AN IMPROVED MULTIPLE CLASSIFIER COMBINATION SCHEME FOR PATTERN CLASSIFICATION

ABDULLAH

DOCTOR OF PHILOSOPHY
UNIVERSITI UTARA MALAYSIA
2015

Awang Had Salleh
Graduate School
of Arts And Sciences

Universiti Utara Malaysia

PERAKUAN KERJA TESIS / DISERTASI
(Certification of thesis / dissertation)

Kami, yang bertandatangan, memperakukan bahawa
(We, the undersigned, certify that)

ABDULLAH

calon untuk Ijazah **PhD**
(candidate for the degree of)

telah mengemukakan tesis / disertasi yang bertajuk:
(has presented his/her thesis / dissertation of the following title):

"AN IMPROVED MULTIPLE CLASSIFIER COMBINATION SCHEME FOR PATTERN CLASSIFICATION"

seperti yang tercatat di muka surat tajuk dan kulit tesis / disertasi.
(as it appears on the title page and front cover of the thesis / dissertation).

Bahawa tesis/disertasi tersebut boleh diterima dari segi bentuk serta kandungan dan meliputi bidang ilmu dengan memuaskan, sebagaimana yang ditunjukkan oleh calon dalam ujian lisan yang diadakan pada : ***18 Jun 2014.***
That the said thesis/dissertation is acceptable in form and content and displays a satisfactory knowledge of the field of study as demonstrated by the candidate through an oral examination held on:
June 18, 2014.

Pengerusi Viva: *(Chairman for VIVA)*	**Assoc. Prof. Dr. Huda Haji Ibrahim**	Tandatangan *(Signature)*
Pemeriksa Luar: *(External Examiner)*	**Prof. Dr. Siti Mariyam Hj Shamsuddin**	Tandatangan *(Signature)*
Pemeriksa Dalam: *(Internal Examiner)*	**Dr. Shahrul Azmi Mohd Yusof**	Tandatangan *(Signature)*
Nama Penyelia/Penyelia-penyelia: *(Name of Supervisor/Supervisors)*	**Prof. Dr. Ku Ruhana Ku Mahamud**	Tandatangan *(Signature)*

Tarikh:
(Date) ***June 18, 2014***

Permission to Use

In presenting this thesis in fulfilment of the requirements for a postgraduate degree from Universiti Utara Malaysia, I agree that the Universiti Library may make it freely available for inspection. I further agree that permission for the copying of this thesis in any manner, in whole or in part, for scholarly purpose may be granted by my supervisor(s) or, in their absence, by the Dean of Awang Had Salleh Graduate School of Arts and Sciences. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to Universiti Utara Malaysia for any scholarly use which may be made of any material from my thesis.

Requests for permission to copy or to make other use of materials in this thesis, in whole or in part, should be addressed to:

Dean of Awang Had Salleh Graduate School of Arts and Sciences
UUM College of Arts and Sciences
Universiti Utara Malaysia
06010 UUM Sintok

Abstrak

Gabungan pengelas berganda dianggap sebagai satu arah baru dalam bidang pengecaman corak untuk meningkatkan prestasi pengelasan. Ketiadaan garis panduan piawai untuk membangunankan pengelas gabung yang tepat dan pelbagai merupakan masalah utama dalam gabungan pengelas berganda. Ini adalah kerana kesukaran untuk mengenal pasti jumlah pengelas homogen dan bagaimana menggabungkan hasil pengelas. Kaedah gabung yang paling biasa digunakan ialah strategi rawak manakala teknik pengundian terbanyak digunakan sebagai penggabung pengelas. Walau bagaimanapun, strategi rawak tidak dapat menentukan bilangan pengelas dan pengundian terbanyak tidak mempertimbangkan kekuatan setiap pengelas, sehingga menyebabkan ketepatan pengelasan yang rendah. Dalam kajian ini, satu skim gabungan pengelas berganda yang lebih baik dicadangkan. Algoritma *ant system* (AS) digunakan untuk melakukan sesekat set ciri dalam pembentukan subset ciri yang mewakili pengelas. Satu ukuran kekompakan diperkenalkan sebagai satu parameter dalam membina pengelas gabung yang tepat dan beragam. Satu kaedah mengundi pemberat digunakan untuk menggabungkan hasil pengelas dengan mempertimbangkan kekuatan pengelas sebelum pengundian dilakukan. Eksperimen telah dijalankan menggunakan empat pengelas asas iaitu *nearest mean classifier* (NMC), *naive bayes classifier* (NBC), *k-nearest neighbour* (*k*-NN) dan *linear discrimimant analisis* (LDA) ke atas set data penanda aras, untuk menguji kredibiliti skim gabungan pengelas berganda yang dicadangkan. Purata ketepatan pengelas gabung homogen NMC, NBC, k- NN dan LDA adalah 97,91 %, 98,06 %, 98.09 % dan 98,12 %. Ketepatan adalah lebih tinggi daripada yang diperolehi melalui penggunaan kaedah lain dalam membangunkan gabungan pengelas berganda. Skim gabungan pengelas berganda yang dicadangkan dapat membantu dalam membangunkan gabungan pengelas berganda untuk pengecaman dan pengelasan corak yang lain.

Kata Kunci: Gabungan pengelas berganda, Ukuran keragaman, Pengecaman dan pengelasan corak, Algoritma *ant system*, Pengundian berberat.

Abstract

Combining multiple classifiers are considered as a new direction in the pattern recognition to improve classification performance. The main problem of multiple classifier combination is that there is no standard guideline for constructing an accurate and diverse classifier ensemble. This is due to the difficulty in identifying the number of homogeneous classifiers and how to combine the classifier outputs. The most commonly used ensemble method is the random strategy while the majority voting technique is used as the combiner. However, the random strategy cannot determine the number of classifiers and the majority voting technique does not consider the strength of each classifier, thus resulting in low classification accuracy. In this study, an improved multiple classifier combination scheme is proposed. The ant system (AS) algorithm is used to partition feature set in developing feature subsets which represent the number of classifiers. A compactness measure is introduced as a parameter in constructing an accurate and diverse classifier ensemble. A weighted voting technique is used to combine the classifier outputs by considering the strength of the classifiers prior to voting. Experiments were performed using four base classifiers, which are Nearest Mean Classifier (NMC), Naive Bayes Classifier (NBC), *k*-Nearest Neighbour (*k*-NN) and Linear Discriminant Analysis (LDA) on benchmark datasets, to test the credibility of the proposed multiple classifier combination scheme. The average classification accuracy of the homogeneous NMC, NBC, *k*-NN and LDA ensembles are 97.91%, 98.06%, 98.09% and 98.12% respectively. The accuracies are higher than those obtained through the use of other approaches in developing multiple classifier combination. The proposed multiple classifier combination scheme will help to develop other multiple classifier combination for pattern recognition and classification.

Keywords: Multiple classifier combination, Diversity measure, Pattern recognition and classification, Ant system algorithm, Weighted voting.

Acknowledgement

First and foremost, I would like to express my gratitude to Allah S.W.T., who has permitted me to complete this thesis.

I would like to thank my supervisor, Prof. Dr. Ku Ruhana Ku Mahamud, who has patiently supervised my work, which spend some her time, for her continuously encouragement, patience, guidance and accuracy in finishing this academic work.

I also wishes to express my gratitude to governor of Riau province, regent of Indragiri Hilir regency, and Islamic University of Indragiri for the study leave granted.

To my beloved wife, Hj. Induk Mekkah and my two sons; M. Rifqi Aufa Abdika and Mohd. Raihan Abdika and two princesses; Salsabila Abdika and Cinta Abdika; I appreciate their understanding and for being there with me while I was working through a difficult and long journey.

To my friends, I appreciate your support and encouragement.

Table of Contents

List of Tables

List of Figures

List of Appendices

List of Abbreviations

ACO	Ant Colony Optimisation
AS	Ant System
ASFSP	Ant System-based Feature Set Partitioning
Bagging	Bootstrap Aggregating
BKS	Behavior Knowledge Space
DECORATE	Diverse Cretion by Oppositional Re-labeling of Artificial Training Examples
DF	Double Fault
DT	Decision Tree
DOG	Decomposed Oblivious Gain
ECOC	Error Correcting Output Codes
GA	Genetic Algorithm
k-NN	k Nearest Neighbour
KW	Kohavi Wolpert
LDA	Linear Discriminant Analysis
MASWOD	maximum of posterior probability average with self-adaptive weight based on output vectors and decision template
MCC	Multiple Classifier Combination
NBC	Naïve Bayes Classifier
NMC	Nearest Mean Classifier
NN	Neural Network
PSO	Particle Swarm Optimisation
RS	Random Subspace
SCP	Set Covering Problem
SPP	Set Partitioning Problem
SVM	Support Vector Machine
UCI	University of California Irvine
WNNE	Weighted Nearest Neighbour Ensemble

CHAPTER ONE
INTRODUCTION

1.1 Background

Pattern classification is the process of classifying patterns into predefined category (or class label) based on their feature set (or attribute set) (Dougherty, 2013). Pattern classification aims to determine pattern categories based on characteristics of the patterns, where the categories have been priorly defined. Classification process is divided into two phases, namely training and testing phases. In the training phase, the pattern sample whose class is known (training object) is used to establish a model. In the testing phase, a model that has been established is tested with the other patterns to determine the model's accuracy (Neelamegam & Ramaraj, 2013). If the accuracy is good, then the model can be used to predict the class of unknown patterns. Figure 1.1 depicts the general framework of classification task.

Figure 1.1 Classification task general framework

Pattern classification is an important area in machine learning and artificial intelligence. The impact of poor classification will put the object into the wrong class which may lead to wrong decisions being made, hence causing losses to the recipient or the decision makers.

Classification task is widely used in the decision-making process, for example on pattern recognition (Kaur & Kaur, 2013). Pattern recognition is a discipline in which the goal is to classify object into categories (Theodoris & Koutroumbas, 2009). Depending on the application, the object can be a text, image, audio or other types of data that need to be classified. Pattern recognition has been applied in many fields such as face recognition, forensic analysis, handwriting recognition, image classification, signature verification, speech recognition, and a few others (Liu et al., 2006; Verma & Blumenstein, 2008). This research field has become an integral part in most machine intelligence systems or automatic machine built for decision making. Automatic pattern recognition and classification are important issues in the various disciplines of engineering and science (Wang, 2011).

A classifier or classifier model is any function D that maps a set of objects represented as feature vector x in a real n-dimensional space $\mathfrak{R}^{\mathrm{n}}$ to a class label ω from a set of predefined class labels $\Omega = \{\omega_1, \omega_2, \ldots, \omega_c\}$. In other words, a given pattern x is to be assigned to one of the c categories based on the n feature vector values. A classifier is the algorithm used to perform the classification task. Selection of an appropriate classifier significantly influences the pattern classification (Lu & Weng, 2007). Figure 1.2 shows the taxonomy of classifiers as cited by Jain et al. (2000) which are: (1) based on similarity; (2) based on probabilistic approach; (3) constructing decision boundaries; and (4) combined classifiers. There are many classifiers that have been developed such as: neural network (NN), support vector machine (SVM), decision tree (DT), nearest mean classifier (NMC), naïve bayes classifier (NBC), k-nearest neighbour (*k*-NN) and linier discriminant analysis (LDA) (or fisher discriminant).

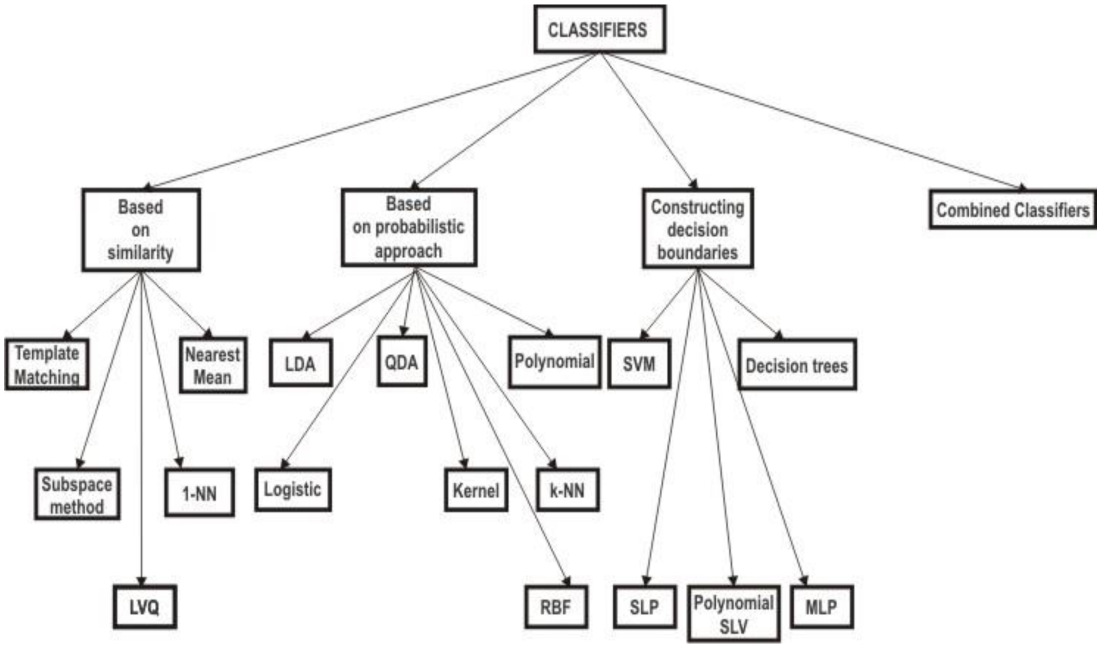

Figure 1.2 Taxonomy of classifiers (Jain, et al., 2000)

Even after nearly fifty years of research and development in the field of pattern recognition and classification, the common problems of recognising and classifying complex patterns remain unsolved. Selection of the appropriate classification algorithm is still the main issue (Basu et al., 2010). Although there are a number of available classifiers, the question of which classifier is suitable to use for a particular pattern classification task is not an easy decision to make. Previous researches indicate that each classifier has its strengths and weaknesses. Each classifier can achieve different levels of success for specific classification problems, but none of them are perfect. Not single classifier can solve all problems; each classifier has a different domain competency (Ponti, 2011). There is no one classifier that achieves the best accuracy for all situations, in other words, no classifier is always the most accurate on every dataset (Ulas et al., 2012).

The combination of several classification algorithms is considered as a new direction to solve classification problems. It aims to exploit the complementary information from other classifiers to obtain comprehensive results by combining the outputs of a number individual classification algorithm. This area is known by different terms, such as: multiple classifier combination, committees of learner, multiple classifier system, mixtures of experts, combining classifiers, ensemble classifier, ensemble learning, hybrid methods, the consensus theory, sensor fusion, multiple experts, opinion pool, decision combination and cooperative agents (Parvin et al., 2009a). Regardless of the different names, basically several classifiers are combined to obtain final classification results (Rokach, 2010). Multiple classifier combination (MCC) has been widely used in several application domains such as: character recognition (He & Suen, 2007), handwriting recognition (Frinken et al., 2010), human emotion recognition (Wu & Liang, 2011), video classification (Sigari et al., 2011), face recognition (Zhang, 2012), medical diagnosis (Srimani & Koti, 2012), email classification (Chharia & Gupta, 2013), biomedical pattern recognition (Koyuncu & Ceylan, 2013) and cancer classification (Margoosian & Abouei, 2013).

A set of classifiers which builds up multiple classifier combination is also called classifier ensemble (Kuncheva, 2004). A classifier ensemble includes a number of classifiers which is usually called the base classifier. Let $D = \{D_1, D_2, \dots, D_L\}$ be a set of L classifiers, and $\Omega = \{\omega_1, \omega_2, \dots, \omega_c\}$ be a set of c class labels. Each classifier D_i $(i = 1, \dots, L)$ gets as input a feature vector $x = [x_1, x_2, \dots, x_n]^T$, $x \in \Re^n$ and assigns it to one of the c class label from Ω, i.e., $D_i : \Re^n \rightarrow \Omega$. The final decision of classifier ensemble is taken by a combiner (Ponti, 2011).

Most of the methods used in the multiple feature classification focused on feature selection methods, i.e. choosing a single feature subset, while ignoring the rest. The selection can be done by identifying the subset of irrelevant features using feature selection algorithm. According to Maimon and Rokach (2005) there are several drawbacks of using feature selection methods, which are: (1) the assumption that the input feature set can be removed to a small subset of relevant features is not always correct. In several cases a small subset of these features are actually influenced by most of the input features and to remove features will lead to a significant loss of valuable information. (2) The subset is formed depending on the size of the training set. If the size of training set is small, the size of a feature subset will be small too. The result is that a relevant feature may be lost. Thus, the classifier may obtain a lower level of accuracy than the classifiers that access all the relevant features. (3) In several cases, even after removing a set of irrelevant features, there remains a quite large number of features.

1.2 Problem Statement

Experimental research have demonstrated that the combination of several classifiers has been very helpful in improving the prediction and reduces the generalisation error (Du et al., 2009; Rokach, 2010; Zaamout & Zhang, 2012; Turhal et al., 2013). A set of accurate and diverse classifiers is an important factor when combining multiple classifiers (Parvin et al., 2009b; Ko & Sabourin, 2013). A good ensemble consist of individual classifiers that have both accuracy and diversity. However, multiple classifier combination problems have not been fully resolved. For example, there is no standard resolution for constructing a set of accurate and diverse classifier, how many classifiers are combined and how to combine classifiers (Hernandez-Lobato & Martinez-Munoz, 2013). A good strategy is needed for combining multiple classifiers, because according to their observations, the addition of classifier members does not automatically improve the ensemble accuracy.

Several approaches have been proposed to construct a classifier ensemble. All of these approaches attempt to generate diversity in classifier ensemble. The ensemble construction aims to establish a set of "diverse" classifiers that complement each other, by creating classifiers that make errors on different patterns, so that they can be combined effectively. For this purpose, ensembles should be built as diverse as possible. One of the approaches for creating the diversity in ensemble members is the feature manipulation approach (Roli, 2009). In this approach, a feature set partition is performed in order to build an ensemble classifier. Training an ensemble classifier on different feature subsets will reduce correlation among the classifiers, does not suffer the curse of dimensionality and suitable for high dimensionality datasets without the feature selection drawback (Rokach, 2010). However, it is difficult to determine how the partition of the feature set to several subsets may lead diversity and better classification accuracy.

A number of diversity measures have been proposed but diversity is not clearly defined, no consensus on diversity measure, and difficulty of measuring diversity (Zhou, 2012). Therefore, there is no diversity measure that is satisfying although it is known that diversity is important in constructing classifier ensemble. The main problem in determining diversity measure is diversity-accuracy dilemma (Li & Gao, 2010). When the diversity approaches the highest level, the base classifier accuracy should decrease and vice versa. Furthermore, theoretically there is no formal evidence that increased diversity leads to better accuracy in multiple classifier (Wozniak et al., 2014). Consequently, there is difficulty on how to use a diversity measure to construct the classifier ensemble.

The multiple classifier combination involves two main steps: the classifier ensemble construction, and combiner construction. Effective combinations should address both steps, unfortunately most of the design methods described in the literature focus only on one phase. For example, design methods that focus only on the construction of classifier ensemble that aims to build a set of mutually complementary classifiers (or error-independent classifiers) and assume that simple fixed combiner can provide optimum accuracy (Du, et al., 2012).

This problem is significant because there is no standard guidelines in constructing a classifier ensemble (Hernandez-Lobato & Martinez-Munoz, 2013). Thus it is difficult to develop multiple classifier combination for pattern classification. Generating diversity and individual classifier accuracy are the main concern developing a multiple classifier combination scheme that gives compactness between the ensemble classifier and the combiner to produce a set of diverse and accurate classifier ensemble. Solving this problem will provide a significant contribution to establish general design of multiple classifier combination scheme that can be used to build strong multiple classifier combination.

1.3 Research Objectives

The main objective of this research is to develop an improved multiple classifier combination scheme for pattern classification. To achieve the main objective, the following specific objectives are to be fulfilled:

i. To propose an algorithm for classifier ensemble construction.
ii. To propose a parameter to measure the compactness in a set of classifier.
iii. To formulate the multiple classifier combination scheme by integrating the ensemble construction algorithm and weighted voting combiner.
iv. To evaluate the improved multiple classifier combination scheme.

1.4 Scope of the Research

Although there are several well-known topologies, this study only uses the parallel topology in the experiments to construct multiple classifier combination. Figure 1.3 shows a comprehensive categorisation proposed by Chen (2007). In parallel topology, the results of each classifier are integrated into a single decision. If either the diverse or accurate classifier is formed, and a combination rule is appropriately selected, then the system can achieve the best performance.

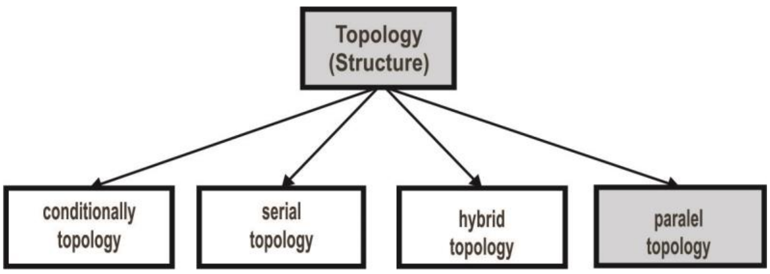

Figure 1.3 The Scope of the research in topology part

Parallel topology is the most commonly implemented topology in the combination of classifiers (Ranawana & Palade, 2005b; Zhang et al., 2006; Wozniak & Crawczyk, 2012). In this topology, several classifiers operate in parallel on the classification task. The output of each classifier is combined to produce the final decision. The advantage is that it is possible to exploit the ability of each classifier. Most of the combinations of classifiers are based on this topology, where each classifier is independent and classified as an unknown pattern. Therefore, parallel topology is chosen in this study.

There are several approaches to construct ensemble of classifiers as cited by (Roli, 2009). Figure 1.4 depicts the approaches to construct classifier ensemble. In this study, the input feature manipulation approach is used to induce diversity. Input feature manipulation approach is used with consideration: (1) training a set of classifiers on different input feature will reduce correlation among the classifiers (Rokach, 2010); (2) the dimensional feature reduction will avoid the impact of the dimensionality problem (Yang et al., 2010); (3) feature decomposition will improve the classification performance for small sample size problems (Yang et al., 2010), and (4) feature decomposition can be applied to the stable and unstable classifiers, and the training process is faster because a set of classifier models is trained simultaneously using a substantially smaller feature set instead of training a classifier model using the whole feature set (Ting et al., 2011). Feature set partitioning is a specific strategy in input feature manipulation approach to construct classifier ensemble. Every single classifier in the classifier ensemble is trained on a different projection of the original training set (or disjoint feature subsets).

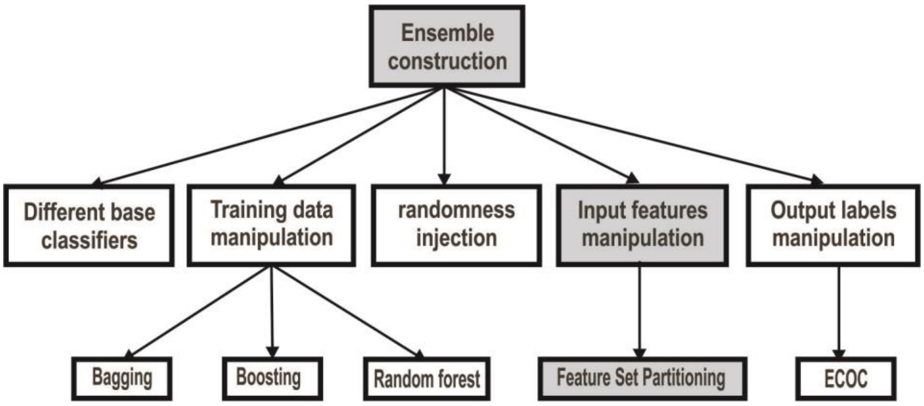

Figure 1.4 The Scope of the research in ensemble construction part

There are many combiners that have been developed. Figure 1.5 shows the various combiners that have been proposed. In this study, only the most commonly used majority voting techniques (Hansen & Salamon, 1990; Kim et al., 2003; Li & Sun, 2009; Bolon-Canedo et al., 2012; Hajdu et al., 2013) in fixed fusion combiner is chosen, where in the experiment, it combines the output of classifiers.

