk, tJRip, tREPTree, tRandomTree, gSMO, gBayesNet, gNaiveBayes, gIBk, tLWL, gPART,	81.03	tRandomTree, tREPTree, tSMO, tJ48, tIBk, tPART, gJRip, gJ48, gPART, gNaiveBayes, gBayesNet, gREPTree, gIBk, gSMO, tBayesNet,	75.73	gLWL, tPART, tSMO, tJ48, tLWL, tJRip, gNaiveBayes, tNaiveBayes, gBayesNet, tIBk, tRandomTree, tREPTree, gIBk, tBayesNet, gSMO,	81.24
Run4	tSMO, tNaiveBayes, tPART, tBayesNet, tJ48, tIBk, tJRip, tREPTree, tRandomTree, gSMO, gBayesNet, gNaiveBayes, gIBk, tLWL, gPART,	80.79	tPART, tJRip, tBayesNet, gREPTree, gLWL, tNaiveBayes, gJRip, gRandomTree, gPART, gNaiveBayes, gJ48, gBayesNet, gIBk, gSMO, tSMO,	70.99	tPART, tBayesNet, tJ48, gRandomTree, tLWL, tJRip, gNaiveBayes, tNaiveBayes, gBayesNet, tIBk, tRandomTree, tREPTree, gIBk, tSMO, gSMO,	80.84
Run5	tNaiveBayes, tBayesNet, tSMO, tPART, tIBk, tJ48, tREPTree, tJRip, tRandomTree, gSMO, gBayesNet, gNaiveBayes, gIBk, gJRip, gREPTree,	80.43	tSMO, tPART, tREPTree, tIBk, gJRip, gRandomTree, tJ48, gJ48, gREPTree, gPART, gNaiveBayes, gIBk, gSMO, gBayesNet, tNaiveBayes,	75.88	tPART, tBayesNet, tJ48, tLWL, tJRip, gJRip, gNaiveBayes, tSMO, gSMO, tIBk, tRandomTree, tREPTree, gIBk, tNaiveBayes, gBayesNet,	82.33

Table B.20: The selected models and its CFD for DLEM when ensemble size is 17

	R0		R1		R2	
	Models	CFD	Models	CFD	Models	CFD
Run1	tSMO, tBayesNet, tIBk, tPART, tNaiveBayes, tJ48, tJRip, tREPTree, tRandomTree, gSMO, gBayesNet, gNaiveBayes, gIBk, gPART, gJRip, tLWL, gJ48,	79.19	tBayesNet, tREPTree, tJ48, tIBk, tPART, gLWL, tJRip, gREPTree, gRandomTree, gPART, gJ48, gJRip, gNaiveBayes, gIBk, gSMO, gBayesNet, tSMO,	74.01	tPART, tBayesNet, gPART, gJ48, tJ48, tLWL, tJRip, gJRip, gNaiveBayes, tNaiveBayes, gSMO, tIBk, tRandomTree, tREPTree, gIBk, tSMO, gBayesNet,	79.19
Run2	tSMO, tBayesNet, tNaiveBayes, tPART, tIBk, tJ48, tREPTree, tJRip, tRandomTree, gSMO, gBayesNet, gNaiveBayes, gIBk, gJ48, tLWL, gJRip, gPART,	78.71	tIBk, tREPTree, tLWL, tJRip, gRandomTree, tPART, gPART, gLWL, tBayesNet, gREPTree, gJ48, gNaiveBayes, gSMO, gJRip, gBayesNet, gIBk, tSMO,	72.39	tPART, tBayesNet, gPART, tJ48, tLWL, tJRip, gJRip, gNaiveBayes, tNaiveBayes, gSMO, gREPTree, tIBk, tRandomTree, tREPTree, gBayesNet, tSMO, gIBk,	78.88
Run3	tBayesNet, tSMO, tNaiveBayes, tPART, tJ48, tIBk, tJRip, tREPTree, tRandomTree, gSMO, gBayesNet, gNaiveBayes, gIBk, tLWL, gPART, gJRip, gREPTree,	78.48	tLWL, gRandomTree, tRandomTree, tREPTree, tSMO, tJ48, tIBk, tPART, gJRip, gJ48, gPART, gNaiveBayes, gBayesNet, gREPTree, gIBk, gSMO, tBayesNet,	74.90	tPART, tSMO, gPART, gJ48, tJ48, tLWL, tJRip, gNaiveBayes, tNaiveBayes, gBayesNet, gREPTree, tIBk, tRandomTree, tREPTree, gIBk, tBayesNet, gSMO,	78.97
Run4	tSMO, tNaiveBayes, tPART, tBayesNet, tJ48, tIBk, tJRip, tREPTree, tRandomTree, gSMO, gBayesNet, gNaiveBayes, gIBk, tLWL, gPART, gREPTree, gJRip,	77.94	tJ48, tRandomTree, tPART, tJRip, tBayesNet, gREPTree, gLWL, tNaiveBayes, gJRip, gRandomTree, gPART, gNaiveBayes, gJ48, gBayesNet, gIBk, gSMO, tSMO,	72.88	tPART, tBayesNet, gPART, tJ48, gRandomTree, tLWL, tJRip, gNaiveBayes, tNaiveBayes, gBayesNet, gREPTree, tIBk, tRandomTree, tREPTree, gIBk, tSMO, gSMO,	78.06
Run5	tNaiveBayes, tBayesNet, tSMO, tPART, tIBk, tJ48, tREPTree, tJRip, tRandomTree, gSMO, gBayesNet, gNaiveBayes, gIBk, gJRip, gREPTree, gPART, tLWL,	80.30	gLWL, tLWL, tSMO, tPART, tREPTree, tIBk, gJRip, gRandomTree, tJ48, gJ48, gREPTree, gPART, gNaiveBayes, gIBk, gSMO, gBayesNet, tNaiveBayes,	74.51	tPART, tBayesNet, gPART, tJ48, tLWL, tJRip, gJRip, gNaiveBayes, tSMO, gSMO, gREPTree, tIBk, tRandomTree, tREPTree, gIBk, tNaiveBayes, gBayesNet,	80.30