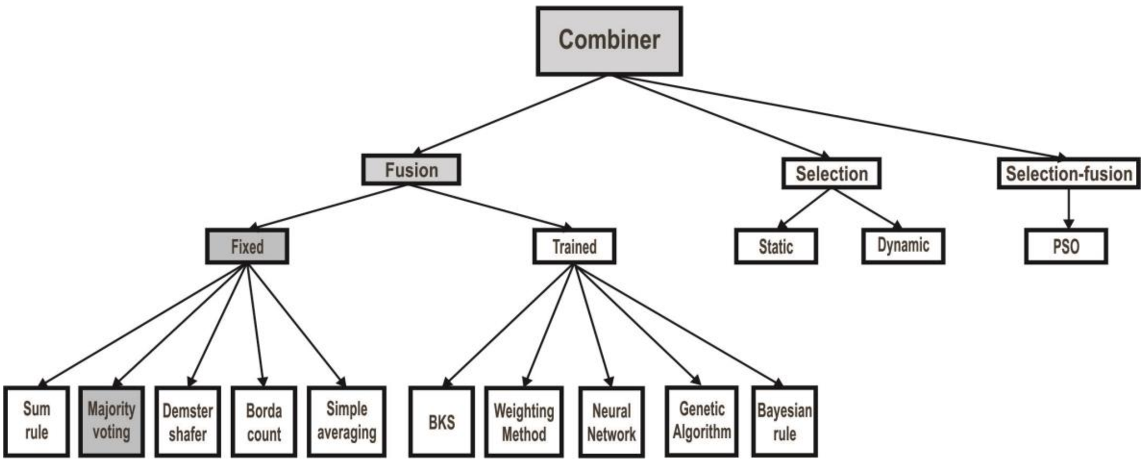

Figure 1.5 The Scope of the research in combiner part

1.5 Significance of the Research

The proposed multiple classifier combination scheme will help to develop a multiple classifier combination for pattern recognition and classification due to:

i. Classifier ensemble construction algorithm builds the classifier ensemble as accurate and diverse as possible. Furthermore, this algorithm can automatically determine the number of classifiers in the ensemble (or ensemble size).

ii. This method makes it possible to divide the high dimensional feature vector space into a number of feature vector space with lower dimension, which will allow to process low-dimensional feature vectors simultaneously. Therefore, this method is expected to overcome the curse of dimensionality.

iii. This method makes it possible to create a classifier for high dimensionality datasets without the feature selection drawback.

iv. Each of these classifiers is combined using different set of features that can provide additional information on each feature set (or comprehensive results). As we know in pattern recognition, classifiers which are suitable for a set of features may not be suitable for other feature sets. Therefore, the combination of several classifiers on different feature sets in this method can be helpful in pattern recognition.

v. This method can be useful to produce various combinations of classifiers for pattern classification problems.

1.6 Thesis Organisation

The structure of this thesis is organised as follows. Chapter 1 describes the need to develop a multiple classifier combination scheme for pattern classification. Chapter 2 reviews the literature that describes several approaches that have been used for diverse classifier ensemble construction, several existing diversity measures that have been used in classifier ensemble, a brief description of several combiners that have been developed. Chapter 3 describes the research methodology which is used in this study. Chapter 4 describes the performance of proposed classifier ensemble construction algorithm and generating diversity. Chapter 5 describes the use of weigted voting combiner in proposed combination scheme and performance of improved multiple classifier combination. Last but not least, Chapter 6 contains the conclusions of this study.

CHAPTER TWO
LITERATURE REVIEW

2.1 Introduction

There are several important things to consider in designing multiple classifier combination. According to Roli and Giancito (2002) and Yang et al., 2004, there are three main steps in designing multiple classifier combination. The three steps are: classifier ensemble construction, combiner construction and performance evaluation. According to Ko et al., (2007), there are two problems in designing a multiple classifier combination. Firstly, classifier ensemble construction algorithm; secondly, combination algorithm (or combination function). According to Wozniak and Zmyslony (2010), there are three problems in designing a multiple classifier combination which are: determining the topology, constructing classifier ensemble and constructing combiner. According to Khakabimamaghani et al. (2010), there are two key factors that influence the performance of an ensemble directly: accuracy of each single member and diversity between the members. In order to design a multiple classifier combination, several important things in ensemble classifier process that have been mentioned above, need to be reviewed such as the multiple classifier combination topology, diversity in a set of classifier, classifier ensemble construction and combiner construction. The literature review roadmap of this study is given in Figure 2.1.

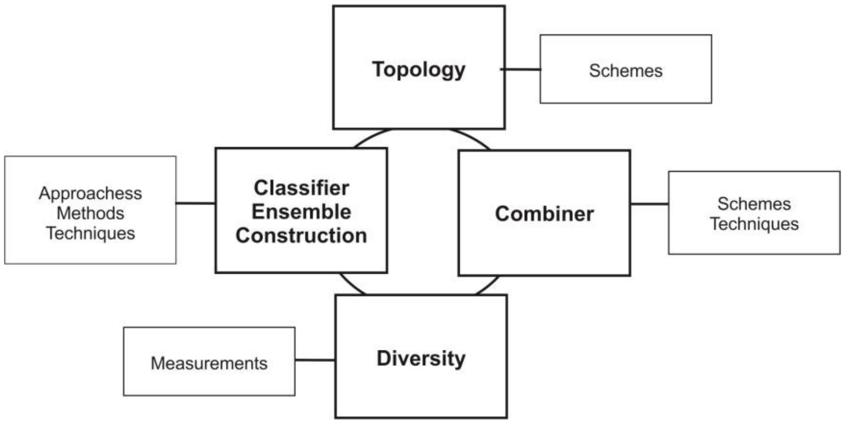

Figure 2.1 Literature review roadmap

The rest of this chapter is organised as follows. Section 2.2 presents several topologies in designing multiple classifier combination. Section 2.3 presents several diversity measures that have been proposed in the literature. Section 2.4 presents several approaches and techniques that have been proposed in constructing classifier ensemble. Section 2.5 presents several schemes of combination strategy of multiple classifier combination and a summary of this chapter is given in section 2.6.

2.2 Multiple Classifier Combination Topology

Ranawana and Palade (2005b) classified the multiple classifier combination topology into three kinds: cascading, hierarchical and parallel. In cascading topology, the output classifier is used as an input for the next classifier. Errors resulted from previous classifiers are not able to be deleted by the next classifier. In parallel topology, the output of each classifier is integrated into a single decision. In hierarchical topology, both parallel and cascading topologies are combined. A more complete explanation of the multiple classifier combination topology is summarised by Dara (2007). The topology of multiple classifier combination is summarised by her as follows: (1) conditional topology where a series of sequential classifiers are applied to the dataset. In the first strategy, one of the main classifiers is chosen to form the classification task. The advantage of this topology is computational efficiency when the classification is started by a good main classifier in pattern recognition. This topology may complicated when there is a large increase in the number of classifiers. (2) Hierarchical topology, where each classifier is applied in a subset of the dataset and will generate a class in each pattern. In conditional topology, classifiers will be completed upon satisfactory classification results obtained, while in the hierarchical topology, all classifiers should be applied to a set of datasets. (3) Sequential topology, where the classifiers are applied in sequence. Each classifier produces a modified set of possible category for each dataset. (4) Parallel (Multiple) topology, where multiple classifiers operatie in parallel on the input to produce several decisions, then decisions are combined to produce the final decision.

The selection of an appropriate topology is crucial in enhancing the performance of multiple classifier combination (Wozniak & Zmyslony, 2010). The extra cost of computing is the weakness of this topology. However the possibility to donate strengths of each classifier is the advantage of this topology. Therefore, most of the combinations of classifiers are based on this topology. In other words, the parallel topology is the most common implementation (Ranawana & Palade, 2005b; Zhang et al., 2006; Wozniak & Crawczyk, 2012). In this topology each classifier can be independent and complement each other to form diverse classifier ensembles.

2.3 Diversity Measure

A set of accurate and diverse classifiers is an important factor when constracting classifier ensemble (Parvin et al., 2009b; Ko & Sabourin, 2013). There exist a number of diversity measures in a set of classifier that have been proposed. Basically, the measure of diversity can be categorised into two groups: pairwise (the Q statistic, the correlation, the disagreement and the double fault) and non-pairwise (the entropy of the votes, the difficulty index, the Kohavi-Wolpert variance, the interrater agreement, the generalised diversity, and the coincident failure diversity). Table 2.1 shows several measures of diversity (Kuncheva & Whitaker, 2003). Diversity can be large if the diversity value is large or vice versa, depending on the measure used. The arrow (↓) indicates that the diversity is greater if the diversity measure value is lower, and the arrow (↑) indicates that the diversity is greater if the diversity measure value is greater. The attribute 'P' means Pair-wise (pairs) where the types are Y (yes) or N (no). In the pair-wise diversity, the value of diversity is calculated for each pair of classifier in the ensemble and then averaged, while in the non-pair-wise it is attempted to measure the diversity of a set of classifiers directly.

Table 2.1

Summary of Ten Measures of Diversity (Kuncheva & Whitaker, 2003)

Name	Symbol	↑/↓	P	Formula	Eq.
Q-Statistic	Q	↓	Y	$Q_{i,k} = \frac{N^{11}N^{00} - N^{01}N^{10}}{N^{11}N^{00} + N^{01}N^{10}}$ $Q_{avg}=\frac{2}{L(L-1)}\sum_{i=1}^{L-1}\sum_{j=k+1}^{L} Q_{i,j}$	(2.1) (2.2)
Correlation coefficient	ρ	↓	Y	$\rho_{i,k} = \frac{N^{11}N^{00} - N^{01}N^{10}}{\sqrt{(N^{11}+N^{10})(N^{01}+N^{00})(N^{11}+N^{01})(N^{10}+N^{00})}}$	(2.3)
Disagreement measure	Dis	↑	Y	$Dis_{i,k} = \frac{N^{01}+N^{10}}{N^{11}+N^{10}+N^{01}+N^{00}}$	(2.4)
Double-fault measure	DF	↓	Y	$DF_{i,k} = \frac{N^{00}}{N^{11}+N^{10}+N^{01}+N^{00}}$	(2.5)
Kohavi-Wolpert variance	Kw	↑	N	$Kw = \frac{1}{NL^2}\sum_{\mathrm{j}=1}^{\mathrm{N}} l(C_j)(L - l(C_j))$ $Kw = \frac{L-1}{2L}\mathrm{Dis}_{\mathrm{avg}}$	(2.6) (2.7)
Interrater agreement	K	↓	N	$K = 1 - \frac{\frac{1}{L}\sum_{j=1}^{N} l(C_j)(L - l(C_j))}{N(L-1)\bar{p}(1-\bar{p})}$ $K = 1 - \frac{L}{(L-1)\bar{p}(1-\bar{p})}KW{=}1 - \frac{1}{2\bar{p}(1-\bar{p})}Dis_{avg}$	(2.8) (2.9)
Entropy measure	E	↑	N	$E = \frac{1}{N}\sum_{j=1}^{N}\frac{1}{(L - \left\lceil\frac{L}{2}\right\rceil)}\min\{l(C_j), L - l(C_j)\}$	(2.10)
Measure of difficulty	Θ	↓	N	$V_I = (L - Li)/L$ $\theta = var(V_i)$	(2.11)
Generalised diversity	GD	↑	N	$p(1) = \sum_{i=1}^{L}\frac{i}{L}p_i$ $p(2) = \sum_{i=1}^{L}\frac{i(i-1)}{L(L-1)}p_i$ $GD = 1 - \frac{p(2)}{p(1)}$	(2.12) (2.13) (2.14)
Coincident failure diversity	CFD	↑	N	$CFD = \begin{cases} 0 & p_0 = 1 \\ \frac{1}{1-p_o}\sum_{i=1}^{L}\frac{L-i}{L-1}p_i & p_0 < 1 \end{cases}$	(2.15)

The Q statistic was proposed as a diversity measure between two classifiers C_i and C_k where $i, k \in 1, \ldots, L$ is defined by Equation 2.1, where N^{11} is the number of patterns classified correctly by both C_j and C_i, N^{10} is the number of patterns classified correctly by C_i and C_j misclassifies, N^{01} is the number of patterns classified correctly by C_j and C_i misclassifies, $N^{00}$is the number of patterns misclassified by both C_i and Cj and $N^{11} + N^{10} + N^{01} + N^{00}$=N. Q_{ij} value varies between -1 to 1. Furthermore, the average Q statistics of all pairs of classification algorithm is formulated in Equation 2.2, where L is the total number of classification algorithms. The correlation coefficient was proposed to measure diversity between two classifiers $C_i$dan C_j by statistical functions (Eq. 2.3). For each of the two classifiers, Both ρ and Q have the same sign, then have proved that | ρ | ≤ | Q |.

The disagreement was proposed to measure diversity in a set of classifier by using the proportion between the number of experiments in which a classifier is right and the other is wrong with the total number of experiments (Eq. 2.4). The double fault (DF) was proposed to measure diversity which is defined as the ratio of objects incorrectly classified by the both classifier to the total number of observations (Eq. 2.5). Table 2.3 below lists the proportion of cases for each pair of base classifier. The value of 1 indicates double fault occurs between pairs of base classifier and value of 0 indicates no double fault occurs.

The Kohavi-Wolpert (KW) was proposed to measure diversity which is defined in Eq. 2.6. The value of KW can be measured by multiplying the average measurement disagreement with a coefficient (Eq. 2.7). The measure of interrater agreement was proposed to measure diversity which measures the degree of agreement in classifier ensemble. If p is the average of the individual classifier accuracy, then the K value can be formulated using Eq. 2.8. K value may also be derived by the K value or average of disagreement measurements (Eq. 2.9).

The entropy was proposed to measure the diversity in classifier ensemble, where the value of entropy is given in Eq. 2.10, where $l(C_j)$ indicates the number of classifiers member in ensemble which correctly classifies x_j, L indicates the number of decisions in ensemble with the same value (1 or 0). The value of 0 indicates the lowest diversity and the value of 1 denotes the greatest probable diversity. The measure of difficulty was propsed to measure diversity which is defined as the variance of a random variable (Eq. 2.11). A number of classifiers L in ensemble runs on the dataset X to obtain the probability of the variable random. The set value $\left\{\frac{0}{L}, \frac{1}{L}, \ldots, 1\right\}$ indicates the proportion of correctly classifying in ensemble which x is randomly selected from the dataset.

A generalised diversity was proposed using a random variable B^2 to denote the proportion of classifiers is wrong on a randomly selected input pattern, p_i is denoted as the probability that B = i / L. Then, p (i) is defined as the probability that i randomly selected classifiers which failed on the randomly selected input pattern x. Then, using Eq. 2.12 and Eq. 2.13, the generalised diversity GD is formulated (Eq. 2.14). The coincidence failure diversity in ensemble was also proposed which is defined in Eq. 2.15 where p_i denote the probability that exactly i number of the base classifiers that make error predictions. This measure gives the highest value of 1 when all the base classifiers make unique mistakes, and give the lowest value of 0 when all the base classifiers make the same predictions.

The diversity in classifier ensemble is an important factor in combining classifiers. The accuracy of an individual classifier is also crucial in classifier combination. Although, there are several diversity measures that have been proposed, but none has focus in determining diversity-accuracy problem. When the diversity approaches the highest level, the base classifier accuracy should decrease and vice versa (Li & Gao, 2010). Therefore, the problem of using diversity measure as a guideline in determining accurate and diverse classifier ensemble construction is still an open research area. There is a need to propose a new parameter in constructing an accurate and diverse classifier ensemble.

2.4 Classifier Ensemble Construction

Several approaches have been proposed to construct a set of diverse classifiers. Roli (2009) generally summarised several approaches to construct a classifier ensemble as follows: (1) using different base classifiers; (2) injecting randomness; (3) manipulating output labels; (4) manipulating training data; and (5) manipulating input features. All these approaches try to induce classifier diversity, i.e. to create classifiers that make errors on different patterns. One approach for constructing classifier ensemble is by using different types of base classifiers (based on different models of classifiers). This approach may work well for applications where complementary information sources are available (e.g., multi-sensor applications) or distinct representations of patterns are possible (Roli, 2009). In ensemble construction technique via injecting randomness approach, several identical classifiers were trained for several times with different random values of initial weight (Ranawana & Palade, 2005a). The output label manipulation approach is typically valuable when the output class number is big. Dietterich and Bakiri (1995) have proposed a general method, called error correcting output codes (ECOC), for constructing classifier by this approach. The training data manipulation approach constructs a classifier ensemble by training the base classifier using different training sets. A small change in the training set will produce different predictions. This approach is more effective for learning problems where the prediction samples have different levels of difficulty. Further more this approach is efficient when data is limited in the training process, because it uses random sampling with replacement. Several methods for constructing diverse classifiers using different training sets have been presented such as boosting (Schapire, 1990), adaboost (Freund & Schapire, 1996), bagging (Breiman, 1996), random forest (Breiman, 2001) and diverse ensemble creation by oppositional relabeling of artificial training examples (DECORATE) (Melville & Mooney, 2005). As the focus of this study, in the following section, the input feature manipulation approach in constructing classifier ensemble is reviewed.

2.4.1 Input Feature Manipulation Approach

Another approach to construct classifier ensemble is to manipulate the input feature set. The idea of input feature manipulation approach is to simply give each classifier a different projection of the training set. Feature decomposition method (also called feature subset-based ensemble method) are those that manipulate the input feature in constructing a set of classifier. In this method, vertical partitioning on the training sets is performed in order to build an ensemble.

The simplest technique for feature decomposition is by assigning a projection of the sample training set on random feature subsets as training sets to ensemble members. Random feature subsets are created by random feature selections of original feature set. The sampling method in order to determine the training set can be done by sampling with and without replacement. The main method of this approach is the Random subspace (RS) method (Ho, 1998). Other methods that have similar idea with this method is the multiple feature subsets (Bay, 1998) and attribute bagging (Bryll et al., 2003). In all the approaches features are randomly assign to each ensemble member. Differences exist only in the parameter determination of feature subset size and the size of the ensemble.

In random subspace, the ensemble classifier is constructed using the following algorithm: (1) Let the number of training objects be *N* and the number of features in the training data be *D*. (2) Choose *L* to be the number of individual classifiers in the ensemble. (3) For each individual classifier, *I*, choose *d_l (d_l <D)* to be the number of input variables for *I*. It is common to have only one value of *d_l* for all the individual classifiers. (3) For each individual classifier, *I*, create a training set by choosing *d_l* features from *D* without replacement and train the classifier (4) For classifying a new object, combine the outputs of the *L* individual classifiers by majority voting.

According to Skurichina and Duin (2002), the random subspace is suitable for high dimensional data. Based on their experiments, if the size of training set is relatively small compared to the dimension of the data, random subspace gives good results. The evaluation has been performed by comparing this method with bagging and boosting in constructing NMC ensembles. Random subspace showed good accuracy compared to the other two methods. Furthermore, random subspace is also able to overcome the instability and overfitting problems (Tao & Tang, 2004). Random subspace method is widely used in experimental studies on constructing classifier ensembles (Ahn, et al., 2007; Serpen & Pathical, 2009; Li et al., 2013). However, there is no control over the accuracy of each member classifier and diversity in the ensemble classifier with random feature decomposition.

2.4.2 Feature Set Partitioning

Feature set partitioning is a particular condition of feature decomposition. Feature set partitioning does not only search for single useful subsets. In this strategy, the original training set is decomposed into several subsets and each subset constructs classifiers. Thus, a classifier ensemble is trained in such a way that each classifier works on a different feature subset. This methodology is appropriate for the classification task with a large number of features (Rokach, 2006). Figure 2.2 presents a Venn diagram of the search space of feature subset-based ensemble which loads the feature set partitioning seach space, and the second loads the search space of feature selection (Rokach, 2008).

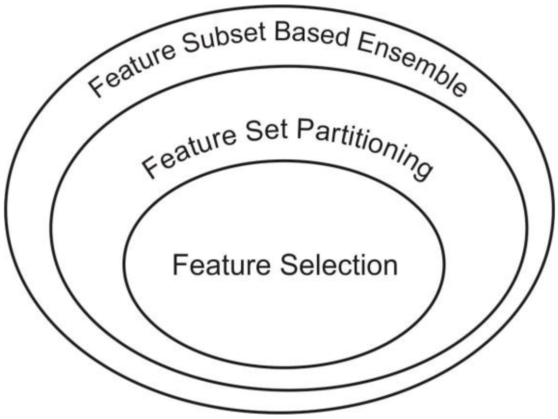

Figure 2.2 Venn diagram of the search space of feature orientation (Rokach, 2008)

Liao and Moody (2000) proposed feature set partitioning by pair-wise mutual information feature grouping. Statistically, correspond features are assigned in to the same feature subset. For this aim, a hierarchical clustering algorithm is used. Furthermore, artificial neural networks are constructed for each group and run to get the final decision. Rokach and Maimon (2005) proposed decomposed oblivious gain (DOG). This partition searches by applying the incremental oblivious decision trees algorithm. One disadvantage is that the DOG method has no backtracking abilities. Furthermore, DOG starts the search beginning an empty partition, which can lead to a quite small subset of features.

Ahn, et al. (2007) showed that the randomly partitioned input features to several subsets, thus each classification algorithm is assigned with different subsets, which is specially helpful for high-dimensional data and unbalanced data. Rokach et al. (2007) proposed to combine the results of different feature selection. The experimental studies show that integrating different feature selection algorithms may significantly enhance the accuracy of classification tasks. Earlier, Rokach and Maimon (2005) developed a general framework for disjointing feature set partitions.

Two were empirically tested using more than one dataset. This framework shows that feature set decomposition can improve the accuracy of decision tree. Rokach (2008) applied genetic algorithm (GA) for feature set partitioning. This algorithm has been tested with different datasets and the results show advantages compared to other methods and this algorithm accelerates the execution.

According to studies that have been reported, feature set may be partitioned by random selection, statistical approaches and genetic algorithm. Ant colony optimisation has showed better performance than other popular heuristics are indicated in some references, for instance in comparison with the simulated annealing algorithm (Su et al., 2005; Carpaneto & Chicco, 2008; Chang, 2008) and with genetic algorithm (Su et al., 2005; Chang, 2008) as cited by Chicco (2011). Therefore, the use of ACO for feature set partitioning is considered in this study.

2.4.3 Ant Colony Optimisation for Set Partitioning Problem

Ant colony optimisation (ACO) algorithm was introduced by Marco Dorigo in the early 1990s. This algorithm is inspired by the behavior of ants in finding the shortest path from the colony to the food. The ability of ants finding the shortest route is that they leave a pheromone on their tour paths. Pheromones are chemicals used to recognise other individuals or groups, and to assist the process of reproduction. In contrast to hormones, pheromones spread outside the body and can only be affected and recognised by other similar individuals of the same species. This process is known as pheromones relics stigmergy, the process of modifying an environment that not only aims to remember the way back to the nest, but also allows the ants to communicate with its colonies. However, the pheromone trail will evaporate and will reduce the strength of its appeal. The longer ants commute through these pathways, the more likely the pheromones will evaporate.

In order to get the optimal path, the ants take several processes: (1) At first, the ants walk around randomly, to find food. (2) When they find food, they will bring them back to their colony while providing a sign with the pheromone trail. (3) If other ants find the path, they will not need to travel randomly again, but will follow the trail. (4) Returning and reinforcing it if they eventually find food. (5) An ant which accidentally finds the optimal path will take this path faster than his colleagues, conduct more frequent round-trips, and consequently leave more pheromones on the paths taken slower. (6) Pheromones are highly concentrated and will attract other ant to change lanes to the most optimal path, while the other lines will be abandoned. (7) In the end, all of the ants that had sought different paths will switch to a single lane that turns toward the most optimal lane from the nest to the food (refer Figure 2.3) (Blum, 2005).

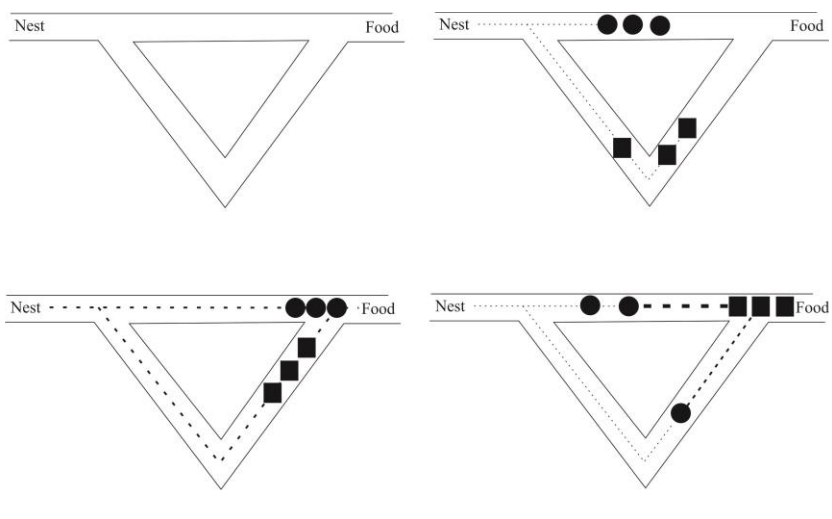

Figure 2.3 The Shortest path finding capability of ant colonies demonstration (Blum, 2005)

Ant colony optimisation algorithm has been developed over two decades. Ant-based algorithm is a group of algorithms that use the ant colony optimisation metaheuristic to handle the optimisation problem. Several variants of the ant colony optimisation algorithm have been developed as listed in Table 2.2.

Table 2.2

The Ant Colony Optimisation Variants

ACO variant	**Main reference**
Ant System (AS)	Dorigo (1992)
Elitist AS (EAS)	Dorigo, et al. (1996)
Ant-Q	Dorigo & Gambardella (1996)
Ant Colony System (ACS)	Dorigo & Gambardella1997)
Max-Min Ant System (MMAS)	Stützle & Hoos (2000)
Rank-based (RAS)	Bullnheimer et al. (1999)
ANTS	Maniezzo & Carbonaro (2000)
Hyper-Cube Framework (HCF)	Blum & Dorigo (2004)
Omicron ACO	G´omez & Bar´an (2005)

Although many ACO variants have been developed, they are all covered by the ACO metaheuristic approach for solving combinatorial optimisation problems. The ACO metaheuristic is a framework for applying ant-based algorithm for the solution of optimisation problems. A general description of the framework of the ACO metaheuristic (Blum, 2005) is depicted in Figure. 2.4.

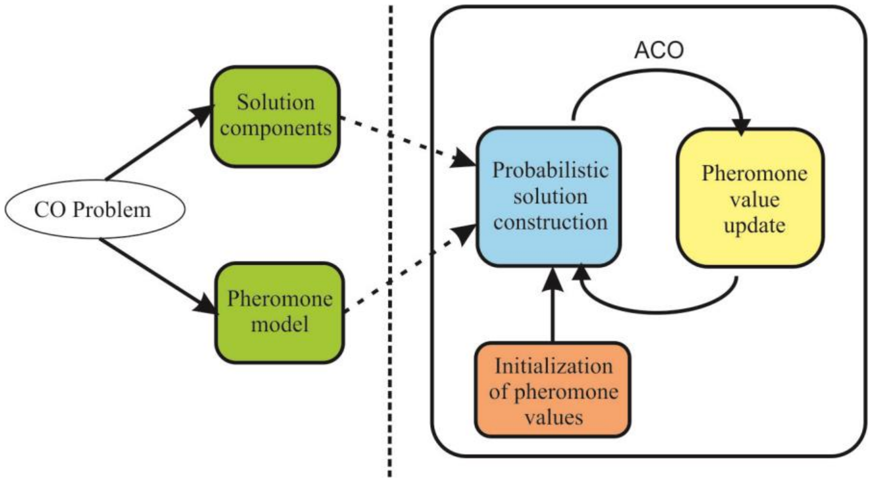

Figure 2.4 The Working of the ACO metaheristic (Blum, 2005)

There are various problem types that can be solved by ACO algorithm. Several examples of common problems are routing, assignment and scheduling (Dorigo & Stützle, 2004). Routing problem is a problem in sending a particular object through the right route. Some applied ACO to routing problems such as travelling salesman (Lizárraga et al., 2013), vehicle routing (Bell & McMullen, 2004) and network routing (Zhao et al.,2010). The Travelling Salesman Problem (TSP) is the problem to find the shortest possible route that will take him to every city exactly once. Vehicle routing problem is a problem to determine the fleet of vehicles in order to serve the customers, where each vehicle should visit the customers. Network routing is the problem of finding the minimum cost path in order to process the transfer of data packets from a source node to a destination node.

Assignment problem is the problem of placing an appropriate resource for a specific activity such that it minimises the cost. Assignment problem is one of the problems often encountered by managers as a resource allocator. Here, the manager must assign a number of resources that have different capabilities in a number of different tasks. Managers will face the prospect of working on a number of assignments for a number of tasks. Managers will assign the various possible assignments to a number of resources. Each assignment will have different advantages. Some examples that include the assignment problem are graph colouring (Bui & Nguyen, 2008), quadratic assignment (Hong, 2013), generalised assignment (Guo & Shi, 2012), channel assignment (Parshapoor & Bilstrup, 2013), course timetabling problem (Ayob & Jaradat, 2009).

Scheduling problem is the problem of determining a set of activities that share the resources of limited capacity and need to be processed such that various constraints, primarily temporal, are satisfied. Scheduling problems arise in a variety of activities. Examples of the application of ACO in scheduling problem are job shop (Xing et al., 2010), flow shop (Yagmahan & Yenisey, 2010), total tardiness (Berrichi & Yalaoui, 2013), project scheduling (Chen & Zhang, 2013) and the total weighted tardiness (Aljanaby & Ku-Mahamud, 2011).

The ant system (AS) algorithm is an example of an ant based-algorithm. Ant system was the original term used to refer to a range of ant-based algorithms, where the specific algorithm implementation was referred to as ant cycle. The so-called ant cycle algorithm is now referred to as ant system. This is an original and most famous in the ant-based algorithm that has been used and is proven to solve various optimisation problems (Zhao & Yan, 2009; Shang & Wang, 2010; Jevtic et al., 2010; Rebeiro & Enembreck, 2013).

The ant system algorithm is the baseline of ant-based algorithms such as elite ant system, rank-based ant system, max-min ant system, and ant colony system. The ant system algorithm consists of two main stages: the ant's solution construction and the pheromone update. Pheromone values τ_{ij} associated with edges, are initially set to a given value τ_0, and the heuristic information $\eta_{ij} = \frac{1}{d_{ij}}$ inversely proportional to the distance between the node i and node j. At each tour iteration, every ant in the colony build its solution according to the probability of moving from node i to node j as formulated in equation 2.16

$$p_{ij}^{k} = \frac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{l \in \mathcal{N}_i^k}[\tau_{il}]^{\alpha}[\eta_{il}]^{\beta}} \; if\ j \in \mathcal{N}_i^k, and\ 0\ otherwise \tag{2.16}$$

where α and β are two parameters that weigh the relative importance of the pheromone trail and the heuristic information, and $\mathcal{N}_i^k$ is the feasible neighbourhood of ant k in node i. When all m ants have built a solution, the pheromone trail is evaporated according to

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij} \tag{2.17}$$

where $0 < \rho \leq 1$ is the pheromone evaporation rate. After the evaporation process, the ants increment the pheromone table

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}{}^{k} \tag{2.18}$$

with

$$\Delta\tau_{ij}{}^{k} = \begin{cases} {}^{1}/_{f(S_k)}, if\ arc\ (i,j) belongs\ to\ the\ tour\ build\ by\ the\ k-th\ ant \\ 0\ otherwise \end{cases} \tag{2.19}$$

where $\Delta\tau_{ij}{}^{k}$ is the amount of pheromone deposit by ant k, on the edge it has visited, which is proportional to the solution S_k (tour length) built by the ant. When the stopping criterion is achieved, the algorithm returns only one solution, which contains the best tour.

Set partitioning problems (SPP), deal with distributing the elements of a set (S) to a specific number of subsets according to a given purpose. The issue is how the elements in a set (S) can be partitioned into smaller subsets such that all elements in S is contained in one and only one partition. Let $I = \{1,2,3, \dots m\}$ be a non-empty and finite set and $S = \{S_1, S_2, S_3, \dots, S_n\}$ be the set of feasible subsets of I. Let a set $P \subset \{1,2,3, \dots, n\}$. P defines a partition of I if and only if all of the following conditions hold: (1) $\emptyset \notin P$, (2) $\bigcup_{j \in P} S_j = I$ and (3) $S_j \cap S_k$=$\emptyset$ $\forall_{j,k \in P, j \neq k}$. Figure 2.5 shows an example of a set partition.

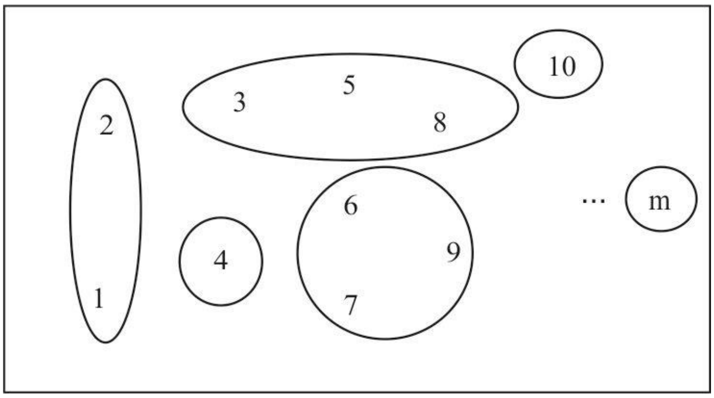

Figure 2.5 An Example of a set partition

Each element is included in exactly one of the subsets S_j that are part of the partition P.Let c_j be the cost associated with S_j. Then $\sum_{j \in P} c_j$ is the cost of partition P. In the Set Partitioning Problem (SPP) the objective is given S find the minimal cost partition P^* of I.

Many real life problems can be formulated as SPP such as workforce planning, vehicle routing, truck delivery management, vehicle scheduling, bus driver scheduling, identification of subgroups in football leagues and team scheduling

(Peker et al., 2012). The well known application of set partitioning is airline crew scheduling (Mesquita & Paias, 2008), In flight scheduling, crew scheduling subtask takes as input a set of data pairs crew, where the selection of a crewmate is done that causes minimal cost and ensure that each flight is covered exactly once.

Set partitioning problems are the most difficult (NP-Hard) problems among combinatorial problems to solve due to their complexities (Cetin et al., 2008). Set partitioning is also a very constrained combinatorial optimisation problem (Crawford et al., 2009). The SPP has been studied extensively in modeling optimisation problems in the real world. In the literature, there have been several studies that have applied ACO for set partitioning. The following are some studies that applied ACO to SPP.

Maniezzo and Milandri (2002) presented a paper to solve set partitioning using ACO algorithm in a paper entitled "An Ant-based Framework for Very Strongly Constrained Problems". This paper presented an extended ant framework improving the effectiveness of ACO algorithm in to such problems. A new algorithm, named BE-ANT, is designed for solving any combinatorial optimisation problem in general, and very hard, tightly constrained in particular instances. The resulting framework can also be applied to problems in which the standard ACO framework is ineffective. Computational results are presented both on standard set partitioning problem instances and on vertical fragmentation problem instances. BE-ANT has a performance comparable with other solution methods. The computational results are still preliminary and need to be improved.

A set partitioning problem study named "ACO with Lookahead Procedures for Solving Set Partitioning and Covering Problems" was conducted by Crawford and Castro (2005). This study proposed on how to describe the SPP by a matrix representation. A vector (column) was used to represent a subset S_j. The vector contains only 0 and 1. The size of the vector is equal to m. The i element in the vector is 1 if i is in S_j and 0 otherwise. In this case, given a set of columns and rows, the objective is to choose a subset of columns covering all rows while minimising costs. The solution is represented by a subset of columns. The solution components are represented by nodes and not by arcs. Each ant starts with an empty set of columns. Then, the ant adds columns one at the time, based on pheromone values, until all rows are covered. Figure 2.6 describe the ACO algorithm to solve SPP.

```
Begin
  InitParameters()
  While remain iterations do
    Foreach ant do
      While solution is not completed and (TabuList <> j) do
        Choose next column j with Transition Rule Probability
        AddColumnToTabuList(j)
      End While
    End Foreach
    UpdateOptimum()
    UpdatePheromone()
  End While
  Return BestSolution
End
```

Figure 2.6 ACO algorithm for set partitioning problem (Crawford & Castro, 2005)

Crawford and Castro (2006) undertook a study to solve set partitioning using ACO algorithm in a paper entitled "Ant Colonies using Arc Consistency Techniques for the Set Partitioning Problem". The authors solved some benchmarks of Set

Partitioning Problem. The techniques used to solve them were hybridisations of ant colony optimisation with constraint programming techniques based on arc consistency. In the proposed hybrid algorithms, the authors explored the addition of this mechanism in the construction phase of the ants so they can generate only feasible partial solutions. Computational results are presented showing the advantages to use this kind of additional mechanism to ant colony optimisation.

A study conducted by Randall and Lewis (2010) on set partitioning problem has implementaed the ant colony optimisation-based algorithm. The effiectiveness of the algorithm was enhanced by adding feasibility restoration, solution improvement algorithms and candidate set strategies. These algorithms can be applied to complete solution vectors and as such can be used by any solver. Moreover, the principles of the support algorithms may be applied to other constrained problems. The experimental results provide that the ant colony optimisation algorithm can efficiently solve small to medium sized problems. Crawford et al. (2013) further presented a hybrid solver based on ant colony optimisation combined with arc consistency for solving the set covering problem (SCP) and set partitioning problem (SPP). The hybrid approach was tested with set covering and set partitioning dataset benchmarks. It was observed that the performance of ACO had been improved by embedding this filtering technique in its constructive phase.

The ACO has been successfully applied in solving the set partitioning problem. Ant system is the most popular variant of ACO and is proven to solve various optimisation problems. It is expected that the ant system can be used for feature set paritioning in constructing classifier ensemble. Furthermore, to form a multiple classifier combination scheme, an appropriate combiner is required to combine the classifier outputs. Therefore, several combiners that have been reported in the literature are reviewed in the next section.

2.5 Combiner Construction

The multiple classifier combination scheme involves two main steps: the classifier ensemble construction, and combiner construction. This section discusses the literature which is related to combiner construction. The combiner (or fuser) aims to create a fusion mechanism that can exploit the diversity of classifiers and optimally combine them. Kuncheva (2004) categorised the operating level of classifier combination based on the output which is produced by the classifier, into three levels namely the abstract level, rank level and measurement level. At the abstract level (abstract-level combination methods) each classifier D_i produces a class label. Thus, for any object $x \in \Re^n$ to be classified, the L classifier outputs define a class label vector $D = [D_1(x), \ldots, D_L(x)]^T$. The top candidate which is produced by each classifier is used. At the ranked-level combination methods, the output of each classifier D_i is a subset of the class label set, the ranking list of candidates which is produced by each classifier is used. At the measurement-level combination method, both ranked-level and similarity measurement or confidence value of each candidate are used. The combination method that works at abstract-level can be applied to any classifier. In contrast, the ranked-level and the measurement-level can cause other difficulties when they are used in which each classifier individually gives the top candidate, or when combined classifier provides a value of similarity measurements (probability, scores, similarity or dissimilarity etc.).

Woods et al. (1997) separated combination scheme into two major types which are classifier fusion and classifier selection. This scheme has been further extended by Kuncheva (2001), in which the differences between static selection and dynamic selection are distinguished. This categorisation is further expanded into the selection-fusion scheme (Kuncheva, 2002a).

2.5.1 Classifier-fusion Scheme

The classifier fusion scheme assumes that all of the classification algorithms are equally qualified and the outputs of all the classification algorithm are considered. Sharkey (2002) further separated the fusion scheme into fixed classifier fusion and trained classifier fusion. In the fixed classifier fusion scheme, the weight of each classifier in the classifier ensemble is fixed. There is no training process to determne the weight of each classifier. The most simple fixed classifier fusion is to implement simple operators such as sum-rule and product-rule to the outputs of all individual classifiers. Result will follow the max or min value of the final classification decision. In general, the benefit of the fixed classifier fusion approach is its simplicity and lower computational cost. However the drawback is the lack of adaptability in the integration procedure (Chen & Kamel, 2007).

Lincoln and Skrzypek (1989) proposed fixed classifier fusion namely simple averaging. In this combination rule, the final output of classifier combination is the average of each classifier output value. Some experiments have shown that the simple average is an effective approach (Hansen & Salamon, 1990; Xu et al, 1992;

Breiman, 1996). But the weakness of this technique is equal treatment to each member of classifiers, there is no emphasis on the performance of classifiers.

Shilen (1990) proposed one combiner at the abstract level namely Dempster-Shafer method. This rule used a priori knowledge of information about the performance of each individual classifier. The Dempster-Shafer combines several different classifiers using a level of recognition and substitution rates as a priori knowledge (Lu & Yamaoka, 1994). Generally, if given an input pattern x, all the classifiers that have the same output is collected into a group of E_k where k is the number of different outputs. Thus, after this combination, each group E_k is equivalent to a new classifier with the recognition rate and substitution rate that is new. The next step is to combine the recognition and substitution rates E_k to calculate the confidence true output and confidence through the equivalent output of one classifier. However, this integration method requires heavy computation and gives low generalisation performance.

Xu et al. (1992) proposed Bayesian method which is based on applying Bayes theorem by error consideration of each classifier. In this method, probabilistic summary for each class is defined earlier. However, one of the significant disadvantages of this method is that the mutual independencies between classifiers are ignored, but this does not always happen in the real application (Kim et al., 2002).

Ho et al. (1992) presented the Borda Count as a vote method on the rank level. In this method, each class which is produced by individual classifiers is ranked. The first rank is given the highest value and last rank is given the lowest value. The output is the class with the highest number of rank overall. The advantages of borda count method are its simplicity, lower computational cost and no training required. Particularly on classification problems with more than two classes, this method provides better results than majority voting, because more information is used. However the weakness of this method is the equal treatment for each classifier, so there is no emphasis on the classifier that gives more contribution to the output.

In the trained classifier fusion, the weight of each individual classifier is considered in the decision. Generally, there is a training process to learn the weights of each classifier. The advantages of trained classifier fusion approach are its flexibility and potentially better performances than fixed classifier fusion; however the disadvantages are high memory and time requirements.

Huang et al. (1994) proposed a combination method using the data transformation and Neural Network. The output value of each classifier is first converted into a likelihood measurement. The measurement value that has been transformed is inputted into the neural network layer, and then the neural network produces the final classification decision.

Lee and Srihari (1995) showed the neural network that consists of multi-layer perceptrons trained continuously until the required accuracy is achieved for the combination classifier. Breve et al. (2006) combined classifiers by neural network for noisy data classification. However, one weakness of using the artificial neural network is expensive computational cost (Bishop, 1995; Lera & Pinzolas, 2002). Jacobs (1995) proposed the weighted averaging as another variant of simple averaging. This technique gives weight to each classifier before calculating the average amount of each output from the classifier. In this technique, a weight is attached to each individual classifier. The final classification result is calculated based on the performance of each member of classifiers. The total weight is 1 and each classifier member gets a part of the total weight according to their performance. Therefore, the strength of each classifier is considered, but the weakness of this technique is sensitive to biased classifiers.

Huang and Suen (1995) proposed a combination scheme known as behavior knowledge space (BKS) which can aggregate the decisions obtained from individual classifiers. BKS is the combination technique on the abstract level that combines the decisions which is generated by each classifier. BKS is followed by two phases namely learning phase and decision phase. During the learning phase, the training set is given to the K classifier to gather a priori knowledge information that is required on the decision phase. BKS combines the decision of the multiple classifiers by creating a lookup table for each nominee. The lookup table consists of all probable combinations of the class categories and each cell in the table is one probable combination. The final classification result is achieved by running a maximum operator on the set of class labels in each cell. Experiments on unconstrained handwritten numerals have shown that this method achieves promising performances and outperforms simple voting, Bayesian, and Dempster-Shafer technique. However, in order to provide good performance, BKS method needs to be trained with a large training dataset.

Kuncheva and Jain (2000) presented two simple ways to use genetic algorithm on multiple classifier combination. The genetic algorithm is used as a combination scheme to optimise the weights connection. The starting process is randomising the weight values gradually. The weight reflects the importance of each classifier. The classifiers used were linear discriminant classifier, quadratic discriminant classifier, and logistic classifier. Kim et al. (2002) proposed a method for combining multiple classifiers based on genetic algorithm. The classifier used was the neural network. The method shows better performance than majority voting, bayesian, behavior knowledge space, borda count, weighted borda count, sum and neural network.

Aslam and Montague (2001) proposed weighted borda count as a variant of the Borda Count which gives weight to the individual classifier. Weights are intended to address the performance of each individual classifier. An advantage of weighted borda count is that it does not require training. Although the weighted borda count considers individual classifier performance, but this technique still requires classifiers that are able to give ratings on the potential class.

2.5.2 Classifier-selection Scheme

In the classifier selection scheme, only one classifier is needed to correctly classify the input pattern. Select a single "best" classifier from base classifiers for the final decision. In order to do this, it is important to define a procedure to choose a member of the ensemble to make the decision. Kuncheva (2002a) further categorised the selection scheme into the static classifier selection and dynamic classifier selection. The subdivision depends on whether the selection is created dynamically or statically. In static classifier selection, the selection of the best classifier is specified during a training phase. In dynamic classifier selection, the choice of a classifier is made during the classification phase. One individual classifier among ensemble classifiers is chosen. It is called “dynamic” because the classifier used critically depends on the test pattern itself.

Woods et al. (1997) proposed the dynamic classifier selection by local accuracy. Other dynamic classifier selection approaches are the prior selection method, posterior selection method and dynamic classifier selection which are based on multiple classifier behavior (Giancito & Roli, 2000). The superiority of the dynamic selection method is that the error-dependency can be omitted (Chen & Kamel, 2007). A critical point in the dynamic classifier selection is that the choice of one individual classifier must exceed any other classifier, so it depends on the ability of the estimated generalisation from the classifier (Kuncheva, 2002a). Dynamic ensemble selection’s advantage is the ability to estimate distribution to a group of classifiers rather than a single individual classifier. So far, this scheme seems to work fine (Ko et al., 2008).

2.5.3 Selection-fusion Scheme

In the selection-fusion scheme, the selection and fusion actions are conducted in order to decide the best choice in classifying unknown input pattern. Normally, there is a certain criterion in establishing either the selection or fusion strategy. The basic idea is to choice the selection strategy if there is a best classifier which is really powerful in classifying the testing pattern. Otherwise, the combination strategy is used. Giancito & Roli (2001) developed a hybrid dynamic classifier selection method which is based on multiple classifier behavior and Kuncheva (2002b) developed a hybrid dynamic classifier selection by using decision templates. Yang et al. (2009) proposed a combination algorithm which implemented classifiers’ selection and combination simultaneously with particle swarm optimisation (PSO). The experiment shows that the proposed method gives the best performance of 5 out of 9 datasets and averagely outperforms majority voting rule, max rule, min rule, mean rule, median rule and product rule.

2.5.4 Voting Approach for Classifier Combiner

The most popular, fundamental and straightforward classifier combiner for class label output is the majority voting (Hansen & Salamon, 1990; Brown, 2010). Majority voting is based on Condorcet jury theorem, proposed in the context of social sciences since the end of 18th century. The theorem proves that the judgment of a committee is superior to those of individuals, provided the individuals have reasonable competence. The application of majority voting for classifier combiner was first proposed by Hansen and Salamon (1990). The majority voting is often used to combine multiple classifiers in order to solve the problems of classification (Wanas & Kamel, 2002; Bryll et al., 2003; Li & Sun, 2009; Bolon-Canedo et al., 2012; Hajdu et al., 2013).