Table B.21: The selected models and its CFD for DLEM when ensemble size is 19

	R0		R1		R2	
	Models	CFD	Models	CFD	Models	CFD
Run1	tSMO, tBayesNet, tIBk, tPART, tNaiveBayes, tJ48, tJRip, tREPTree, tRandomTree, gSMO, gBayesNet, gNaiveBayes, gIBk, gPART, gJRip, tLWL, gJ48, gREPTree, gRandomTree,	76.75	tLWL, tRandomTree, tBayesNet, tREPTree, tJ48, tIBk, tPART, gLWL, tJRip, gREPTree, gRandomTree, gPART, gJ48, gJRip, gNaiveBayes, gIBk, gSMO, gBayesNet, tSMO,	74.79	tPART, tBayesNet, gPART, gJ48, tJ48, gRandomTree, tLWL, tJRip, gJRip, gNaiveBayes, tNaiveBayes, gSMO, gREPTree, tIBk, tRandomTree, tREPTree, gIBk, tSMO, gBayesNet,	79.19
Run2	tSMO, tBayesNet, tNaiveBayes, tPART, tIBk, tJ48, tREPTree, tJRip, tRandomTree, gSMO, gBayesNet, gNaiveBayes, gIBk, gJ48, tLWL, gJRip, gPART, gRandomTree, gREPTree,	76.11	tNaiveBayes, tRandomTree, tIBk, tREPTree, tLWL, tJRip, gRandomTree, tPART, gPART, gLWL, tBayesNet, gREPTree, gJ48, gNaiveBayes, gSMO, gJRip, gBayesNet, gIBk, tSMO,	73.97	tPART, tBayesNet, gPART, gJ48, tJ48, gRandomTree, tLWL, tJRip, gJRip, gNaiveBayes, tNaiveBayes, gSMO, gREPTree, tIBk, tRandomTree, tREPTree, gBayesNet, tSMO, gIBk,	78.88
Run3	tBayesNet, tSMO, tNaiveBayes, tPART, tJ48, tIBk, tJRip, tREPTree, tRandomTree, gSMO, gBayesNet, gNaiveBayes, gIBk, tLWL, gPART, gJRip, gREPTree, gJ48, gRandomTree,	76.18	tNaiveBayes, gLWL, tLWL, gRandomTree, tRandomTree, tREPTree, tSMO, tJ48, tIBk, tPART, gJRip, gJ48, gPART, gNaiveBayes, gBayesNet, gREPTree, gIBk, gSMO, tBayesNet,	74.78	gLWL, tPART, tSMO, gPART, gJ48, tJ48, tLWL, tJRip, gJRip, gNaiveBayes, tNaiveBayes, gBayesNet, gREPTree, tIBk, tRandomTree, tREPTree, gIBk, tBayesNet, gSMO,	78.97
Run4	tSMO, tNaiveBayes, tPART, tBayesNet, tJ48, tIBk, tJRip, tREPTree, tRandomTree, gSMO, gBayesNet, gNaiveBayes, gIBk, tLWL, gPART, gREPTree, gJRip, gJ48, gRandomTree,	75.34	tLWL, tREPTree, tJ48, tRandomTree, tPART, tJRip, tBayesNet, gREPTree, gLWL, tNaiveBayes, gJRip, gRandomTree, gPART, gNaiveBayes, gJ48, gBayesNet, gIBk, gSMO, tSMO,	73.30	tPART, tBayesNet, gPART, gJ48, tJ48, gRandomTree, tLWL, tJRip, gJRip, gNaiveBayes, tNaiveBayes, gBayesNet, gREPTree, tIBk, tRandomTree, tREPTree, gIBk, tSMO, gSMO,	78.06
Run5	tNaiveBayes, tBayesNet, tSMO, tPART, tIBk, tJ48, tREPTree, tJRip, tRandomTree, gSMO, gBayesNet, gNaiveBayes, gIBk, gJRip, gREPTree, gPART, tLWL, gJ48, gRandomTree,	77.98	tJRip, tBayesNet, gLWL, tLWL, tSMO, tPART, tREPTree, tIBk, gJRip, gRandomTree, tJ48, gJ48, gREPTree, gPART, gNaiveBayes, gIBk, gSMO, gBayesNet, tNaiveBayes,	75.91	tPART, tBayesNet, gPART, gJ48, tJ48, gRandomTree, tLWL, tJRip, gJRip, gNaiveBayes, tSMO, gSMO, gREPTree, tIBk, tRandomTree, tREPTree, gIBk, tNaiveBayes, gBayesNet,	80.30