The majority voting is one of the fixed classifier fusion scheme. Several popular ensemble methods used the majority voting as a combiner as cited by Yang et al. (2010). Figure 2.7 shows three popular ensemble methods, namely bagging (Breiman, 1996), boosting (Freund & Schapire, 1996) and random forest (Breiman, 2001) using voting in combining classifiers outputs.

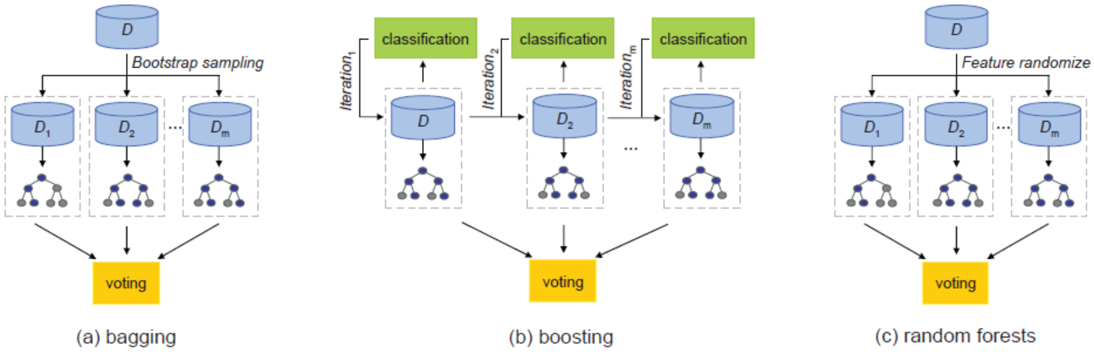

Figure 2.7 Three popular ensemble methods (Yang et al., 2010)

This technique considers only the most likely class provided by each classifier and chooses the most frequent class label among the classifier outputs. In order to alleviate the problem of ties, the number of classifiers used for voting is usually odd. Here, every individual classifier votes for one class label. The class label that most frequently appears in the output of individual classifiers is the final output. The ensemble decision for the majority voting can be described as follows: class label *k* is assigned to *x* if and only if

$$\sum_{j=1}^{L} d_{j,k}(x) \geq \sum_{j=1}^{L} d_{j,i}(x), \qquad i = 1, \dots, c, \;\; d_{i,j}(x) \in \{0,1\} \tag{2.20}$$

One of the advantages of majority voting is the ability to combine the output of each classifier regardless of what classifier is used. The weakness of this combiner is that it does not consider the strength of classifier, in other words, the strength of each classifier is considered equal in vote.

The weighted voting is a trainable version of majority voting proposed by Littlestone and Warmuth (1994). Unlike majority voting, this technique gives weight to each classifier before voting. The weight for each classifier is obtained through the training process. To make an overall prediction, a weighted vote of the classifier predictions is performed to predict the most weighted class. Although this technique considers the strength of each classifier, but the lack of this technique is it only considers the first rank class or classes that most probably found in each classifier.

Wanas and Kamel (2002) presented a feature based approach as well as training algorithm. In the feature based approach, each classifier is trained independently. This algorithm is based on the adaptive training algorithm for training neural network ensembles. This training approach helps optimise the weights to achieve better overall classification. Based on the experiment, the results on two benchmark problems and comparison to a single classifier show that the approach improved on classification accuracy.

A novel multiple classifier combination that incorporates global optimisation based on a genetic algorithms to develop multiple classifiers was introduced by Stefano et al. (2002). The multiple classifier combination adopts the weighted voting approach to combine the output of the classifiers. The weights are obtained by maximising the performance of the ensemble. This multiple classifier combination has been tested on a handwritten digit recognition problem. Based on the results of an experiment conducted on 30,000 digits from the NIST database, it shows good performance.

Gangardiwala and Polikar (2005) presented a modified approach in determining the weight for majority voting. The classifiers are weighted dynamically for each instance, depending on the estimated likelihood to correctly classify the instance. The idea of this approach is that the classifier, whose training dataset is closest to the given instance, has more information about the instance. Therefore it is more likely to classify the instance correctly. The proposed algorithm provides improved performance compared to Adaboost based experiments over benchmark dataset.

Zhang et al. (2006) proposed a new parallel multiple classifier combination method namely the maximum of posterior probability average with self-adaptive weight based on output vectors and decision template (MASWOD). This method aims to build a robust classifier combination and enhance the traditional weight calculated by the confusion matrix, using decision templates and error punishment factor. Each individual classifier outputs are treated as inputs to the second level classifier which combines the results of each first level classifier using a self-adaptive weight. This algorithm requires a training phase to train second level classifiers. Experiments were performed on the University of California, Irvine (UCI) machine learning repository datasets (Frank & Asuncion, 2010) to compare MASWOD with the classical Bayesian algorithm in order to combine several classifier outputs. Experimental results showed MASWOD algorithm can efficiently improve the performance of classification, where the classical Bayesian cannot always improve classification performance. This proves that the algorithm is efficiently and better because the self-adaptive weight can improve individual classifier's influence on the decision making.

The weighted majority voting has also been used as a combiner in predicting financial distress (Sun & Li, 2008). The voting weight was specified by a priori performance measure which was calculated from confusion matrix. In the experiment, 135 pairs of Chinese listed companies and 35 financial ratios were initially used. The stepwise discriminant analysis method was used in feature selection. After voting weight determination, the performance of financial distress prediction was compared with single classifier approach. It was concluded that the use of weighted majority voting for financial distress prediction give higher accuracy and lower variance than any single classifiers.

Mu et al. (2009) conducted an experiment for analysing the performance of a weighted voting technique to combine the output of multiple classifiers. The weighted voting technique is applied to face and voice recognition problem. The effectiveness of this technique was tested on images and voice benchmark dataset. The results show the benefit of developing weighted voting based multiple classifier combination. The weighted voting successfully achieved high identification rates and outperforms the majority voting. It can be concluded that the weighted voting technique can be used in combining any independent classifiers.

A multi-weighted majority voting strategy to improve the performance of classification task for complex facial security application was proposed by Huang and Wang (2009). Support vector machine was used as the classification algorithm. The hierarchical classification method and the multi-weighted majority voting strategy are two important parts in this strategy. Exerimental results indicate that the proposed algorithm improves the performance of face authentication when tested with a massive number of training and testing data.

Wozniak (2009) presented an evolutionary approach to produce classifier ensemble based on weighted voting. Several classifier fusion methods were evaluated through experiments on seven (7) datasets from UCI. The goal of the experiment is to evaluate the ability of weight-based fuser. Experimental results justified the use of weighted voting for classifier combination. Unfortunately, it is difficult to determine the weights in an analytic way. Hence the use of heuristic optimisation method seems a promising patterrn classification research direction.

Kim et al. (2011) proposed a weight adjusted voting algorithm where a weight vector of classifiers and a weight vector of instances are used. Higher weihgt assigned by the instance weight vector to observations that are hard to classify. Larger weight assigned by the classifier weight vector to classifiers which provide better accuracy to classify instances. The final output of the classifier ensemble is determined by voting according to weight vector. The proposed weighted voting have been applied to bootstrap aggregation (Bagging). The performance has evaluated on twenty eight datasets. In general, the proposed weighted voting exceed the majority voting.

Various combination schemes have been proposed in the literature. Majority voting is the most popular and commonly used as a combiner. This is because the generality of this combiner that can combine any type of classifier. The disadvantage of this combiner, is that, it has no consideration of the strength of each classifier. Therefore, a weighted voting that considers the performance of each classifier should be considered.

2.6 Summary

The parallel topology is the most commonly used in combining classifier. In this topology each classifier can be independent and complement each other to form a diverse ensemble classifier. The input feature manipulation approach also generates a set of diverse classifier. One of these approaches is known as feature set partitioning technique. Each classifier in a set of classifier is trained on different projections of the original feature set. The advantages of this method are: (1) All of the information available in the original training set is used. There is no irrelevant feature was removed in the original training set. Irrelevant features do not need to be removed in combining classifiers, since these features may possibly contain important information, furthermore to provide additional information and comprehensive results. (2) Disjoint feature set partitioning reduces the search space, which is important and useful compared to a non-disjoint, thus disjoint feature set partitioning approach gives greater possibility for reducing the execution time. Since the learning algorithm on disjoint approach lowers the computational complexity. (3) This way can be used to overcome the dimensionality problem by partitioning the original set of features into several disjoint feature subsets. Thus, it requires a feature set partitioning algorithm for ensemble construction to determine the appropriate feature subsets and the ensemble size. The ant system algorithm is considered to partition feature subset because the ant system has been proven to solve SPP problems. Finally, in order to provide better results, it is necessary to combine this ant system with an appropriate combiner. The weighted voting is considered to combine the classifier outputs because the generality of this combiner that can combine any type of classifier and considers the performance of each classifier in combining classifier outputs.

CHAPTER THREE

RESEARCH METHODOLOGY

3.1 Introduction

This chapter presents the framework and methodology of this reseach to develop an improved multiple classifier combination scheme. Furthermore base classifier as a forming homogeneous classifier ensemble, the dataset description and evaluation measure are also presented in this chapter. The rest of this chapter is organised as follows: Section 3.2 discusses the research framework. Section 3.3 presents the base classifier that is used to construct classifier ensemble. Section 3.4 gives datasets' description which is used in experiments. Section 3.5 provides evaluation measures that are used in this study. Finally, section 3.6 summaries this chapter.

3.2 The Research Framework

The research framework is the roadmap of the research that aims to provide guidance to researchers for conducting research (Forrester, 2006). In accordance with the purpose of this research which is to develop an improved multiple classifier combination scheme, thus to achieve this goal, there are four phases of the research work. For every phase, there is an objective that will be achieved in this study. The research framework phases from the first to the end as depicted in Figure 3.1.

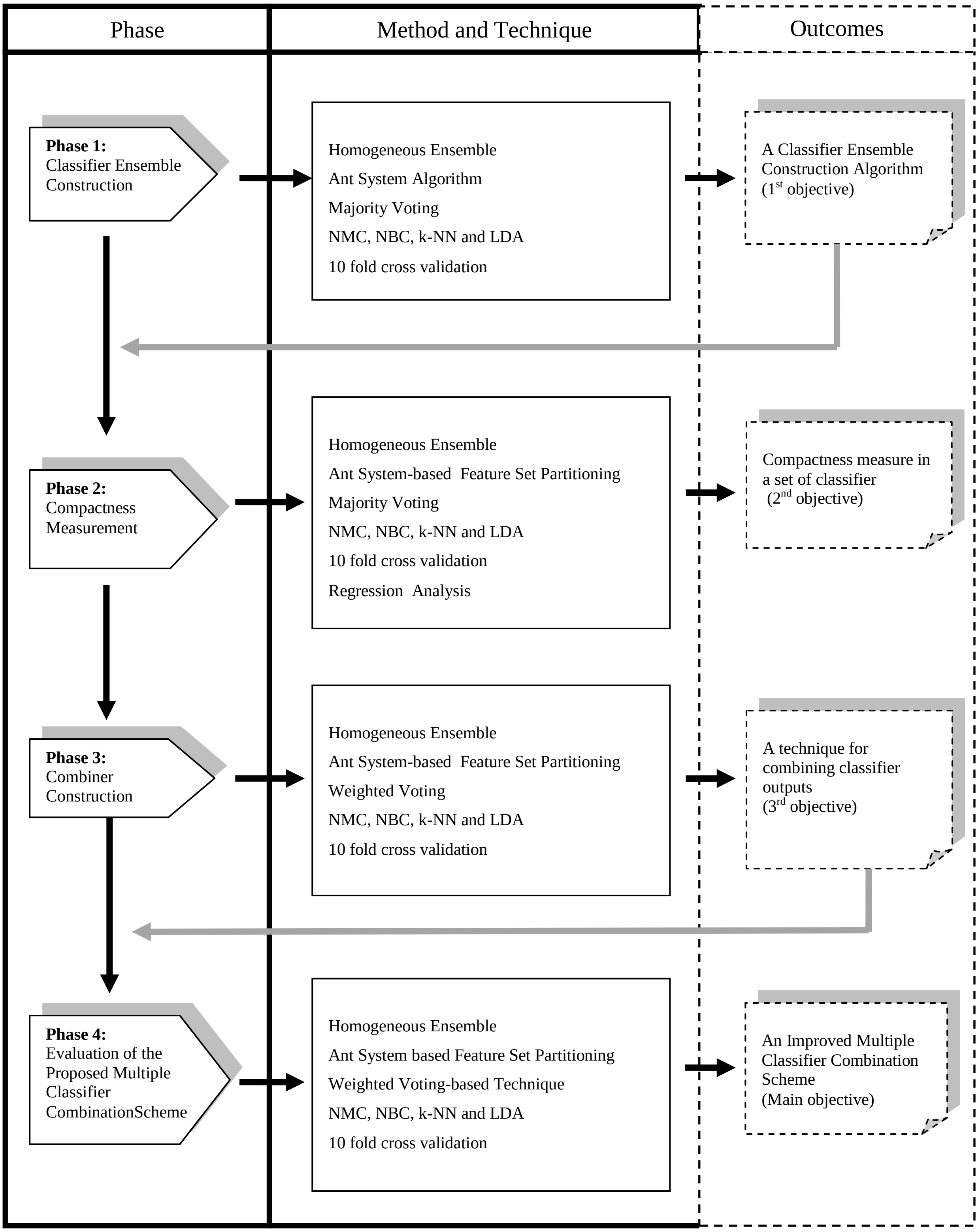

Figure 3.1 Research framework phases

The four main phases of the research work are: (1) develope an algorithm for classifier ensemble construction; (2) formulate compactness measure for a set of classifier; (3) construct a technique for combining classifier outputs; and (4) evaluate the improved multiple classifier combination scheme. Each phase of the framework has its own research method. The methods are described in the following section. For the evaluation experiments were conducted using benchmark datasets.

Multiple classifier combination consists of a set of classifiers (or classifier ensemble) and a combiner for combining classifier outputs (Wozniak et al., 2014). Figure 3.2 depicts the standard structure of the multiple classifier combination.

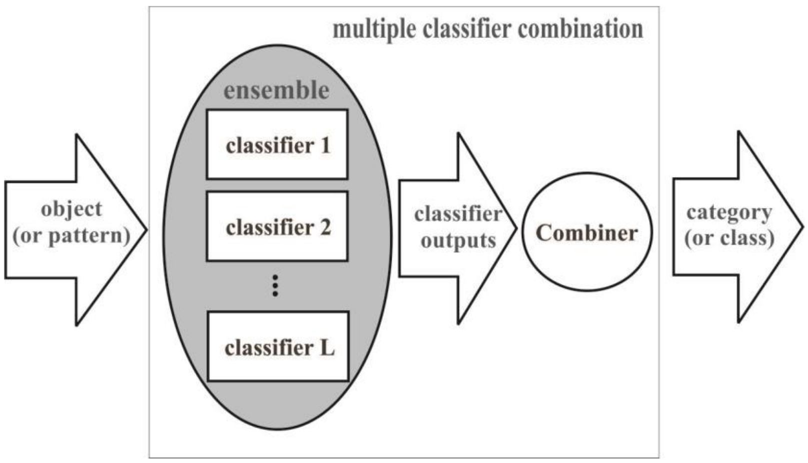

Figure 3.2 Standard structure of multiple classifier combination (Wozniak et al., 2014)

Design of a multiple classifier combination is to combine a set of classifier with a combiner. Standard multiple classifier combination design process includes three phases which are classifier ensemble construction, combiner construction and performance evaluation (Yang et al., 2004; Wozniak & Zmyslony, 2010). In this study, enhancements have been performed on the classifier ensemble construction and the combiner. In addition, a formulation to calculate the compactness is proposed. Furthermore the relationship of compactness in a set of classifier with the ensemble accuracy is tested.

3.2.1 Classifier Ensemble Construction

This section explains the research method to develop an algorithm in constructing classifier ensemble. The research method is conducted as follows: (1) The proposed ant system-based feature set partitioning algorithm was developed; (2) Homogeneous classifier ensemble with different features was constructed by proposed algorithm and majority voting technique is used as combiner; (3) Classification experiments on constructed classifier ensemble was carried out on several benchmark datasets; (4) The cross validation method was applied for prediction accuracy calculation; (5) Finally the experiment results were evaluated and compared to random partitioning method. The steps of the research method on classifier ensemble construction are depicted in Figure 3.3.

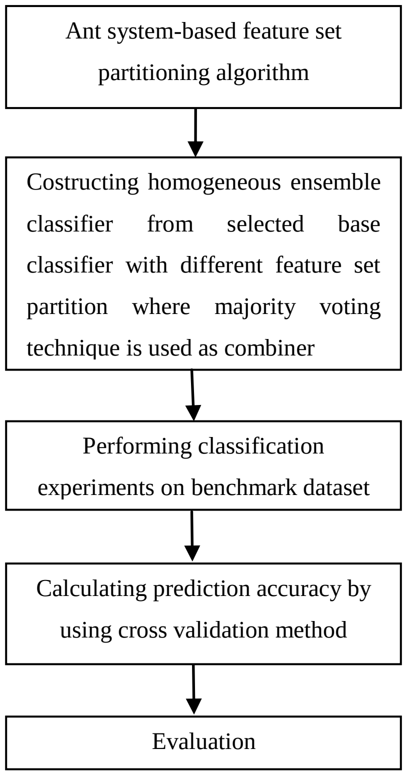

Figure 3.3 Steps of classifier ensemble construction

3.2.2 Compactness Measurement

This section explains the research method on compatnness measurement. The research method is conducted as follows: (1) The proposed compactness measure was formulated to measure diversity-accuracy in a set of classifier; (2) Classification experiments on benchmark dataset were performed by using constructed classifier ensemble, where standard majority voting is used as combiner; (3) The compactness value versus ensemble accuracy were calculated where cross validation was used for validation; (4) Finally, the relationship between proposed compactness measure and ensemble accuracy was evaluated by regression test. Figure 3.4 shows the experimental research method for compactness measurement.

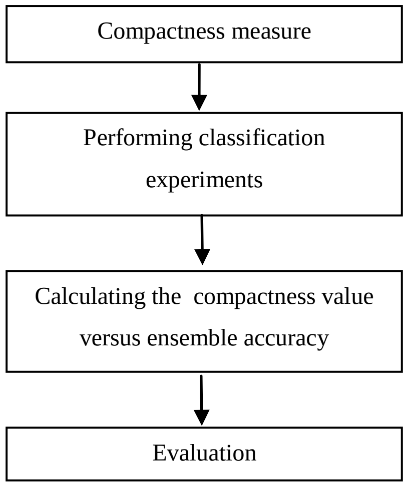

Figure 3.4 Steps of compactness measurement

3.2.3 Combiner Construction

This section explains the research method on combiner construction. The steps of research method are as follows: (1) A weighted voting technique was created as a combiner which will be suitable to combine several outputs of individual classifiers; (2) The proposed technique was applied to combine constructed classifier outputs; (3) Classification experiments were performed on benchmark datasets; (4) The accuracy of multiple classifier combination was calculated to test this proposed combiner and (5) Finally the experiment results were evaluated and compared to majority voting combiner. Figure 3.5 shows the experimental research method to create combiner.

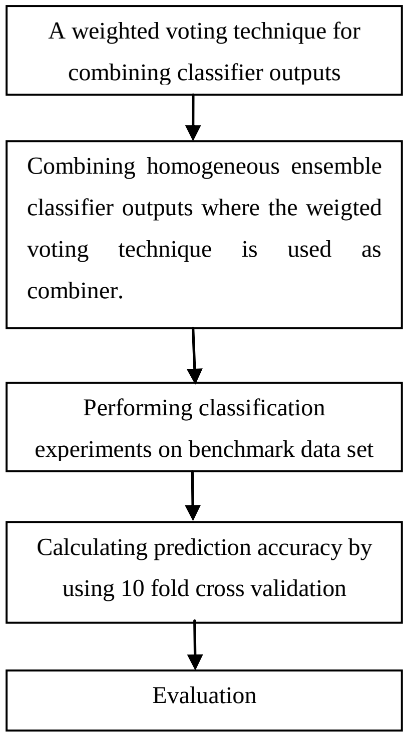

Figure 3.5 Steps of classifier combiner construction

3.2.4 Evaluation of the Proposed Multiple Classifier Combination Scheme

This section explains, the steps of research method to develop an improved multiple classifier combination scheme are described. The steps of research method are as follows: (1) The ant system-based feature set partitioning algorithm at ensemble construction part was used to build diverse and accurate ensemble; (2) The weigted voting technique at combiner part was used for dynamically combining multiple classifier outputs; (3) Classification experiments were performed on several benchmark datasets; (4) The accuracy of constructed multiple classifier combination was calculated to test the suitability of ant system-based feature set partitioning algorithm and weighted voting in the combination scheme and (5) Finally the performance of constructed multiple classifier combination by proposed combination scheme were evaluated and compared to the original single classifier and other multiple classifier combination on the same base classifier and dataset in terms of accuracy. Figure 3.6 shows the experimental research method for multiple classifier combination scheme development.

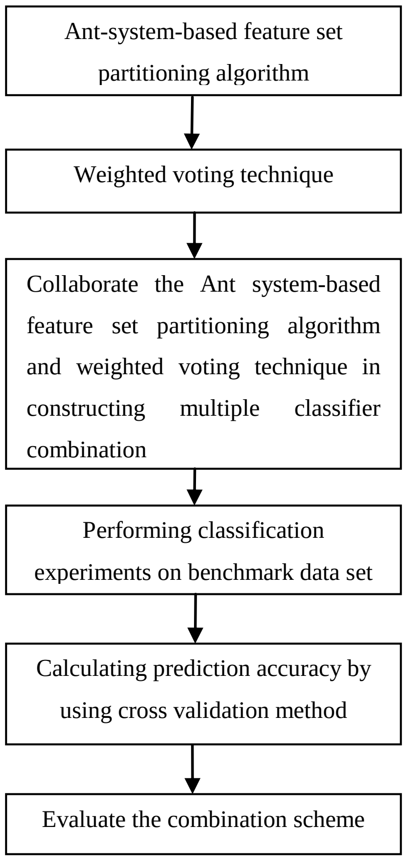

Figure 3.6 Steps of multiple classifier combination scheme development

In order to evaluate the proposed multiple classifier combination scheme, hence several experiments were performed to test their ability. For the simplicity of experiments without affecting the focus of research on developing multiple classifier combination scheme thus the homogeneous approach i.e., classifier combination formed using a single model type (or identical classifier models) (Kuncheva, 2001) was used. Four homogeneous ensembles is constructed by four single classifier which are nearest mean classifier (NMC), naïve bayes classifier (NBC), k-nearest neihbour (*k*-NN) and linear discriminant analysis (LDA) . Four single classifiers are not combined to form a better ensemble, but each classifier is used as base classifier to form homogenous ensemble with different feature subset. Therefore there are four homogeneous ensemble classifiers (e.g. multiple NMC combination, multiple NBC combination, multiple *k*-NN and multiple LDA combination). Figure 3.7 shows the four homogeneous ensembles were constructed to evaluate the proposed multiple classifier combination scheme.

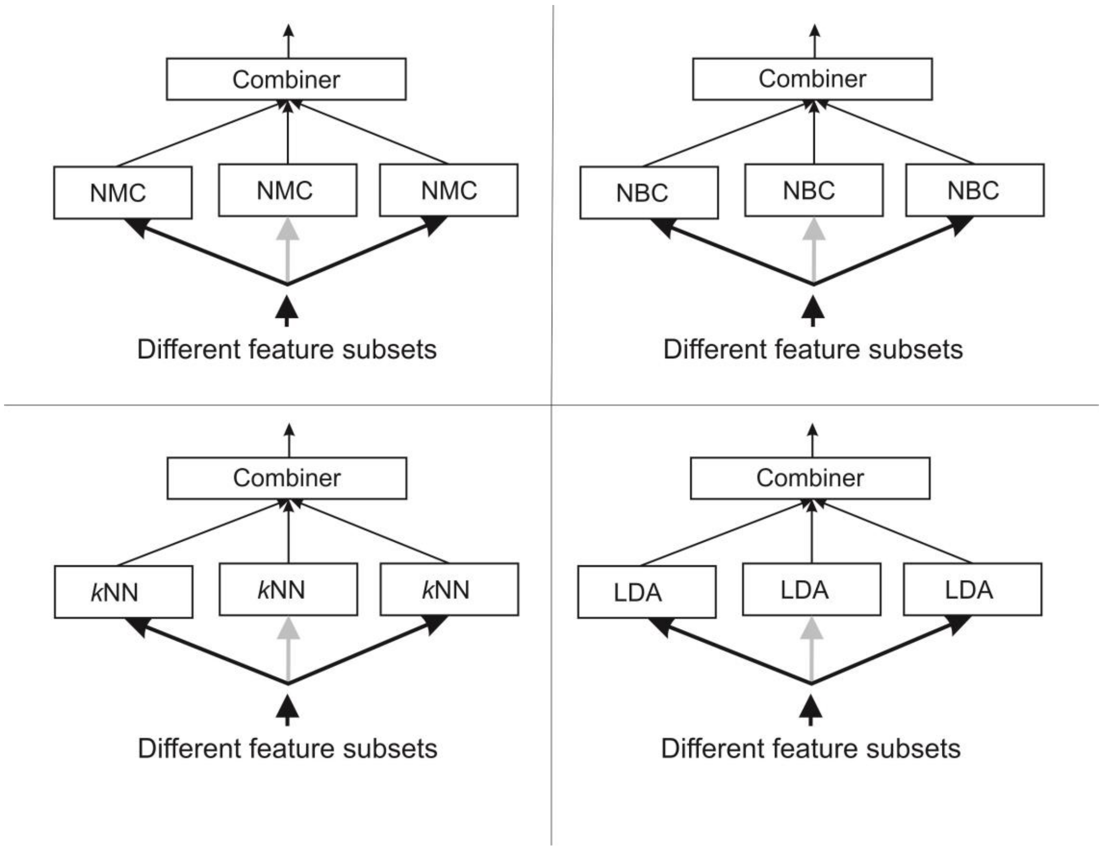

Figure 3.7 The four homogeneous ensembles for combination scheme evaluation

3.3 Base Classifier Description

This section presents a brief description of base classifier that are used in the experiments. Four kinds of single classifiers are selected as base classfier. These classifiers are used in several previous experimental studies to construct a homogeneous ensemble classifier.

The Nearest Mean Classifier is a famous fast and simple classifier. The Nearest Mean Classifier has been effectively applied to various classification tasks and showed robust and powerful (Shin & Kim, 2009). Several experimental studies on ensemble diversity and accuracy used the NMC (Skurichina et al., 2002; Kuncheva et al., 2002; Talha & Solung, 2013).

Naïve bayes classifier (NBC) is useful, efficient for solving classification problems (Farid et al., 2013). Naive Bayes classifier is frequently used in studying classifier ensemble (Hongbo & Yali, 2008; Zanda, 2010; Bolon-Canedo et al., 2012). Simulation results show that the ensemble of naive bayes has considerably better accuracy than several classification algorithms such as the Support Vector Machine and artificial neural networks based on cancer dataset (Margoosian & Abouei, 2013).

The *k*-nearest neighbour classifier (*k*-NN) is one of the most commonly used in constructing classifier ensemble and it is easy to find in the literature (Tahir & Smith, 2010; Hamzeloo et al., 2012; Parvin & Parvin, 2012; Ko & Sabourin, 2013). Thus the performance of the proposed ensemble construction method can be evaluated by comparing it with other ensemble methods**.**

Linear discriminant analysis is a strong and stable classifier, which means that small changes in the training set could not cause large changes in the classifier output. However, it is necessary to test the ability of the method to induce diversity for the improvement of LDA performance. LDA is one of widely-used techniques in pattern classification. Several experimental studies on ensemble diversity and accuracy used the LDA (Kong et al., 2005; Wang et al., 2012; Fook et al., 2013).

3.4 Dataset Description

This section presents a brief description of several benchmark datasets that are used in the experiments. A collection of 9 (nine) datasets taken from UCI repository are used in the experiment to test and evaluate the performance of the improved multiple classifier combination (MCC). The datasets involved are haberman, iris, lenses, liver, ecoli, pima indians diabetes, tic-tac-toe, glass, breast cancer (Wisconsin). A summary of datasets used are presented in Table 3.1.

Table 3.1

Summary of Datasets Used in the Experiments

No.	Datasets	Number of Instances	Number of Classes	Number of Features	Features Types
1	Haberman	306	2	3	Integer
2	Iris	150	3	4	Real
3	Lenses	24	3	4	Categorical
4	Liver	345	2	6	Categorical, Integer, Real
5	Ecoli	336	8	7	Real
6	Pima Indians Diabetes	768	2	8	Integer, Real
7	Tic-Tac-Toe	958	2	9	Categorical
8	Glass	214	6	9	Real
9	Breast Cancer (Wisconsin)	699 (683)	2	9	Categorical

The haberman dataset contains the survival status of the patients who had undergone breast cancer surgery. This dataset has 306 instances (or samples) each of which has 3 input features and 2 output classes. The values of feature 1 ranges from 30 to 83, feature 2 from 58 to 69, and feature 3 from 0 to 52. This is associated with the age of the patient at the time of operation, patient's year of operation and the number of positive axillary nodes detected. The output is in categorical form, which is either 0 or 1. This is indicated to survival status (1 = the patient survived 5 years or longer and 2 = the patient died within 5 years). For this data, 225 patients survived more than 5 years and 81 patients died within 5 years. In other words, 74.5 % of instances are 0 whereas 26.5 % of instances are 1.

The iris dataset or Fisher's iris data consists of 50 instances from each of three classes of *iris* plant (*setosa*, *virginica* and *versicolor*). There are no missing values. Four features were measured from each instance. These features are the sepal length in cm, sepal width in cm, petal length in cm and petal width in cm. Based on the combination of the four features, most ensemble methods can distinguish among the irises. All four features for this dataset are continuous (real).

The lenses dataset contains 24 instances of ophthalmic data analysis, to predict an appropriate lens by determining relevant features of the patient. The dataset has 4 input features which are: Age of the patient: (1) young, (2) pre-presbyopic, (3) presbyopic; Spectacle prescription: (1) myope, (2) hypermetrope; Notion on astigmatic: (1) no, (2) yes; Tear production rate: (1) reduced, (2) normal. There are 3 kinds of classes that give the appropriate lens prescription for patient which are: (1) the patient should be fitted with hard contact lenses, (2) the patient should be fitted with soft contact lenses; (3) the patient should not be fitted with contact lenses. The class distribution of this dataset: 4 patients with hard contact lenses, 5 patients with soft contact lenses and 15 patients who do not need lenses.

The liver dataset was originally created by BUPA Medical Research Ltd. This dataset has 345 instances and 6 features which are taken from blood tests. The features are: mean corpuscular volume, alkaline phosphotase, alamine aminotransferase, aspartate amino transferase, gamma glutamyl transpeptidase, number of half-pint equivalents of alcoholic beverages. The selector field is used to split data into two sets. This dataset was separated into two classes which are whether liver disorder exists or not.

The ecoli dataset are characterised by features calculated from the amino acid sequences. This dataset consists of 768 instances. The class distribution of this dataset: 143 patterns of cytoplasm (cp), 77 of inner membrane without signal sequence (im), 52 of periplasm (pp), 35 of inner membrane without uncleavable signal sequence (imU), 20 of outer membrane without lipoprotein (omL), 5 of outer membrane with lipoprotein (omL), 2 of inner membrane without lipoprotein (imL) and 2 patterns of inner membrane with cleavage signal sequence (imS). All 7 features for this dataset are the numerical (continuous) type.

The pima indian diabetes dataset is a collection of medical diagnostic reports of 768 examples from a population living near Phoenix, Arizona, USA. The patients in the dataset are females aged at least twenty-one (21) years old. The problem is to predict whether a patient would test positive for diabetes given a number of physiological measurements and medical test results according to World Health Organisation criteria. The original dataset consists of 768 instances and divided into two classes: tested positive for diabetes (268 examples) and not tested positive for diabetes (500 examples). All 8 features for this dataset are the numerical type.

The tic-tac-toe dataset encodes the complete set of possible board configurations at the end of tic-tac-toe games. There are 958 instances divided into 2 classes: negative (332 instances) and positive (626 instances). All 9 features for this dataset are nominal or categorical type.

The glass dataset is used to differentiate seven kinds of glasses by the basic materials such as iron, silicon, calcium and aluminum. The categories include vehicle windows non float processed, vehicle windows float processed, building windows non float processed, building windows float processed, tableware, headlamps and containers. All nine features for this dataset are of numerical type.

The breast cancer (Wisconsin) dataset is one of the breast cancer databases at UCI, collected at the University of Wisconsin. *Breast cancer wisconsin (Original) Dataset:* This is the breast cancer dataset. The objective is to predict the class using 9 input features. This dataset has 699 instances each of which has 9 categorical features. There are two classes: benign (458 examples) and malignant (241 examples). This dataset has 16 instances with missing values which should be removed from the dataset. Finally, this dataset has 444 benign instances and 239 malignant instances, thus in total there are 683 instances.

3.5 Evaluation Measures

Any classification prediction results obtained must be evaluated for determining the classification performance. There are several standard methods for evaluation. Only the measures used in this study will be presented as follows.

3.5.1 Cross Validation

The k-fold cross validation (Kohavi, 1995) was used to estimate the classification accuracies. A set of labeled samples from dataset randomly partitioned into k disjoint folds of equal size. Then, one of the k folds is randomly selected as the testing set and (k - 1) the remaining folds as the training set with the assumption that there is at least one sample per class.

3.5.2 Classification Accuracy Measurement

The classification accuracy (ac) is the ratio of numbers of all correctly classified instances and the total number of instance using the following formula:

$$Acc = \frac{no. of\ all\ correctly\ classified\ instances}{total\ number\ of\ instance} * 100\% \tag{3.1}$$

Finally, the procedure calculates the estimation of classification accuracy, by dividing the total of all classification accuracies by the total number of folds or rounds. Now the accuracy of cross validation estimation is defined as follows:

$$Acc_{CV} = \frac{1}{k}\sum_{i=1}^{k} acc_i \tag{3.2}$$

where acc_i is the classification accuracy of round *i* and *k* is the number of folds. A common *choice* for *k*-fold cross validation is k=*10.* Extensive experiments have shown that ten (10) is the best choice to get an accurate estimate (Yang & Browne, 2004; Wozniak, 2008; Wozniak, 2009) Therefore in this research, the experiments are conducted on the 10 fold cross-validation method. Figure 3.6 illustrates the use of 10 fold cross-validation method.

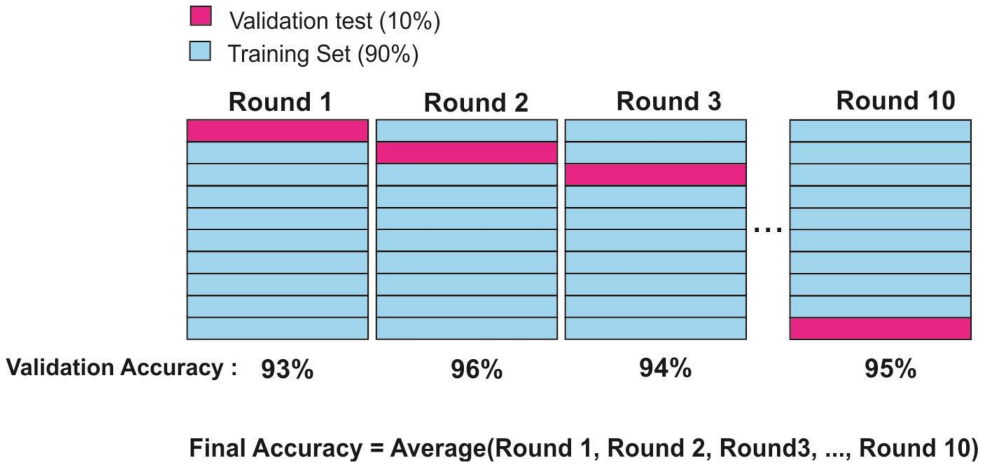

Figure 3.8 The 10-fold cross validation method

Figure 3.8 shows that each round a certain accuracy was obtained for example 93%, 96%, 94%, 95% and onwards until the tenth round. However final accuracy was obtained by averaging the accuracy of the whole round.

3.6 Summary

The main objective of this research is to develop an improved multiple classifier combination scheme for pattern classification. To achieve the main objective, a framework has been proposed to guide the study. Experimental research method has been adopted in conducting this study. Four selected single classifiers were used as base classifier to construct homogeneous ensemble classifier. The experiments are conducted on several UCI benchmark datasets. The cross validation method is used to test and evaluate the performance of the improved multiple classifier combination scheme. Finally the performance of developed multiple classifier combination scheme were evaluated and compared to the original single classifier and other multiple classifier combination on the same base classifier and dataset in terms of accuracy.

CHAPTER FOUR

ANT SYSTEM-BASED FEATURE SET PARTITIONING FOR CLASSIFIER ENSEMBLE CONSTRUCTION

4.1 Introduction

One of the approaches that have been used to create a set of diverse classifier is the input feature manipulation. The characteristic of input feature manipulation approach is to train each classifier on different feature subsets (usually, the same classifier is used). Feature set partitioning is a technique that manipulates the input feature set in creating the ensemble. Figure 4.1 depicts the general framework of feature set partitioning. However, it is difficult to determine how to partition the feature set to several subsets which may lead to a better classifier ensemble.

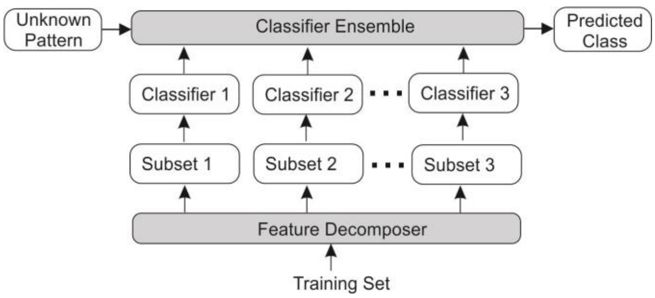

Figure 4.1 General framework of feature set partitioning (Maimon & Rokach, 2005)

This chapter presents ant system-based feature set partitioning algorithm in constructing classifier ensemble. In this algorithm, feature partition on the training sets is performed in order to build a classifier ensemble. Each classifier is trained on different partitions of features. All available features in the training set are utilized. There are no irrelevant features in the training set that are removed. The irrelevant features do not need to be removed in combining classification algorithm, since this removed feature may contain important information (Wang et al., 2005).

4.2 Proposed Ant System-based Feature Set Partitioning Algorithm

An improved algorithm is developed to train each classifier on different feature set partition using ant colony optimisation which leads to a better classifier ensemble. In this proposed algorithm namely Ant System-based Feature Set Partitioning (ASFSP), classifier ensemble is constructed based on feature set partitioning technique. Feature set on the training set is partitioned into different feature subset. There is no feature in the training set is eliminated. Furthermore each classifier in the ensemble is trained on a different projection of the original training set to induce diversity. The flowchart of the generic ant system-based feature set partitioning for classifier ensemble construction is provided in Figure 4.2.

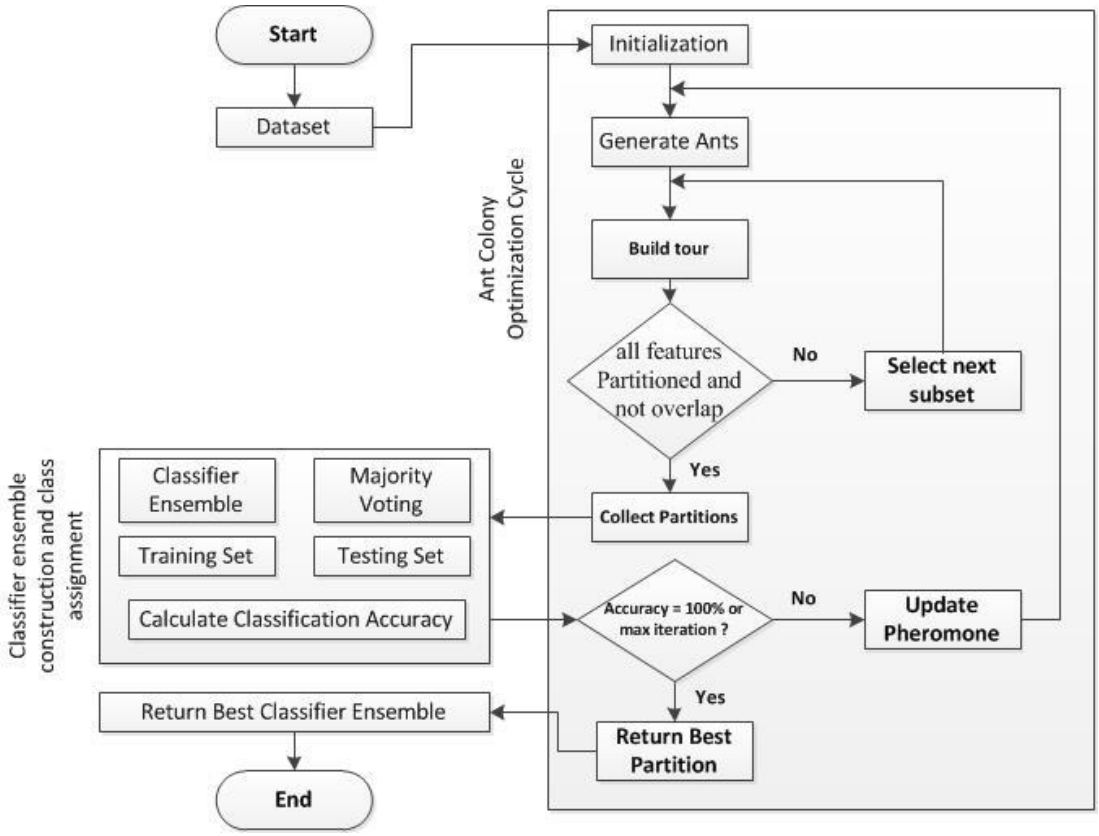

Figure 4.2 Flowchart of the generic ant system-based feature set partitioning algorithm

In the implementation process of ACO, the required inputs are features in dataset. The pheromone table is initialized followed by the generation of the ants. Each ant then builds a tour in the form of a feature partition which is considered as a possible solution. The tour is evaluated if it contains all the features and no overlap features. Otherwise the next feature subset is selected until the feature partitions have been collected. This will be done repeatedly until a possible solution is built. Furthermore partitioned feature is used to construct classifier ensemble. The class assignment is performed using constructed classifier ensemble by using majority voting combiner. The best partition will be formed if classification accuracy reaches 100% or the maximum iteration limit has been reached. The pheromone is then updated and another ant is generated if any criterion is not fulfilled. The whole process is repeated until the best partition is formed. The number of classifiers is determined by the number of feature set partitions are formed. Figure 4.3 presents the generic pseudocode of proposed algorithm.

```
%Input      : Features in Dataset
%Output     : Best Feature Partition, classification accuracy
Begin
[b,a]=loaddata('dataset.xxx');%load features in dataset
[n nod d h]=generate_problem(a) % generate graph problem
[t,iter,alpha,beta,rho,m,el]=initialization(n); %initialization
for i=1:iter
      [app]=generate_ants(m,n) %generate ants
      [tabu]=build_tour(app,m,n,nod,h,t,alpha,beta)%build tour
      [clust]=konversi(tabu)%build tour
      [jalur]=subtitutes(nod,clust)%collect partition
      [jalur error accuracy]=ensemble_accuracy(b,a,jalur) % class assignment using ensemble classifier
      [maxaccuracy(i),number]=max(accuracy)
      besttour(i,:)=jalur(number,:)
      if max(accuracy)==100
         break
      end
      [t]=ants_traceupdating1(t,clust,accuracy,rho);%update pheromone
 end
[k,l]=max(maxaccuracy)
accuracy=k
best_partition=[{besttour{l,:}}]%return best partition
End
```

Figure 4.3 Generic pseudocode for ant system-based feature set partitioning

Kuncheva and Whitaker (2001) showed the ability of feature subspace method to improve performance in multiple classifier combination. In their study the set of features is partitioned and each subset is used by classifier in the team. Breast Cancer dataset from UCI repository was used in their experiments. All partitions of a set of 10 features are enumerated randomly into 3 subsets containing (4, 4, 2) features and (4, 3, 3) features. The chances of the ensemble classifier outperforms the single best classifier if the feature space is partitioned randomly has been showed. In this study, the improvement of the ensemble classifier performance is obtained by using non-randomly feature space partitioning. Ant colony optimisation was used in the proposed combination scheme for determining feature subsets to produce the best performance.

Santana et al. (2010) have applied and compared ACO and GA in constructing classifier ensemble. Experiments conducted by constructing heterogeneous ensembles that used three classifier which are *k*-nearest neighbour (*k*-NN), decision tree (DT) and neural network (NN). Based on its structure, the ensemble can be divided into two approaches i.e: heterogeneous and homogeneous. In the first approach, different types of classifiers are combined as an ensemble. On the other hand, for the second approach, same types of classifiers are trained by different feature subsets to construct an ensemble (Kuncheva, 2001). In their experiments, heterogeneous ensemble structure is constructed using different training datasets (feature selection technique) and the ensemble size is predetermined. In this study, homogeneous ensemble structure is constructed by training the classifiers with different feature subsets (feature partitioning technique) and the ensemble size is not specified. Although both of these studies used the ACO-based algorithm, ACO-based algorithm is used to perform feature selection technique, where several features may be removed. In this study ACO-based algorithm is used to perform feature partitioning technique where no features are removed.

ACO in the context of ensemble and partitioning has been introduced by Parvin and Bidgoli, (2011) and also by Parvin and Beigi, (2011). Both research used of ACO-based algorithm and ensemble approach to partition data. The purpose of this research is different from those studies. This study aims to propose a classifier ensemble construction where ACO-based partitioning technique is used to form different feature subsets. Different feature subset is used to train several single classification algorithms to generate diversity (Kuncheva, 2001). Their research aims to propose a consensus function to aggregate a set of partitions to partition data. Different partitions were produced by several single clustering algorithms which is also called the cluster ensemble (Strehl and Ghosh, 2002). The approach used in this study is vertical partitioning of dataset while their study used horizontal partitioning of dataset. In vertical partitioning, the dataset is partitioned into a number of datasets that have the same number of instances samples as the original dataset, each containing a subset of the original feature set. In horizontal partitioning the dataset is partitioned into several datasets that have the same features as the original dataset, each containing a subset of the instances in the original dataset (Rokach, 2010). However, The ACO can automatically determine the number of partitions generated. So it can determine the number of clusters and number of classifier accordingly. Table 4.1 below shows the comparison of both studies.

Table 4.1

The Comparison of Both Studies

This study	Parvin et al.,
Classifier ensemble approach	Clustering ensemble approach
Vertical partitioning approach	Horizontal partition approach
To construct a set of classifier	To aggregate a set of partitions
Automatically determine the number of classifier	Automatically determine the number of cluster

4.3 Experiments on Classifier Ensemble Construction

Four sets of experiments were conducted to apply the proposed ant system-based feature set partitioning algorithm in constructing classifier ensemble by using MATLAB. The homogeneous classifier ensemble approach (Kuncheva, 2001) is constructed to simplify the experiments without affecting the objective of the study. Four homogeneous classifier ensembles with different feature subsets which were constructed are homogeneous NMC ensembles, homogeneous NBC ensembles, homogeneous *k*-NN ensembles and homogeneous LDA ensembles. For purposes of comparison, the standard majority voting combiner (Hansen & Salamon, 1990) is used in the experiments. The majority voting is the most commonly used combiner to combine classifier outputs (Bay, 1998; Bryll et al., 2003; Li & Sun, 2009; Bolon-Canedo et al., 2012; Hajdu et al., 2013). Ant system as a particular version of ant system-based algorithm is used as a case study in the experiments. This original and most famous algorithm in the family of ACO has been used and proven to solve SPP problems (Crawford et al., 2009; Lizárraga et al., 2013) and various optimisation problems (Dorigo & Gambardella, 1997; Shang & Wang, 2010; Jevtic et al., 2010). The experiments are performed on nine (9) benchmark datasets from UCI machine learning repository to test the performance of constructed classifier ensembles.