Appendix C

All Results Obtained by HES, FLEM and DLEM

In this section we show all obtained results in HES, FLEM and DLEM experiments. Each sub-figure shows the results obtained with different conflagrations which are the method used for the ensemble and the number of models involved on the ensemble.

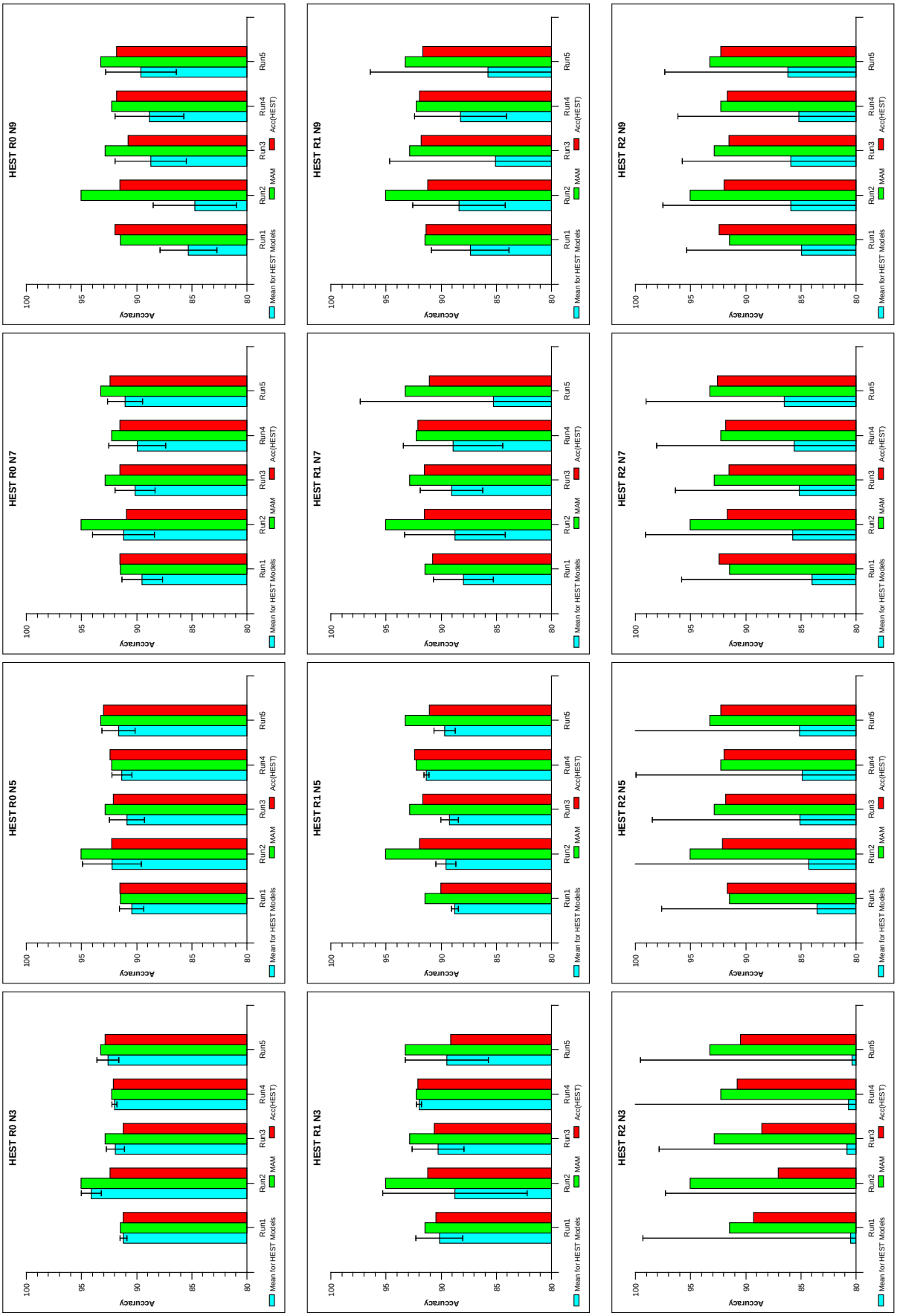

Figure C.1: The HEST results for rules R0, R1 and R3; and ensemble sizes 3, 5, 7 and 9. The blue bar represents the mean accuracy of the HEST models, the green bar represents the most accurate model and the red bar represents HEST accuracy.