To obtain powerful performance estimation and comparisons, a large number of estimates are always preferred. In k-fold cross-validation, only k estimates are obtained. A commonly used method to increase the number of estimates is to run k-fold cross-validation multiple times. Repeating 10 (Ten) times of the 10-fold cross-validation approach is used in testing classification accuracy. Several recent studies used ten runs of 10-fold cross validation to obtain performance estimation (Burton et al., 2012; Medrano-Gracia, et al., 2013; Zhao et al., 2013; Fook et al., 2013). The following subsections present the experimental results.

The goal was to empirically evaluate the performance of proposed ant system-based feature set partitioning algorithm in terms of classification accuracy. For this purpose, the four sets of experiments proceed in two treatments. In the first treatment, random subspace (Ho, 1998; Skurichina & Duin, 2002; Tao & Tang, 2004; Serpen & Pathical, 2009; Li et al., 2013) applied to construct classifier ensemble. The random subspace assigns a random subset of features to train individual classifiers and also using the majoriting voting combiner. In random subset approach each classifier is trained with random subsets of the original feature set. This process is repeated to produce an ensemble with a number of classifiers. In this study, a number of classifier is set with the aim to see its influence on the accuracy of the ensemble. In the second treatment, the proposed algorithm applied to construct classifier ensemble. During the experiments, the average of the classification accuracy of ensemble classifier which is constructed by the ant system-based feature set partitioning is compared to the average of the classification accuracy of ensemble classifier which is constructed by the random subspace.

4.3.1 Experimental Results on NMC Ensembles

Experiments were conducted to test the ant system-based feature set partitioning in constructing homogneous NMC ensembles. Tables 4.2 and 4.3 depict the average and standard deviation of the classification accuracies of constructed homogeneous ensembles based on RS and ASFSP respectively.

Table 4.2

Classification Accuracy of NMC Ensembles by RS

Experiment #	Haberman	Iris	Lenses	Liver	Ecoli	Pima	Tic-Tac-Toe	Glass	Breast Cancer
1	67.97	90.00	70.83	55.36	80.95	67.58	66.08	44.39	96.34
2	73.86	92.67	62.50	60.87	82.74	69.53	63.67	43.93	96.34
3	69.93	92.00	75.00	54.78	80.36	65.89	62.42	43.93	96.63
4	73.53	92.67	66.67	55.94	82.74	64.32	63.36	46.73	96.63
5	58.82	90.67	66.67	57.68	80.36	70.31	65.76	42.99	96.34
6	71.90	93.33	70.83	57.10	80.36	63.02	63.57	44.39	96.63
7	71.90	92.67	66.67	55.65	82.74	69.92	66.49	44.86	96.49
8	69.61	92.00	70.83	55.94	81.85	72.27	64.41	44.39	96.49
9	70.92	90.67	54.17	55.07	81.85	68.75	64.93	44.39	96.34
10	74.84	94.00	58.33	55.94	82.74	67.19	64.20	44.39	96.78
Average	**70.33**	**92.07**	**66.25**	**56.43**	**81.67**	**67.88**	**64.49**	**44.44**	**96.50**
Standard deviation	4.56	1.27	6.35	1.79	1.07	2.86	1.31	0.95	0.16

Table 4.3

Classification Accuracy of NMC Ensembles by ASFSP

Experiment #	Haberman	Iris	Lenses	Liver	Ecoli	Pima	Tic-Tac-Toe	Glass	Breast Cancer
1	69.93	94.67	70.83	64.06	81.25	73.05	73.17	52.80	97.22
2	71.57	94.67	66.67	62.61	82.14	72.79	72.76	53.27	97.22
3	71.24	94.67	62.50	64.35	81.25	72.53	72.65	55.14	97.22
4	73.86	94.67	70.83	65.51	82.74	72.92	74.01	52.80	97.22
5	67.97	95.33	70.83	64.64	81.25	73.70	72.86	52.34	97.22
6	68.63	94.67	62.50	64.35	82.74	73.18	72.86	52.34	97.36
7	69.93	94.67	70.83	63.77	82.14	72.53	72.86	52.80	97.22
8	70.92	94.67	58.33	64.35	82.14	73.96	72.65	53.27	97.22
9	69.28	92.00	66.67	64.64	81.25	72.79	73.28	54.21	97.22
10	70.59	94.67	66.67	64.64	81.25	72.79	72.96	53.27	97.22
Average	**70.39**	**94.47**	**66.67**	**64.29**	**81.82**	**73.02**	**73.01**	**53.22**	**97.23**
Standard deviation	1.67	0.89	4.39	0.75	0.63	0.47	0.41	0.87	0.04

Based on the experimental results, it can be seen that a small deviation of the classification accuracies was obtained which showed that the experiments were accurate and good. The average of accuracy of the newly constructed homogeneous NMC ensemble by the proposed algorithm is compared with the average of accuracy of constructed homogeneous NMC ensembles by the random subset approach. The comparison between ant system-based feature set partitioning and random subspace to construct homogeneous NMC ensembles is shown in Table 4.4 and in Figure 4.5.

Table 4.4

Comparison of RS and ASFSP in Constructing NMC Ensembles

Dataset	**Classifier Ensemble Construction**					
	Random Subspace			**Ant System-based Feature Set Partitioning**		
	Average of Accuracy (%)	**Feature Subset**	**# of classifier**	**Average of Accuracy (%)**	**Feature Partition**	**# of classifier**
Haberman	70.33	[1 3][1 2 3][[1 2][3]	4	70.39	[1][2 3]	2
Iris	92.07	[1 2 3 4][1 2 3][1 3 4][3]	4	94.47	[1][2 3][4]	3
Lenses	66.25	[2 3 4][1 3][1 3 4][2]	4	66.67	[1 2 3 4]	1
Liver	56.43	[1 4 5][5 6][1 2 3 4 5 6][2 3 4 5]	4	64.29	[1 2 4 6][3][5]	3
Ecoli	81.67	[2 5 7][1 2 4 5 7][1 3 7][2]	4	81.82	[1 2 3 4 5 6 7]	1
Pima	67.88	[1 2 3 6 7 8][3 4 5][3 6 7][2 3 5 6]	4	73.02	[3 4 5 7][1 6][8][2]	4
Tic-Tac-Toe	64.49	[1 4 5 9][1 2 5 6][2 3 5 6][1 2 3 4 5 7 8 9]	4	73.01	[2 4 5 8][7][3 6 9][1]	4
Glass	44.44	[2 3 5 6 9][3 7 8 9][1 2 3 4 8 9][1 3 5 6 8]	4	53.22	[2 3 5 7][1 4 8 9][6]	3
Breast Cancer	96.50	[6 7 8 9][3 4 6 8][1 2 3 4 6 7 8][1 4 5 8]	4	97.23	[1 2 3 4 5 7 9][6 8]	2

Investigation of these combination methods in building ensemble has been performed. Table 4.4 shows the representation of feature subset using both method. The two combination methods are different in forming partitions. It can be seen from the results obtained. In random subset, the features are randomly selected with replacement thus feature subsets can be overlap. Several features are also possible not selected. While on ant system-based feature set partitioning all the features are used and no features that overlap. In both methods, partitions or feature subsets formed are used in building ensemble. The number of feature subset or partition indicates the number of classifier in the ensemble. The number of classifier on a random subset method specified beforehand, while the number of ant system-based classifier algorithm automatically determined. In random subspace experiments, the number of classifier in homogenous ensemble sets of four (4). This number represents the maximum number of partition formed by ant system-based feature set partitioning. This number is chosen to indicate whether a greater number of classifier gives good results.

The Comparison of both method can also be seen in Table 4.4. It is intended to determine which is better between these methods. Both of these methods adapted the input feature manipulation approach (Roli, 2009). Random subspace is chosen because it is a standard technique in this approach. Nevertheless random subspace ensemble method is popular and widely used by previous studies to build an ensemble classifier (Serpen & Pathical, 2009; Li et al., 2013). Based on the results, the usage of random feature subsets provide lower accuracy, although the classifier set number large. Ant system-based feature set partitioning algorithm successfully deliver better results despite a smaller number of classifier. In some datasets it can seen that ant system-based feature set partitioning algorithm does not partition the feature. This means that this algorithm will choose the single best classifier, instead of an ensemble classifier. Unlike the random subset that the decision to build the ensemble has been defined previously and predetermined number of classifier ensemble is built. This is the advantages of the proposed algorithm. Accuracy obtained always exceed or at least equal to the best single classifier accuracy.

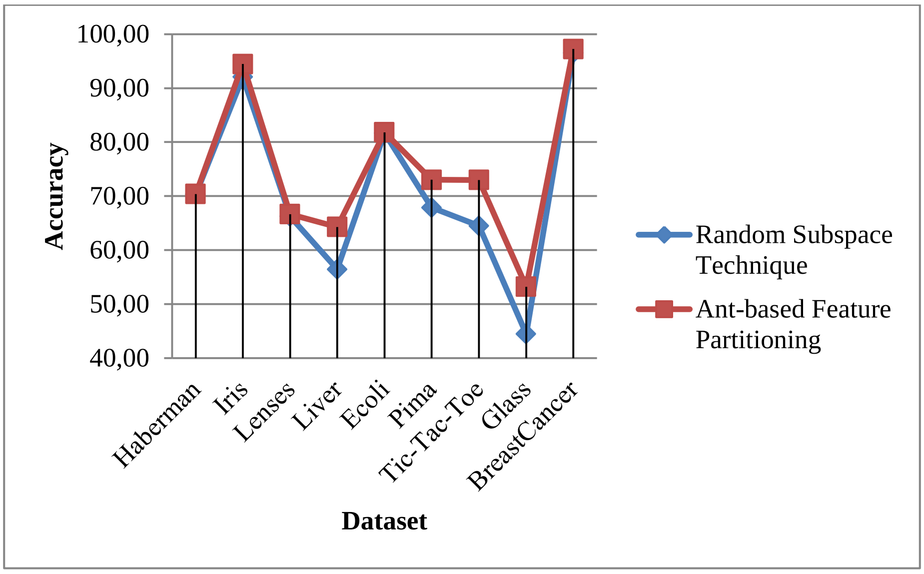

Figure 4.4 Comparison of RS and ASFSP in constructing NMC ensembles

Based on the experimental results, it can be seen that classification accuracy improvement can be obtained on all datasets. The ant system-based feature set partitioning successfully partitioned feature sets on several datasets. Obvious improvement accuracy is obtained for the feature sets that are successfully partitioned. While on the feature set is not partitioned tends accuracy is the same. Most of the datasets used successfully partitioned, which are haberman, iris, liver, pima, tic-tac-toe, glass and breast cancer. The use of ant system-based feature set partitioning can easily determine the optimal number of classifiers. Furthermore, although the number of classifiers is smaller, it can give better classification results.

4.3.2 Experimental Results on NBC Ensembles

Experiments were carried out to test the ant system-based feature set partitioning in building homogeneous NBC ensembles. Table 4.5 describes the average and standard deviation of the classification accuracies of constructed homogeneous NBC ensembles using random subspace. Table 4.6 shows the average and standard deviation of the classification accuracies of the constructed homogeneous NBC ensembles using ant system-based feature set partitioning.

Table 4.5

Classification Accuracy of NBC Ensembles by RS

Experiment #	**Haberman**	**Iris**	**Lenses**	**Liver**	**Ecoli**	**Pima**	**Tic-Tac-Toe**	**Glass**	**Breast Cancer**
1	75.16	94.67	62.50	58.26	75.25	75.91	66.91	73.89	96.19
2	74.84	92.00	62.50	58.55	75.25	75.91	72.03	73.35	96.19
3	74.84	93.33	62.50	61.16	75.25	73.83	67.01	73.89	95.61
4	74.51	96.00	62.50	62.03	75.25	75.91	66.70	73.35	96.49
5	73.53	95.33	62.50	64.64	75.25	76.30	66.60	72.35	96.63
6	74.84	96.00	62.50	58.55	75.25	75.91	68.48	73.35	95.90
7	75.16	95.33	62.50	59.13	75.25	75.78	68.89	72.35	96.05
8	73.86	96.00	62.50	57.39	75.25	75.91	68.89	73.35	96.34
9	74.51	93.33	62.50	60.58	75.25	75.65	66.81	72.35	95.61
10	74.84	96.00	62.50	60.87	75.25	75.91	71.09	73.89	96.34
Average	**74.61**	**94.80**	**62.50**	**60.12**	**75.25**	**75.70**	**68.34**	**73.21**	**96.14**
Standard deviation	0.53	1.43	0.00	2.18	0.00	0.68	1.93	0.64	0.35

Table 4.6

Classification Accuracy of NBC Ensembles by ASFSP

Experiment #	Haberman	Iris	Lenses	Liver	Ecoli	Pima	Tic-Tac-Toe	Glass	Breast Cancer
1	75.49	96.00	62.50	64.93	75.45	75.39	72.34	72.55	97.66
2	74.18	95.33	62.50	63.77	75.46	75.65	72.86	73.15	97.51
3	74.84	95.33	62.50	63.77	74.45	75.13	72.03	72.55	97.51
4	74.84	95.33	62.50	62.32	74.65	75.00	72.96	72.55	97.66
5	74.84	96.00	62.50	63.77	75.85	75.65	71.71	71.85	97.66
6	75.16	95.33	62.50	63.48	75.45	75.52	72.96	72.55	97.51
7	74.84	95.33	62.50	63.19	76.15	75.65	72.76	79.65	97.80
8	74.51	95.33	62.50	62.61	75.45	75.39	73.28	72.55	97.51
9	74.51	95.33	62.50	62.90	76.95	75.52	72.44	72.55	97.80
10	74.84	95.33	62.50	64.35	75.45	75.52	72.76	72.55	97.66
Average	**74.81**	**95.46**	**62.50**	**63.51**	**75.53**	**75.44**	**72.61**	**73.25**	**97.63**
Standard deviation	0.36	0.28	0.00	0.79	0.71	0.22	0.48	2.27	0.11

Furthermore, a comparison is also performed between the random subspace and ant system-based feature set partitioning in building homogeneous NBC ensembles. The comparison results are shown in Table 4.7 and in Figure 4.5.

Table 4.7

Comparison of RS and ASFSP in Constructing NBC Ensembles

Dataset	Classifier Ensemble Construction					
	Random Subspace			Ant System-based Feature Set Partitioning		
	Average of Accuracy (%)	Feature Subset	# of classifier	Average of Accuracy (%)	Feature Partition	# of classifier
Haberman	74.61	[1 2 3][1 2][2][3]	4	74.81	[1 2 3]	1
Iris	94.80	[1 3 4][2 4][1 2 4][2]	4	95.46	[1 2 3 4]	1
Lenses	62.50	[2 3 4][3 4][1 2 3][1 2 3 4]	4	62.50	[1 2 4][3]	2
Liver	60.12	[2 3][2 4 5 6][1 2 3 4 6][2 5]	4	63.51	[1 2 3 4][5][6]	3
Ecoli	75.25	[2 3 4 7][2 4 5 6][4 5][1 2 3 5]	4	75.53	[1 2 3 4 5 6 7]	1
Pima	75.70	[1 2 3 5 7 8][2 4 7 8][1 2 3 4][1 2 3 7]	4	75.44	[1 2 3 4 5 6 7 8]	1
Tic-Tac-Toe	68.34	[2 3 5 7 8 9][1 3 4 5 6 7][1 3 5 8 9][1 2 3 4 6]	4	72.61	[1 2 3 4 5 6 7 8 9]	1
Glass	73.21	[1 3 6][1 3 4 6 9][[3 8][2 4 5 7 8 9]	4	73.25	[1 2 3 4 5 6 7 8 9]	1
Breast Cancer	96.14	[1 3 4 7 8][5 6][2 3 4 6 8][2 6 9]	4	97.63	[4 5 8 9][1 2 7][6][3]	4

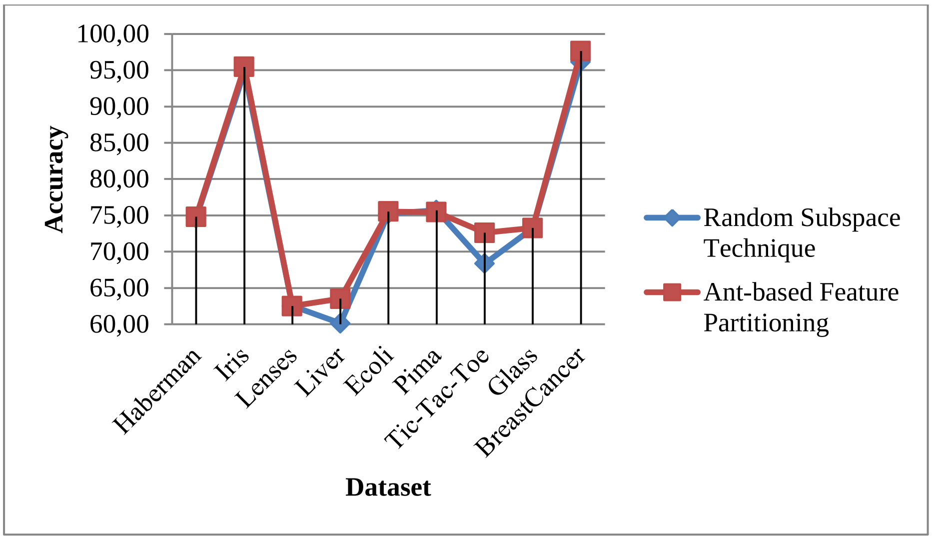

Figure 4.5 Comparison of RS and ASFSP in constructing NBC ensembles

Only a little improvement of classification accuracy is achieved when NBC is used as base classifier. The increase is only clearly seen in two datasets: liver and breast cancer. This is because the two datasets successfully performed partition on the feature sets. The lenses dataset successfully partitioned, with comparable classification accuracy, but at least does not reduce the level of accuracy. Particularly there is a significant increment in the liver dataset when the number of partition is 3. As in previous experiments, although with a smaller number of classifiers, this algorithm can compensate the random subspace which requires uncertain number of classifiers.

4.3.3 Experimental Results on *k*-NN Ensembles

Experiments were carried out to test the ant system-based feature set partitioning in constructing *k*-NN ensembles. Table 4.8 and Table 4.9 present the mean and standard deviation of the classification accuracies of constructed *k*-NN ensembles using random subspace and ant system-based feature set partitioning respectively.

Table 4.8

Classification Accuracy of k-NN Ensembles by RS

Experiment #	Haberman	Iris	Lenses	Liver	Ecoli	Pima	Tic-Tac-Toe	Glass	Breast Cancer
1	67.32	93.33	70.83	55.07	80.36	69.79	72.13	73.83	97.36
2	65.03	94.00	58.33	60.00	81.25	71.09	76.10	71.96	97.36
3	69.28	92.67	62.50	58.84	78.87	68.10	77.56	74.77	96.93
4	69.28	94.00	66.67	66.09	84.23	72.66	77.56	71.50	97.80
5	67.32	93.33	58.33	57.97	80.95	69.53	72.13	73.36	96.93
6	68.30	92.67	62.50	56.81	79.46	70.83	76.10	73.36	97.22
7	64.38	93.33	58.33	57.10	81.25	69.92	74.53	75.70	96.78
8	70.26	93.33	58.33	60.29	82.44	70.96	78.18	71.96	97.07
9	70.59	93.33	62.50	64.35	83.63	70.31	78.18	68.69	97.66
10	67.32	94.00	66.67	64.06	79.46	72.66	74.53	71.96	97.22
Average	**67.91**	**93.40**	**62.50**	**60.06**	**81.19**	**70.59**	**75.70**	**72.71**	**97.23**
Standard deviation	1.96	0.47	4.17	3.48	1.70	1.32	2.19	1.86	0.31

Table 4.9

Classification Accuracy of k-NN Ensembles by ASFSP

Experiment #	Haberman	Iris	Lenses	Liver	Ecoli	Pima	Tic-Tac-Toe	Glass	Breast Cancer
1	72.22	96.00	79.17	65.80	80.95	71.48	74.74	73.36	97.95
2	72.88	96.00	79.17	62.61	81.25	71.74	74.32	72.43	97.51
3	72.22	96.00	79.17	65.51	81.25	70.18	76.83	72.90	97.80
4	72.55	96.00	79.17	64.06	80.65	70.44	75.78	72.90	97.51
5	72.22	96.00	79.17	62.61	82.14	71.48	76.10	72.90	97.51
6	73.20	96.00	79.17	62.61	81.25	70.18	75.47	74.30	97.51
7	72.88	96.00	79.17	64.06	80.36	71.22	76.10	73.83	97.36
8	72.22	96.00	79.17	65.51	80.95	70.44	76.83	71.03	97.36
9	74.51	95.33	79.17	66.25	81.55	72.14	75.05	72.90	97.80
10	72.55	96.00	79.17	62.61	81.55	70.83	76.10	72.43	97.66
Average	**72.75**	**95.93**	**79.17**	**64.16**	**81.19**	**71.01**	**75.73**	**72.90**	**97.60**
Standard deviation	0.71	0.21	0.00	1.50	0.50	0.70	0.84	0.88	0.20

The ant system-based feature set partitioning successfully partitioned feature sets on liver, pima and breast cancer datasets. During the experiment, a comparison is also performed between the random subspace and ant system-based feature set partitioning in constructing homogeneous *k*-NN ensembles. The comparison result is shown in Table 4.10 and in Figure 4.6.

Table 4.10

Comparison of RS and ASFSP in Constructing k-NN Ensembles

Dataset	Classifier Ensemble Construction					
	Random Subspace			Ant System-based Feature Set Partitioning		
	Average of Accuracy (%)	Feature Subset	# of classifier	Average of Accuracy (%)	Feature Partition	# of classifier
Haberman	67.91	[1 2][3][1 3][2 3]	4	72.75	[1 3][2]	2
Iris	93.40	[3 4][2 4][1 4][1 2]	4	95.93	[1 2 3 4]	1
Lenses	62.50	[1][2 4][1 2 4][3 4]	4	79.17	[1 2 3 4]	1
Liver	60.06	[3][1 3 5 6][5 6][2 3 4 5]	4	64.16	[1 4 6][3 5][2]	3
Ecoli	81.19	[1 2 3 4 5 6 7][1 5][4 5 6][1 6]	4	81.19	[1 2 3 4 5 6 7]	1
Pima	70.59	[1 2 3 4 5 7][3][2 3 4 5 6 7 8][2 6 8]	4	71.01	[1 3 4 7][5 6 8][2]	3
Tic-Tac-Toe	75.70	[1 2 3 5 6 7][2 6 7 9][1 2 3 4 6 7 8 9][5 6]	4	75.73	[1 2 3 4 5 6 7 8 9]	1
Glass	72.71	[4 5 6 7][1 2 3 5 7 8 9][1 6 9][1 2 4 5 6 7]	4	72.90	[1 2 3 4 5 6 7 8 9]	1
Breast Cancer	97.23	[1 2 3 6][1 3 6][5 8 9][1 3 8 9]	4	97.60	[1 2 4 7 9][3 5][6][8]	4

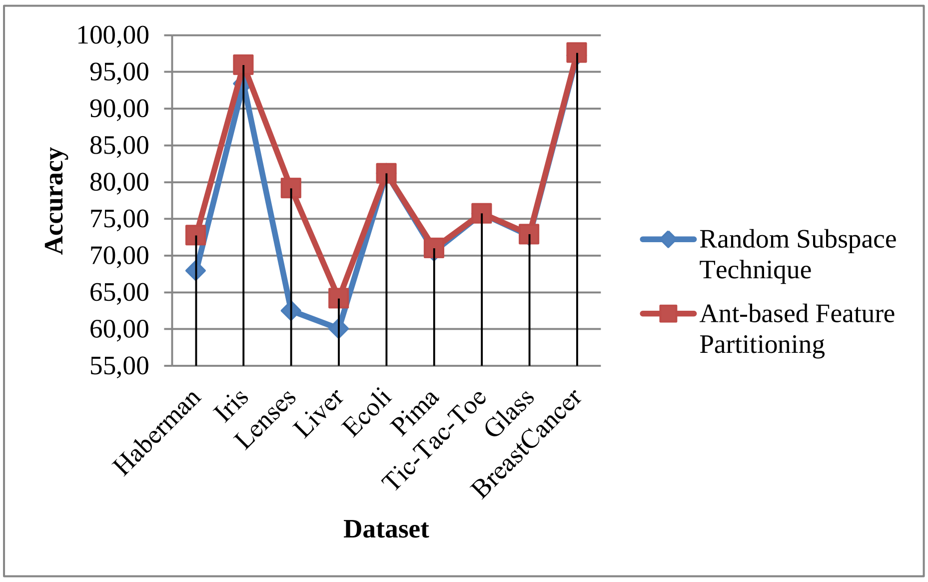

Figure 4.6 Comparison of RS and ASFSP in constructing *k*-NN ensembles

Improvement classification accuracy frequently appears when *k*-NN classifier is used as base classifier. It can be seen clearly in haberman, liver, pima and breast cancer datasets. Especially on the liver dataset, a significant classification accuracy improvement is obtained when the number of partition is 3 (three). On the other datasets, the accuracy of constructed homogeneous *k*-NN ensembles is at least not less than the accuracy of the random subspace. It is clearly that the use of ant system-based feature set partitioning effectively used to determine the number of classifiers.