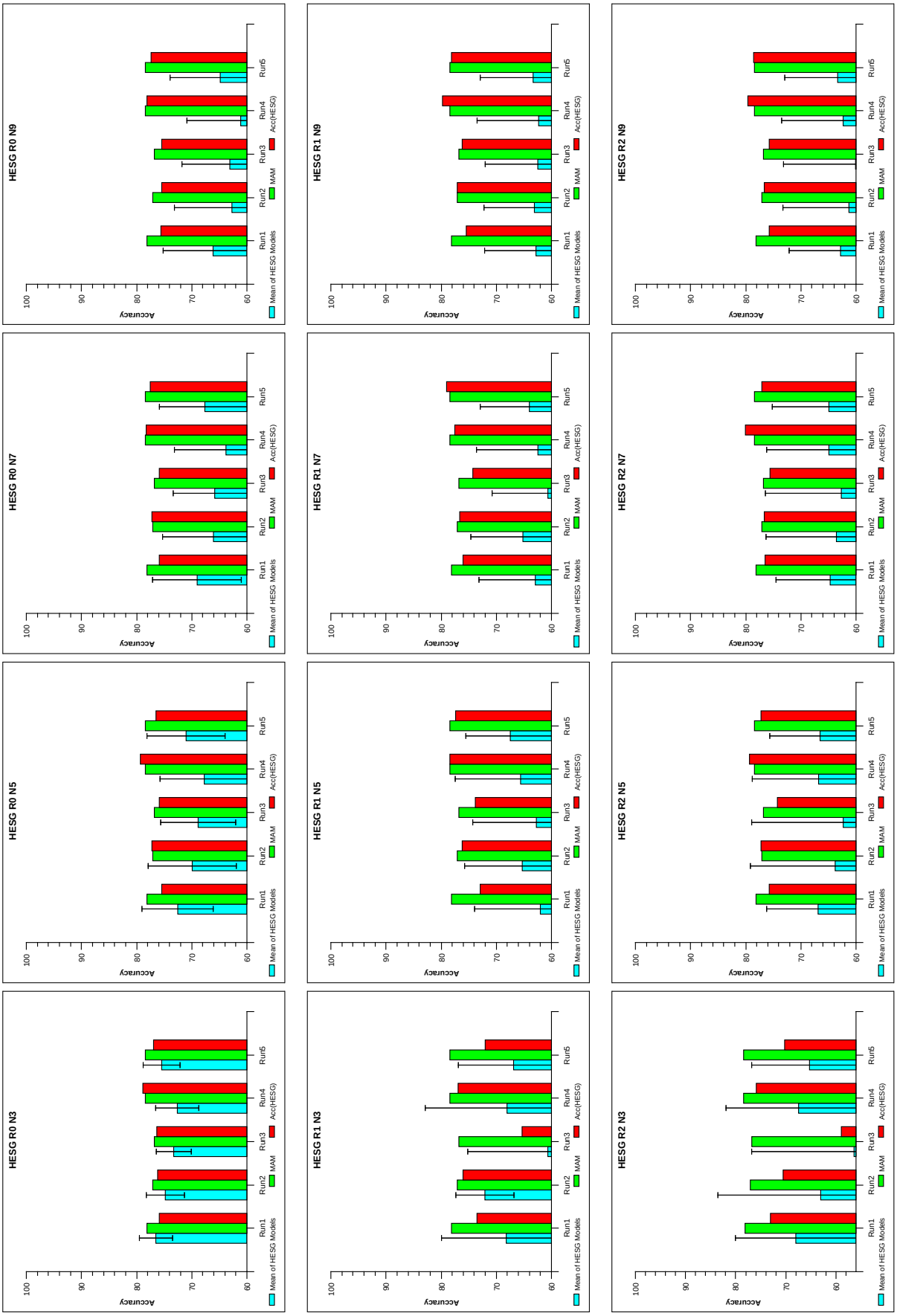

Figure C.2: The HESG results for rules R0, R1 and R3; and ensemble sizes 3, 5, 7 and 9. The blue bar represents the mean accuracy of the HESG models, the green bar represents the most accurate model and the red bar represents HESG accuracy.

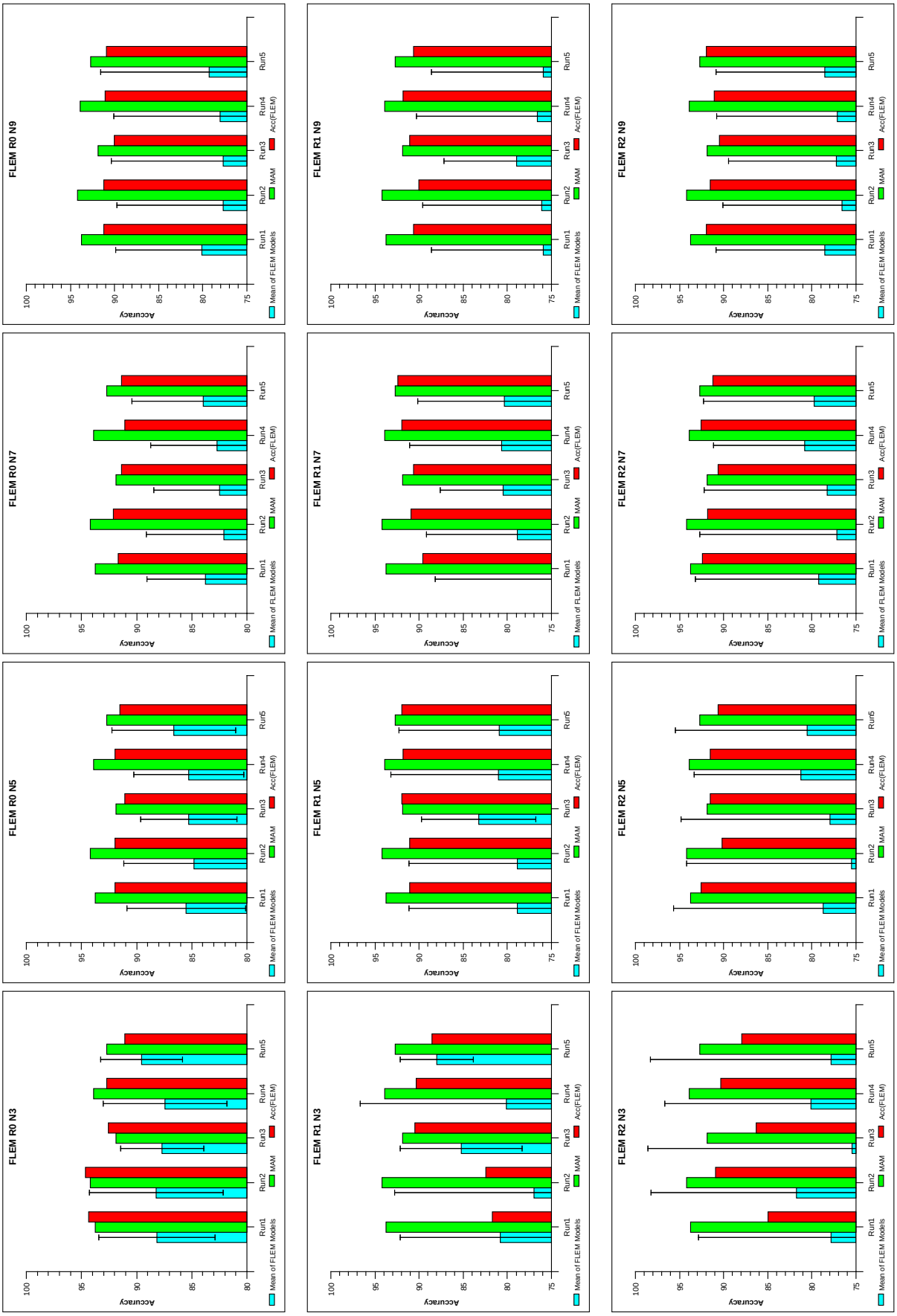

Figure C.3: The FLEM results for rules R0, R1 and R3; and ensemble sizes 3, 5, 7 and 9. The blue bar represents the mean accuracy of the FLEM models, the green bar represents the most accurate model and the red bar represents FLEM accuracy.