4.3.4 Experimental Results on LDA Ensembles

Experiments were conducted to determine the ability of both random subspace and ant system-based feature set partitioning to construct homogeneous LDA ensembles. Table 4.11 and Table 4.12 present the mean and standard deviation of the classification accuracies of constructed homogeneous LDA ensembles respectively.

Table 4.11

Classification Accuracy of LDA Ensembles by RS

Experiment #	Haberman	Iris	Lenses	Liver	Ecoli	Pima	Tic-Tac-Toe	Glass	Breast Cancer
1	74.84	96.00	79.17	62.03	73.23	75.00	65.66	57.48	96.19
2	72.88	95.33	79.17	62.03	73.23	74.87	65.66	57.94	96.49
3	72.88	94.67	83.33	63.77	73.03	75.00	65.66	59.35	96.19
4	74.84	95.33	87.50	62.03	73.23	74.61	66.18	60.28	96.49
5	73.20	94.67	79.17	63.48	73.23	74.87	65.66	56.54	95.90
6	73.20	97.33	79.17	62.61	74.41	75.39	64.72	59.81	96.19
7	72.88	96.00	79.17	60.58	73.23	73.44	65.66	59.81	96.49
8	74.84	96.67	79.17	63.48	73.00	75.13	64.72	60.75	95.90
9	74.84	95.33	83.33	60.58	73.23	75.13	66.18	59.81	96.05
10	73.20	96.00	75.00	63.77	72.99	75.91	66.18	60.28	96.19
Average	**73.76**	**95.73**	**80.42**	**62.44**	**73.28**	**74.94**	**65.63**	**59.21**	**96.21**
Standard deviation	0.94	0.84	3.43	1.21	0.41	0.63	0.53	1.39	0.22

Table 4.12

Classification Accuracy of LDA Ensembles by ASFSP

Experiment #	Haberman	Iris	Lenses	Liver	Ecoli	Pima	Tic-Tac-Toe	Glass	Breast Cancer
1	74.84	98.00	87.50	63.48	75.96	75.52	73.05	62.66	97.07
2	75.16	98.00	87.50	63.77	75.90	75.91	72.79	62.66	97.07
3	74.51	98.00	83.33	63.77	75.96	76.17	72.53	62.76	97.22
4	74.51	98.00	87.50	63.77	76.00	75.78	72.92	62.66	97.22
5	74.51	98.00	87.50	64.06	75.89	75.65	73.70	61.79	97.51
6	74.51	98.00	87.50	64.06	75.96	76.04	73.18	62.66	97.22
7	75.16	98.00	87.50	64.06	75.96	76.17	72.53	62.66	97.22
8	74.84	98.00	83.33	63.48	75.79	76.17	73.96	62.03	97.22
9	75.16	98.00	87.50	64.06	75.96	76.04	72.79	62.66	97.22
10	75.16	98.00	87.50	64.06	75.96	76.82	72.79	62.66	97.07
Average	**74.84**	**98.00**	**86.67**	**63.86**	**75.93**	**76.03**	**73.02**	**62.52**	**97.20**
Standard deviation	0.31	0.00	1.76	0.24	0.06	0.36	0.47	0.33	0.13

Based on experimental results presented above, there is a performance comparison of both techniques performed. The comparison between proposed technique and random paritioning to construct homogeneous LDA ensembles can be seen in Table 4.13 and in Figure 4.7.

Table 4.13

Comparison of RS and ASFSP in Constructing LDA Ensembles

Dataset	Classifier Ensemble Construction					
	Random Subspace			Ant System-based Feature Set Partitioning		
	Average of Accuracy (%)	Feature Subset	# of classifier	Average of Accuracy (%)	Feature Partition	# of classifier
Haberman	73.76	[3][2 3][2][1 3]	4	74.84	[1][2 3]	2
Iris	95.73	[1 3 4][4][1 4][3]	4	98.00	[1 2 3 4]	1
Lenses	80.42	[1 4][1][1 2 3 4][1 3 4]	4	86.67	[1 2 3 4]	1
Liver	62.44	[1 2 6][3 4 5 6][1 2 3 5 6][2 3]	4	63.86	[1 3 4 6][2][5]	3
Ecoli	73.28	[1 4 5 7][2 4 6][1 2 3 4][2 5 6]	4	75.93	[1 3 5][4 6][2 7]	3
Pima	74.94	[1 2 7][2 3 8][1 2 3 7][4 7 8]	4	76.03	[1 2 3 4 5 6 7 8]	1
Tic-Tac-Toe	65.63	[1 3 6 7][4 6 7 9][3 4 5 8][1 2 7 8 9]	4	73.02	[2 4 5 6 8][1][3][7][9]	5
Glass	59.21	[1 2 3 4 7 9][1 2 3 5 6 7][[7 9][3 5 9]	4	62.52	[2 3 5 7][4 8 9][1 6]	3
Breast Cancer	96.21	[4 5 7 8][2 3 4 6 7][6 7 8][3 4 5 6 7]	4	97.20	[2 4 8][7 9][3][1 5 6]	4

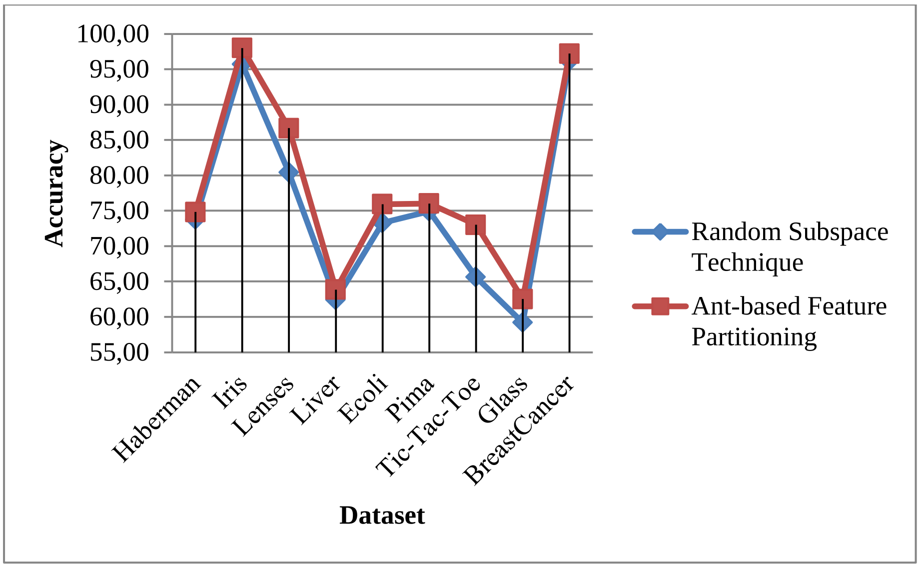

Figure 4.7 Comparison of RS and ASFSP in constructing LDA ensembles

Most of the datasets used on the experiment are successfully partitioned. It generates diversity in the ensemble, thus accuracy improvement is obtained on corresponding datasets. Especially on the tic-tac-toe dataset, a significant accuracy improvement is obtained when the number of partitions is 5 (five). Similarly, on the glass dataset, significant improvement of classification accuracy is obtained when the number of partitions is 3 (three). In general, homogeneous LDA ensemble which is constructed using ant system-based feature partioning gives better classification results. The right feature set partitioning technique can reduce the data dimensionality and also improve the classification accuracy with the use of optimal number of classifiers.

4.3.5 Comparison of Constructed Classifier Ensembles

The experimental results on applying proposed ant system-based feature set partitioning to each classifier are given in the previous section. In this section, the comparison of four constructed homogeneous classifier ensembles on whole datasets is given as well. It is intended to easily perform a comprehensive comparison. Table

4.14 shows the comparison of four constructed classifier ensembles on the whole dataset.

Table 4.14

Comparison of Constructed Classifier Ensembles

Dataset	Base Classifier of Homogeneous Ensembles	Average of Accuracy (%)	
		Random Subspace	Ant System-based Feature Set Partitioning
Haberman	NMC	70.33	70.39
	NBC	74.61	74.81
	k-NN	67.91	68.53
	LDA	73.76	74.35
Iris	NMC	92.07	94.47
	NBC	94.80	95.46
	k-NN	93.40	95.93
	LDA	95.73	98.00
Lenses	NMC	66.25	66.67
	NBC	62.50	62.50
	k-NN	62.50	79.17
	LDA	80.42	86.67
Liver	NMC	56.43	64.29
	NBC	60.12	63.51
	k-NN	60.06	64.16
	LDA	62.44	63.86
Ecoli	NMC	81.67	81.82
	NBC	75.25	75.53
	k-NN	81.19	81.19
	LDA	73.28	75.93
Pima	NMC	67.88	73.02
	NBC	75.70	75.44
	k-NN	70.59	71.01
	LDA	74.94	76.03
Tic-Tac-Toe	NMC	64.49	73.01
	NBC	68.34	72.61
	k-NN	75.70	75.73
	LDA	65.63	73.02
Glass	NMC	44.44	53.22
	NBC	73.21	73.25
	k-NN	72.71	72.90
	LDA	59.21	62.52
Breast Cancer	NMC	96.50	97.23
	NBC	96.14	97.63
	k-NN	97.23	97.60
	LDA	96.21	97.20

Furthermore, for easy comparison between random subspace and ant system-based feature set partitioning on constructing four classifier ensembles, the average of combined classifier accuracies are again presented in Figure 4.8.

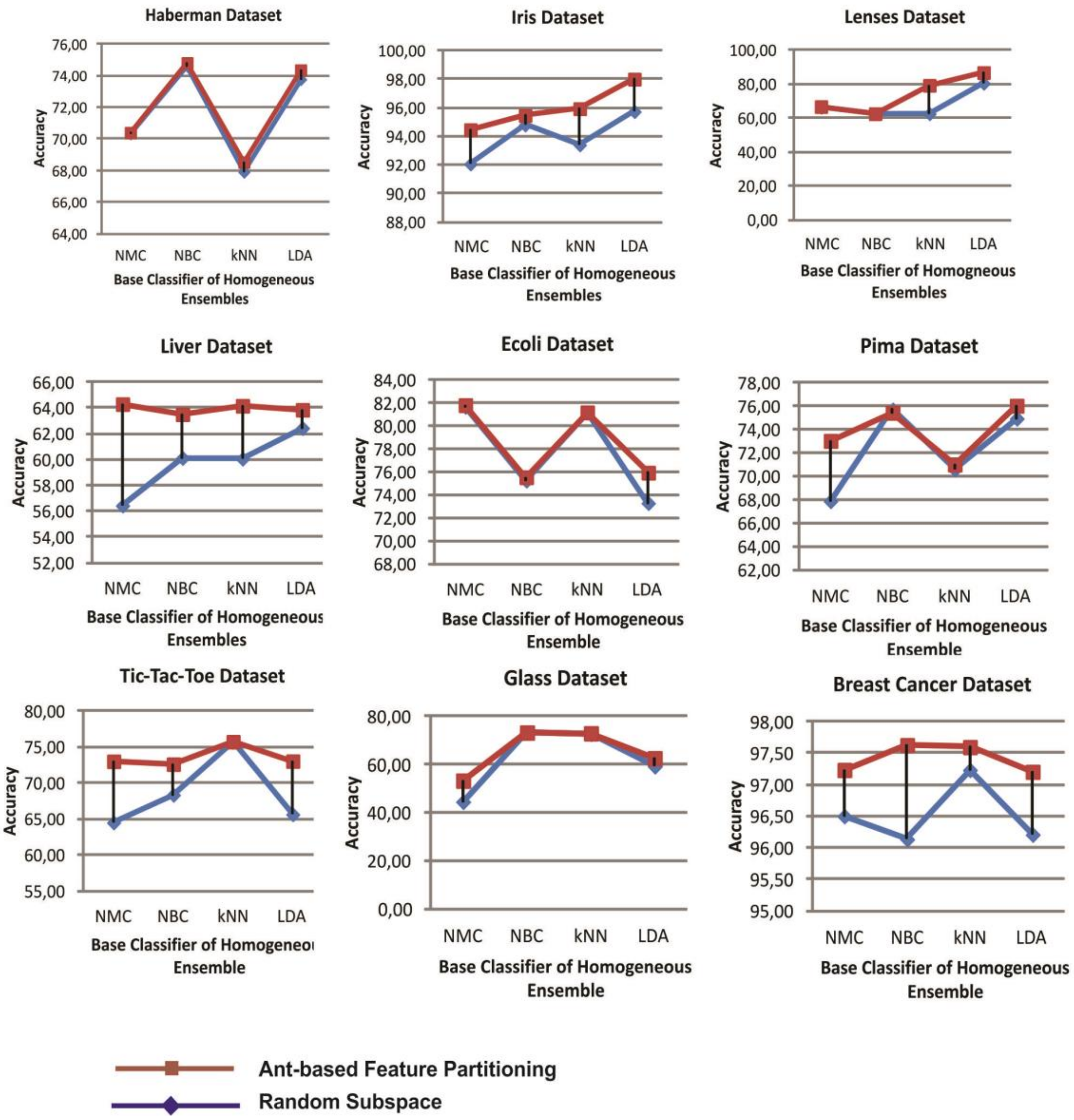

Figure 4.8 Comparison of four homogeneous classifier ensembles for whole datasets

Based on the comparison results between constructed classifier ensembles using random subspace and ant system-based feature set partitioning, it can be seen that generally there is an improvement of classification accuracy on all datasets. Despite this improvement, accuracy varies from large to small increments. Only the *k*-NN performance is significantly improved on haberman and lenses datasets, in contrast, the *k*-NN performance tends to not increase on the tic-tac-toe dataset. The performance of NMC is significantly improved on the pima dataset. Performance enhancement occurs in all type of classifiers on liver and breast cancer datasets. On the other hand, performance improvement does not occur in all type of classifiers on ecoli and glass datasets.

4.4 Proposed Compactness Measure

A parameter is introduced in this study to measure compactness in a set of classifier. This measurement is called compactness measure which reflects the overall support of all the classifiers regardless of their diverse or similar situations. This parameter is introduced with the aim of answering diversity-accuracy dilemma.

Diversity in classifier ensemble is an important thing in classifier combination (Kuncheva & Whitaker, 2003). However there is no consensus on diversity, and hence no diversity measure has satisfy all researchers (Zhou, 2012). Furthermore, the “good” and “bad” diversity phenomenon occurs when the majority voting is used as combiner in the ensemble (Brown & Kuncheva, 2010). It is intuitive that increasing diversity among the member of ensemble should lead to better accuracy of the ensemble, but there is no formal proof of this dependency and theoritical results have not been proven yet (Wozniak, et al., 2014). The main problem in determining diversity is the diversity-accuracy dilemma. When diversity in classifier ensemble increases, the base classifier accuracy decreases and vice versa (Li & Gao, 2010). Therefore, the problem of using diversity measure in combining clissifier is also a focus of this study.

The compactness can be measured simultaneously in a set of classifier. Therefore, the non-pairwise approach is considered in this study. This proposed parameter value can be measured and tested through usage of several methods which are ant system-based feature set partitioning, majority voting combiner, homogeneous ensemble, the 10 fold cross validation method and regression analysis. In constructing the support formulation for compactness measurement, several notations and definitions are adopted from previous study of Kuncheva (2001) as follows: let $D = \{D_1, \ldots, D_L\}$ be a set of classification algorithm (pool, committee, mixture, team, ensemble). In addition, let $\Omega = \{\omega_1, \ldots, \omega_c\}$ be a set of class labels and $X = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$ be a training set where $x_i \in \Re^n$ be a vector with n features and $y_i \in \Omega$ is the class label of i-th data instance. The output of the classifiers D_j on samples x_i is $D_j(x_i)$ for $1 \leq i \leq N$ and $1 \leq j \leq L$ can be represented as N-dimensional vector $v = [D_1(x_1), D_2(x_2), \ldots, D_L(x_N)]^T$, the output of ensemble defined as follows: let the output of classifier $D_j(x_i)$ = 1 if D_j correctly recognise x_i and otherwise $D_j(x_i)$=0. This type of output is also called oracle output (Pasti & de Castro, 2007). The oracle output simply says if the classifier presented a correct classification or not (1 or 0). This study focused on the oracle output for two (2) reasons. (1) Most of the researches on diversity measures are based on output oracle. Ten kinds of diversity measures have been reported by Kuncheva and

Whitaker (2003) based on the oracle output. (2) In oracle output, no prior knowledge of the data is needed and the type of output can be produced by any classifier. The output is on whether the samples are classified correctly or not. Therefore, the oracle output is a common model for analysing classifier ensemble, and the conclusions which are obtained from this model can be easily generalised to the classifier ensemble construction method.

The support ($\mathcal{s}$) is defined as the proportion of input pattern correctly classified by all classifiers and the number of observations. This is also the ratio between the numbers of observations on which all classifiers are correct to the total number of observations. Therefore, the value of this parameter is expressed in a percentage, thus the range of values of this parameter value is from 0 to 100. A small value indicates a less compactness of classifier ensemble in supporting the right decision. High value indicates a more compactness of classifier ensemble in supporting the right decision. In this way, the compactness in a set of classifiers is measured by support formulation and can be written as follows:

$$\mathcal{s} = \frac{1}{N}\sum_{i=1}^{N} t_i \tag{4.1}$$

where

$$t_{i=}\begin{cases} 1: \text{ if } D_1(x_i) = D_2(x_i) = \cdots = D_L(x_i) = y_i \\ 0: \text{otherwise} \end{cases}$$

where

L: total number of classifiers

N: total number of samples

$D_j(x_i)$: the output of classifier D_j on sample x_i

y_i: the class label of sample x_i

The idea of this formulaion was adopted and adapted from the "support" formulation of market basket analysis (Han et al., 2011). Market basket analysis is a method for recognition of dependencies in data. "Support" is the ratio of number of times two or more items occur together to the total number of transactions. The proportion of items occur together can be analogized as compactness classifier to support the same decision.

There are several diversity measures that have been proposed in the literature (Kuncheva & Whitaker, 2003). However, no clear relationship has been found so far between each diversity measure and all other diversity measures as cited by Canuto et al. (2007). The correlation between each of the methods of combination and each existing diversity measures are not in the scope of this study, but the correlation between of proposed compactness measure and ensemble accuracy is focus in this study.

Several diversity measures have been proposed but none can be used to strongly correlate with ensemble accuracy (Shipp & Kuncheva, 2002; Banfield et al., 2005; Lofstrom et al., 2007; Musehane et al., (2008); Bi, 2012). Shipp and Kuncheva (2002) studied the relationships between different methods of classifier combination methods and 10 diversity measures. Two (2) datasets from UCI repository of machine learning namely breast cancer (Wisconsin) and pima indians diabetes were used in their experiments. Summary of the datasets is provided in Table 4.15.

Table 4.15

Summary of Datasets Used in the Experiments (Shipp & Kuncheva, 2002)

Datasets	Instances	Classes	Feature partition	Training/ Testing
Breast Cancer (Wisconsin)	569	2	4,4,2 4,3,3	Hold out
Pima Indians Diabetes	768	2	3,3,2	10-fold cross validation method

The results of their experiments showed low correlation between the combination methods and diversity measure as shown in the Table 4.16 where *p* is the range of significant value. This result is discouraging because diversity measure should be able to give a prediction of the performance of the classifier combination.

Table 4.16

Correlation Diversity Measure and Ensemble Accuracy with p Value.

Diversity Measure	p value
Q	0.7-0.9
P	0.7-0.9
DF	0.7-0.9
K	0.7-0.9
0	0.7-0.9
D	0.7-0.9
kw	0.7-0.9
Ent	0.7-0.9
GD	0.7-0.9
CFD	0.7-0.9

The correlations between diversity measure and combination methods are low and not consistent. The ambiguous relationship between diversity and accuracy is difficult to optimize diversity measure.

The relationships between four (4) different methods of classifier combination methods i.e. bagging, random forest, random trees, and random forest and three (3) diversity measure i.e. PCDM, *Q* and *K* have been evaluated by Banfield et al., (2005). Several datasets from UCI machine learning repository were used for evaluation as presented in Table 4. 17.

Table 4.17

Summary of Datasets Used in the Experiments (Banfield et al., 2005)

Datasets	Instances	Classes	Features
Iris	150	3	4
glass	214	6	9
breast cancer (Wisconsin)	683	2	9

The results of their experiments are as shown in Table 4.18. The determination coefficient (R^2), is a measure to determine how well the linear regression fits to the measured data. Table 4.18 shows the R^2 value of the correlation test.

Table 4.18

The R^2 Values for Each of the Ensemble Construction Methods and Diversity Measures (Banfield et al., 2005).

Ensemble methods	R^2		
	PCDM	**Q**	***K***
Bagging	0.6353	0.5046	0.1775
Random forest	0.6905	0.6105	0.2602
Random tree	0.5765	0.5614	0.2106
Random subspace	0.6096	0.5071	0.3512

The results indicated a weak correlation between them. The concept of diversity is interesting because its effects can easily be seen. However, its quantification and manipulation are not quite well defined.

Experiments to evaluate the diversity measure have also been carried out by Lofstrom et al. (2007). Neural network was used as base classifier to build neural network classifier ensembles. The homogeneous neural network ensembles were trained by varying the network architecture to induce diversity. Eight (8) datasets from the UCI machine learning repository were used in their experiments (refer Table 4.19).

Table 4.19

Summary of Datasets Used in the Experiments (Lofstrom et al., 2007).

Datasets	Instances	Classes	Continue	Category
Cleve	303	2	6	7
Cmc	1473	3	5	4
Crx	690	2	6	9
Ecoli	336	8	5	3
Hypo	3163	2	7	18
Pima	768	2	8	0
Sat	6435	7	36	0
Vehicle	846	4	18	0

The goal of the experiments is to evaluate the ten (10) diversity measures that have been summarized by Kucheva and Whitaker (2003). According to them, the difficulty (Θ) and double faults (DF) are good to measure diversity. However, the use of these diversity measures was not successful in building a better classifier combination.

The study on relationship between diversity and accuracy of the ensemble has also been carried out by Musehane et al., (2008). Neural network has been been used as the base classifier. The parameters of the neural network within the committee were varied to induce diversity and the proposed parameter for the measure of diversity are Shannon measure (Shannon, 1948) and Simpson measure (Simpson, 1949). The demographic dataset used is derived from antenatal clinic in South Africa and it was collected by the Department of Health in 2001 as shown in Table 4.20.

Table 4.20

The Demographic Dataset Used in the Experiments (Musehane et al., 2008)

Variable	Type	Range
Age	Integer	13-50
Education	Integer	0-13
Parity	Integer	0-9
Gravidity	Integer	1-12
Province	Integer	1-9
Age of father	Integer	14-60
HIV status	Binary	0-1

The results showed that an increased in the diversity of the ensemble also resulted in the increased of the ensemble accuracy. However the used of this method is computational expensive because of the used of GA.