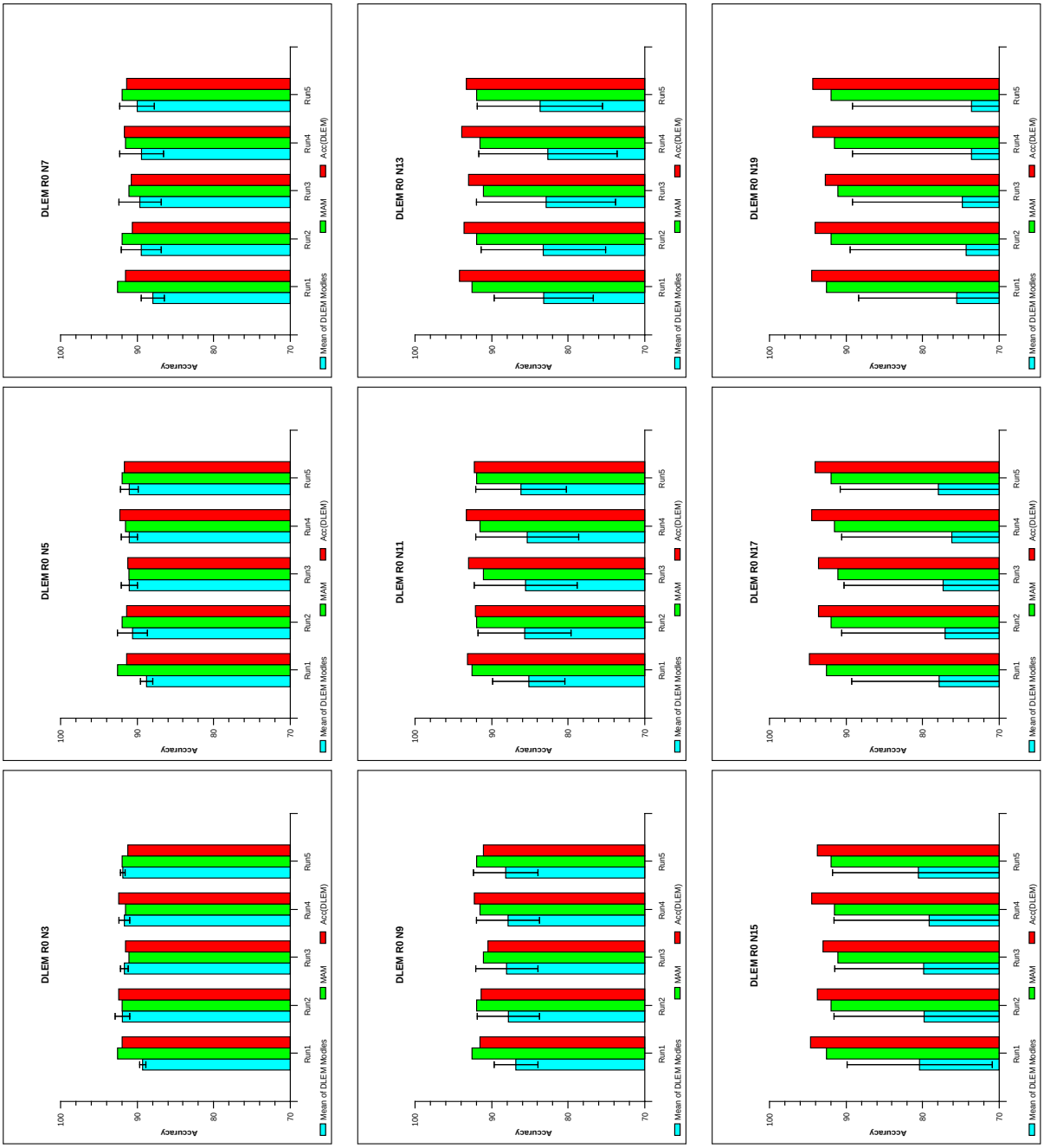

Figure C.4: The DLEM results for rule R0 and ensemble sizes 3, 5, 7, 9, 11, 13, 15, 17 and 19. The blue bar represents the mean accuracy of the DLEM models, the green bar represents the most accurate model and the red bar represents DLEM accuracy.

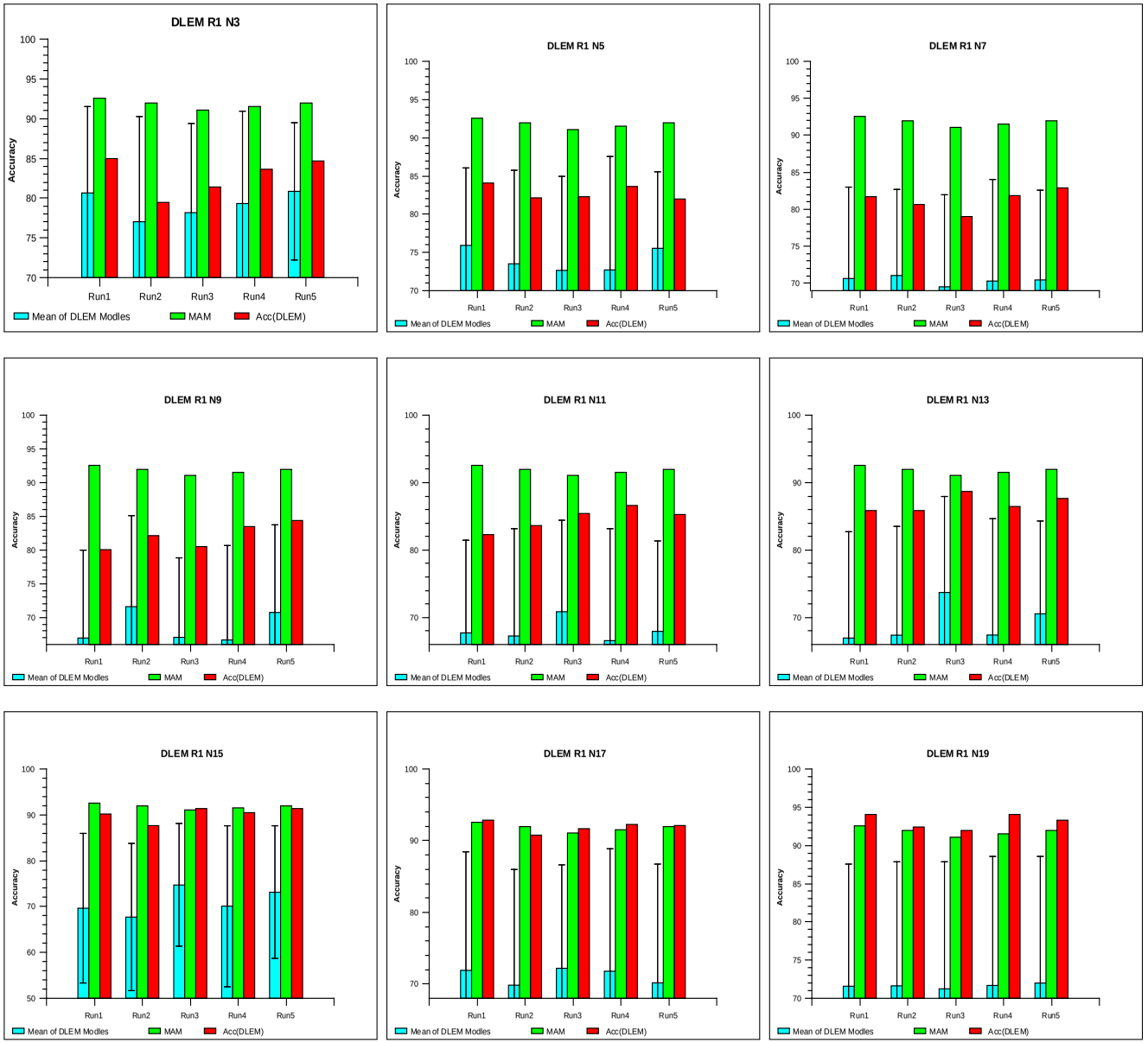

Figure C.5: The DLEM results for rule R1 and ensemble sizes 3, 5, 7, 9, 11, 13, 15, 17 and 19. The blue bar represents the mean accuracy of the DLEM models, the green bar represents the most accurate model and the red bar represents DLEM accuracy.

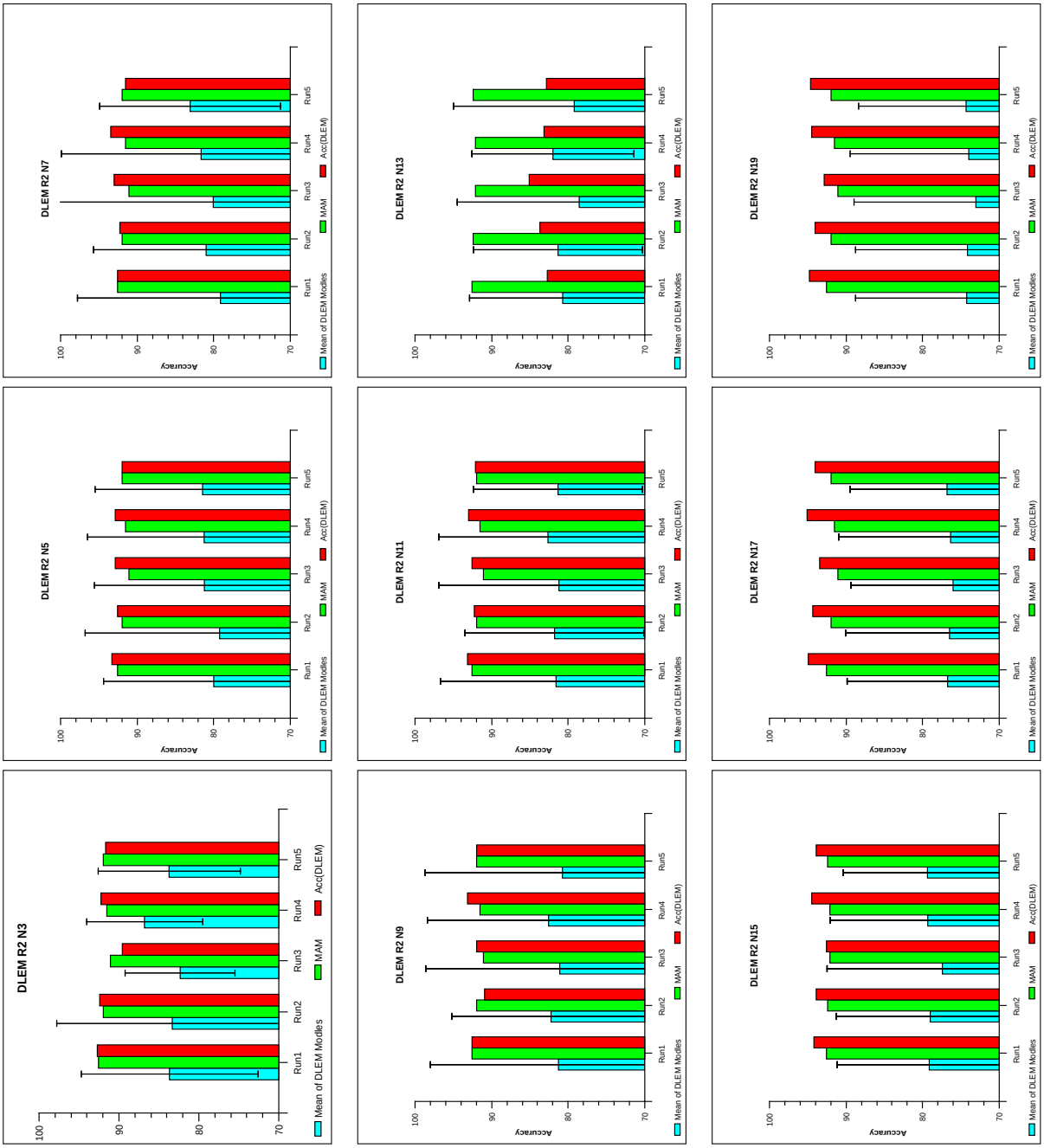

Figure C.6: The DLEM results for rule R2 and ensemble sizes 3, 5, 7, 9, 11, 13, 15, 17 and 19. The blue bar represents the mean accuracy of the DLEM models, the green bar represents the most accurate model and the red bar represents DLEM accuracy.