A study by Bi (2012) focused on the impact of diversity on the accuracy of the ensemble. Twelve (12) datasets from the UCI repository machine have been used. The general description of datasets can be seen in Table 4.21.

Table 4.21

The General Description of Datasets Used in the Experiments (Bi, 2012).

Datasets	**Instance**	**Classes**	**Features**
Anneal	798	6	38
Audiology	200	23	69
Balance	625	3	4
Car	1728	4	6
Glass	214	7	9
Autor	205	6	25
Iris	150	3	4
Letter	20000	26	16
Segment	1500	7	19
Soybean	683	19	35
Wine	178	3	13
Zoo	101	7	17

Thirteen (13) classifiers have been used as base classifier and experiments were performed by using ten-fold cross-validation. After calculating the correlation between diversity and ensemble accuracy, empirical results indicate that the increased diversity makes the ensemble accuracy decreases and vice versa. The increased in diversity is not consistent with the increased accuracy of the ensemble. Therefore, it reinforces that there is a conflict between them. Table 4.22 shows the summary of correlation between accuracy and diversity (↑: positive correlation; **🠉**: strongly positive correlation; ↓: negative correlation; **🠋**: strongly negative correlation; ↕: neutral correlation).

Table 4.22

Summary Correlation Between Diversity and Ensemble Accuracy (Bi, 2012)

Ensemble Method	Ensemble accuracy				Improved accuracy			
	kw	**qs**	**dis**	**K**	**kw**	**qs**	**dis**	**k**
Dempster's rule	↑	↑	↑	↑	**↓**	↓	**↓**	**↓**
Smets' rule	↑	↑	↑	↑	**↓**	↑	**↓**	**↓**
Proportion rule	**↑**	↑	**↑**	**↑**	↕	↕	↕	↕
Yager's rule	**↑**	↑	**↑**	**↑**	**↑**	↑	**↑**	**↑**

Based on the demonstrated empirical results, the effectiveness of the scheme needs to be optimized. In general, there is a need to design better mechanisms that will be used to successfully construct a classifier ensemble without sacrificing accuracy and diversity.

4.5 Experimental Results on Compactness Measurement

This section presents the experiment conducted in calculating compactness value. Compactness value was calculated based on four classifier ensembles which have been constructed in the previous experiments. The four classifier ensembles are homogeneous NMC ensembles, homogeneous NBC ensembles, homogeneous *k*-NN ensembles and homogeneous LDA ensembles. Nine (9) datasets from UCI repository were used in the experiment. Ten (10) experiments were performed only on datasets that form classifier ensembles. During the experiment, the compactness value in classifier ensembles was calculated using the support formulation. The ensemble accuracy was calculated by compactness value. The most commonly used combiner, majority voting, is used to combine classifier outputs. 10-fold cross-validation approach is used to validate performance of ensemble.

4.5.1 Calculating Compactness in NMC Ensembles

Experiments were conducted to calculate the compactness value versus NMC ensembles accuracy on created partition. Table 4.23 – Table 4.29 and Figure 4.9 – Figure 4.15 show compactness value versus NMC ensembles accuracy on related dataset and also the formed partition based on ACO. Eeach partition results will provide compactness value and the ensemble accuracy regardless of how the partitioning performed. Thus the relationship between proposed parameter and ensemble accuracy can be tested empirically.

Table 4.23

Compactness vs NMC Ensembles Accuracy on Haberman Dataset

Experiment #	**Partition Based on ACO**	**Compactness (s)**	**(%)**	**Ensemble Accuracy (%)**
1	[1][2,3]	29.27		69.93
2	[1][2,3]	29.18		71.57
3	[1][2,3]	31.98		71.24
4	[1,2][3]	32.16		73.86
5	[1][2,3]	30.38		67.97
6	[1,2][3]	31.42		68.63
7	[1][2,3]	30.57		69.93
8	[1][2,3]	31.88		70.92
9	[1,2][3]	33.33		69.28
10	[1][2,3]	31.22		70.59
	Average	**31.14**		**70.39**

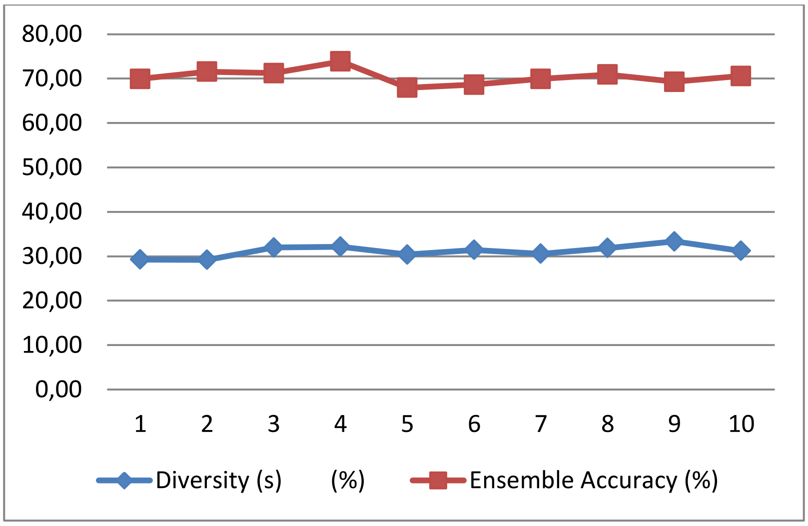

Figure 4.9 Compactness vs NMC ensembles accuracy on haberman dataset

Table 4.24

Compactness vs NMC Ensembles Accuracy on Iris Dataset

Experiment #	Partition Based on ACO	Compactness (s)	(%)	Ensemble Accuracy (%)
1	[23][4][1]	71.33		94.67
2	[4][3][12]	77.33		94.67
3	[24][1][3]	69.78		94.67
4	[4][23][1]	71.11		94.67
5	[3][12][4]	78.00		95.33
6	[4][3][12]	77.33		94.67
7	[4][23][1]	71.33		94.67
8	[4][23][1]	71.11		94.67
9	[3][124]	79.67		92.00
10	[23][1][4]	70.89		94.67
	Average	**73.79**		**94.47**

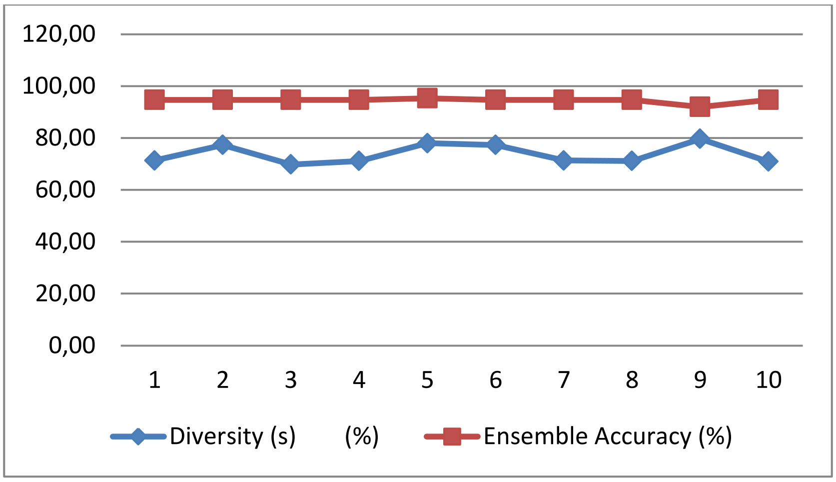

Figure 4.10 Compactness vs NMC ensembles accuracy on iris dataset

Table 4.25

Compactness vs NMC Ensembles Accuracy on Liver Dataset

Experiment #	Partition Based on ACO	Compactness (s)	(%)	Ensemble Accuracy (%)
1	[1,2,4,6][3][5]	8.86		64.06
2	[1,2,4,6] [3] [5]	9.38		62.61
3	[2,4,6] [1,5] [3]	7.92		64.35
4	[1,2,4] [3,6] [5]	9.30		65.51
5	[1,2,4,6][3][5]	9.26		64.64
6	[1,2,4,6][3][5]	8.86		64.35
7	[2,4,6][1,5][3]	8.33		63.77
8	[2,4,6] [1,5] [3]	7.92		64.35
9	[1,2,4,6][3][5]	9.26		64.64
10	[1,2,4,6][3][5]	9.26		64.64
	Average	**8.84**		**64.29**

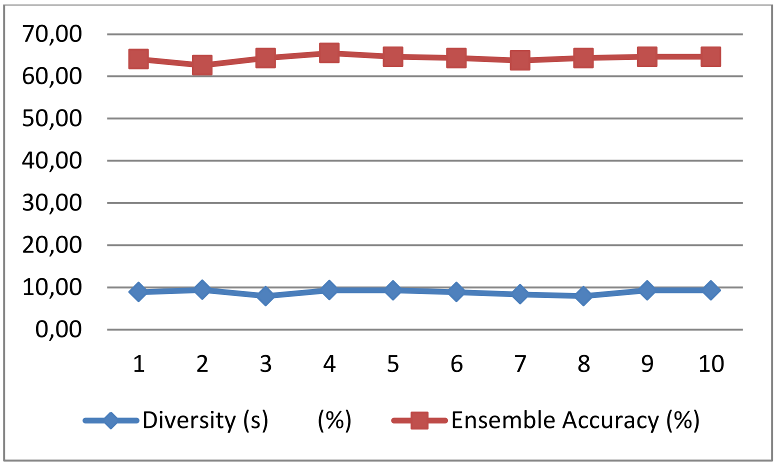

Figure 4.11 Compactness vs NMC ensembles accuracy on liver dataset

Table 4.26

Compactness vs NMC Ensembles Accuracy on Pima Dataset

Experiment #	**Partition Based on ACO**				**Compactness (s)**	**(%)**	**Ensemble accuracy (%)**
1	[3,4,5,7]	[1,6]	[8]	[2]	24.71		73.05
2	[3,7,8]	[5]	[2,4]	[1,6]	24.45		72.79
3	[3,8]	[1,6]	[5,7]	[2,4]	24.77		72.53
4	[3,7,8]	[5]	[2,4]	[1,6]	24.45		72.92
5	[3,4,5,7]	[1,6]	[8]	[2]	24.71		73.70
6	[3,4,5,7]	[1,6]	[8]	[2]	24.71		73.18
7	[3,8]	[1,6]	[5,7]	[2,4]	24.77		72.53
8	[3,4,5,7]	[1,6]	[8]	[2]	24.71		73.96
9	[3,7,8]	[5]	[2,4]	[1,6]	24.45		72.79
10	[3,7,8]	[5]	[2,4]	[1,6]	24.45		72.79
				Average	**24.62**		**73.02**

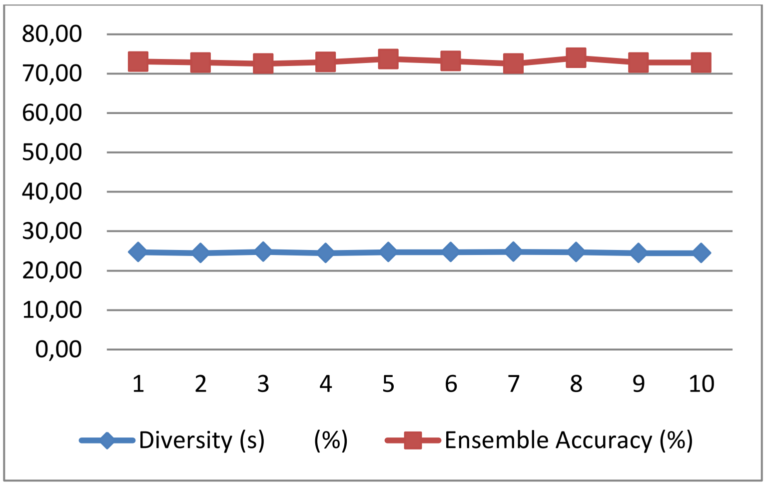

Figure 4.12 Compactness vs NMC ensembles accuracy on pima dataset

Table 4.27

Compactness vs NMC Ensembles Accuracy on Tic-Tac-Toe Dataset

Experiment #	**Partition Based on ACO**					**Compactness (s) (%)**	**Ensemble Accuracy (%)**
1	[2,5,6,8]	[1]	[4,9]	[3]	[7]	24.32	73.17
2	[1,4,6,7]	[2,5,8]	[9]	[3]		24.54	72.76
3	[7]	[4,5,8]	[2,6,9]	[3]	[1]	24.78	72.65
4	[4,7]	[1,2]	[5,6,8]	[9]	[3]	24.71	74.01
5	[2,5,6,8]	[1,4,7]	[3]	[9]		24.45	72.86
6	[2,3,6,8,9]	[4,5]	[1]	][7]		24.51	72.86
7	[3,6,8,9]	[2,4,5]	[1]	[7]		23.98	72.86
8	[2,4,5,8]	[7]	[3,6,9]	[1]		24.71	72.65
9	[2,4,5,8]	[6,9]	[3]	[1]	[7]	24.54	73.28
10	[1,2,3,4,6,8]	[7]	[5]	[9]		24.45	72.96
					Average	**24.50**	**73.01**

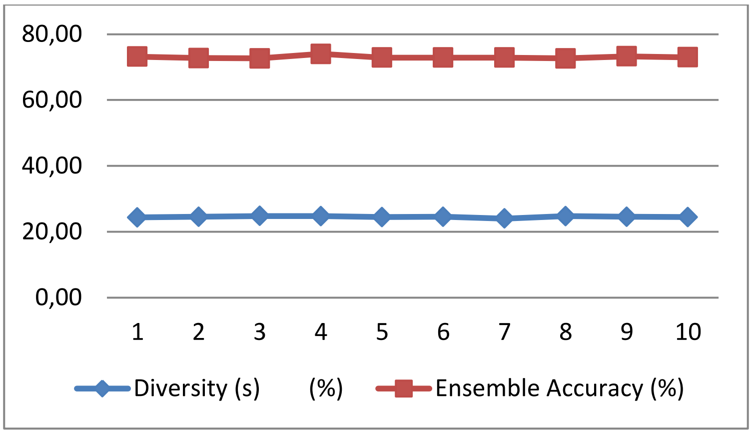

Figure 4.13 Compactness vs NMC ensembles accuracy on tic-tac-toe dataset

Table 4.28

Compactness vs NMC Ensembles Accuracy on Glass Dataset

Experiment #	**Partition Based on ACO**				**Compactness (s)**	**(%)**	**Ensemble Accuracy (%)**
1	[5,7]	[2,3,9]	[6,8]	[1,4]	2.76		52.80
2	[2,3,5,7]	[1,4,8,9]	[6]		2.53		53.27
3	[1,2,3]	[6,8,9]	[5,7]	[4]	2.06		55.14
4	[5,7]	[2,3,9]	[6,8]	[1,4]	2.76		52.80
5	[1,4,8]	[5,7,9]	[2,3]	[6]	2.79		52.34
6	[1,2,3]	[6,8,9]	[5,7]	[6]	2.06		52.34
7	[2,3,5,7]	[4,8,9]	[1,6]		2.18		52.80
8	[2,3,5,7]	[1,4,8,9]	[6]		2.53		53.27
9	[4,8]	[2,3,5,7,9]	[1,6]		2.67		54.21
10	[2,3,5,7]	[1,4,8,9]	[6]		2.53		53.27
				Average	**2.49**		**53.22**

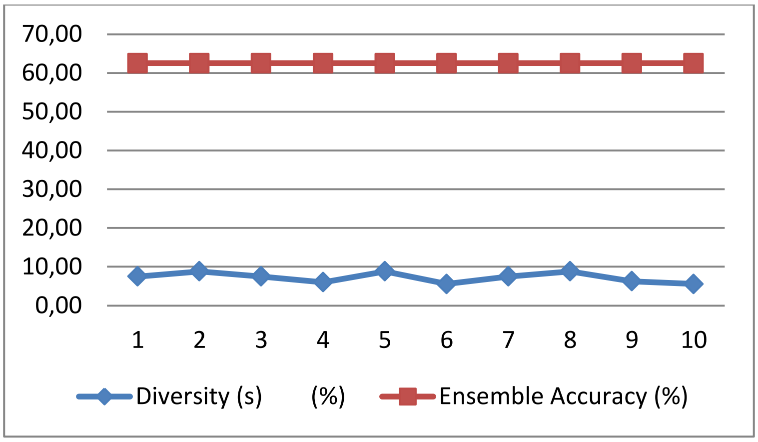

Figure 4.14 Compactness vs NMC ensembles accuracy on glass dataset

Table 4.29

Compactness vs NMC Ensemble Accuracy on Breast Cancer Dataset

Experiment #	Partition Based on ACO		Compactness (s)	(%)	Ensemble Accuracy (%)
1	[2,3,4,5,8,9]	[1,6,7]	92.40		97.22
2	[2,4,5,6,7,8,9]	[1,3]	92.39		97.22
3	[1,2,3,4,5,7,9]	[6,8]	92.30		97.22
4	[1,2,3,4,5,7,9]	[6,8]	92.30		97.22
5	[1,3,5,6,7,9]	[2,4,8]	92.35		97.22
6	[1,5,6,7,8]	[2,3,4,9]	92.80		97.36
7	[2,4,6,7,8]	[1,3,5,9]	92.54		97.22
8	[1,3,5,6,7,9]	[2,4,8]	92.29		97.22
9	[2,4,6,7,8]	[1,3,5,9]	92.18		97.22
10	[2,4,5,6,7,8]	[1,3,9]	92.18		97.22
		Average	**92.37**		**97.23**

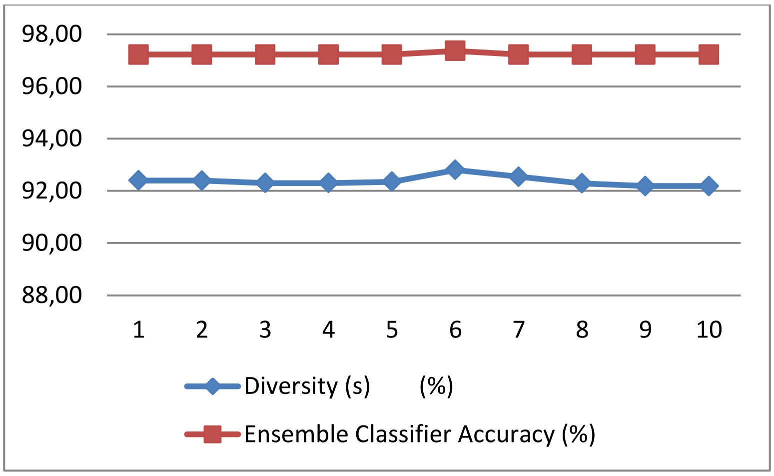

Figure 4.15 Compactness vs NMC ensembles accuracy on breast cancer dataset

Based on the experimental results, it can be seen that the compactness in the NMC ensemble is directly proportional to the accuracy of the ensemble. The larger value of compactness will provide higher accuracy and vice versa.

4.5.2 Calculating Compactness in NBC ensembles

Experiments were conducted to calculate the compactness value versus NBC ensembles accuracy on created partition. Table 4.30 – Table 4.32 and Figure 4.16 – Figure 4.18 show the partition, compactness value and NBC ensembles accuracy on related datasets.

Table 4.30

Compactness vs NBC Ensembles Accuracy on Lenses Dataset

Experiment #	**Partition Based on ACO**			**Compactness (s)**	**(%)**	**Ensemble Accuracy (%)**
1	[3]	[1,2,4]		7.50		62.50
2	[2]	[1,3,4]		8.83		62.50
3	[3]	[1,2,4]		7.50		62.50
4	[1,2]	[3]	[4]	6.00		62.50
5	[2]	[1,3,4]		8.83		62.50
6	[4]	[3]	[1,2]	5.56		62.50
7	[3]	[1,2,4]		7.50		62.50
8	[2]	[1,3,4]		8.83		62.50
9	[1,2,4]	[3]		6.25		62.50
10	[1,2]	[4]	[3]	5.56		62.50
			Average	**7.24**		**62.50**

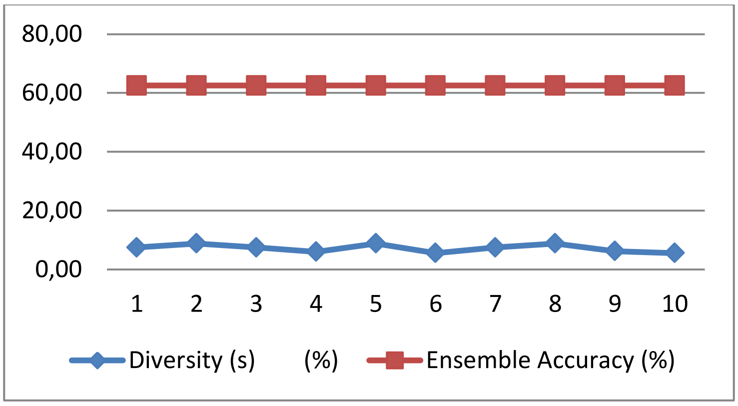

Figure 4.16 Compactness vs NBC ensembles accuracy on lenses dataset

Table 4.31

Compactness vs NBC Ensembles Accuracy on Liver Dataset

Experiment #	Partition Based on ACO				Compactness (s)	(%)	Ensemble Accuracy (%)
1	[4,5,6]	[1,2]	[3]		10.41		64.93
2	[1,2,3,4]	[6]	[5]		9.37		63.77
3	[1,2,3,4]	[6]	[5]		9.37		63.77
4	[3,5,6]	[2]	[1,4]		10.29		62.32
5	[1,2,3,4]	[6]	[5]		9.37		63.77
6	[3,5,6]	[4]	[1]	[2]	7.60		63.48
7	[2,5]	[3,4]	[1]	[6]	7.65		63.19
8	[3,5,6]	[1,4]	[2]		9.88		62.61
9	[4,5,6]	[3]	[1,2]		10.15		62.90
10	[4,5,6]	[1,2]	[3]		10.25		64.35
				Average	**9.43**		**63.51**

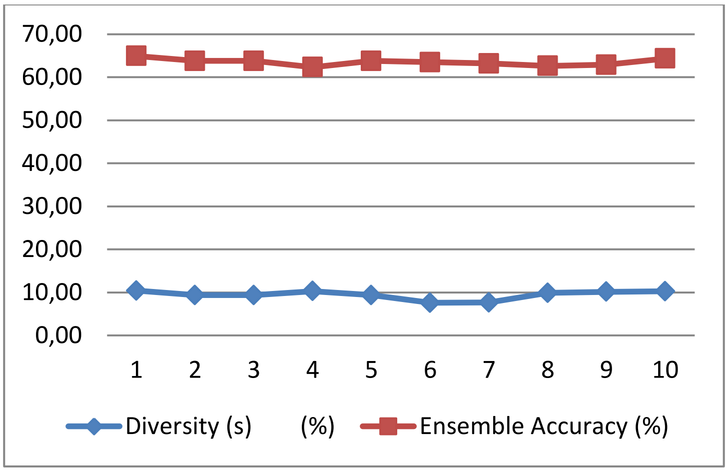

Figure 4.17 Compactness vs NBC ensembles accuracy on liver dataset

Table 4.32

Compactness vs NBC Ensembles Accuracy on Breast Cancer Dataset

Experiment #	**Partition Based on ACO**				**Compactness (s)**	**(%)**	**Ensemble Accuracy (%)**
1	[2,4,6,9]	[1,5,8]	[3,7]		87.21		97.66
2	[1,3,5,9]	[2,6,7]	[4,8]		85.11		97.51
3	[2,6,7,9]	[1,4,5,8]	[3]		86.19		97.51
4	[6]	[4,5,8,9]	[1,2,7]	[3]	78.62		97.66
5	[6]	[4,5,8,9]	[1,2,7]	[3]	78.62		97.66
6	[1,4,5,8]	[2,9]	[3,7]	[6]	80.39		97.51
7	[4,5,8,9]	[1,2,7]	[3]	[6]	78.69		97.80
8	[6]	[4,5,8,9]	[1,2,7]	[3]	78.62		97.51
9	[4,5,8,9]	[1,2,7]	[3]	[6]	78.50		97.80
10	[6]	[4,5,8,9]	[1,2,7]	[3]	78.62		97.66
				Average	**81.06**		**97.63**

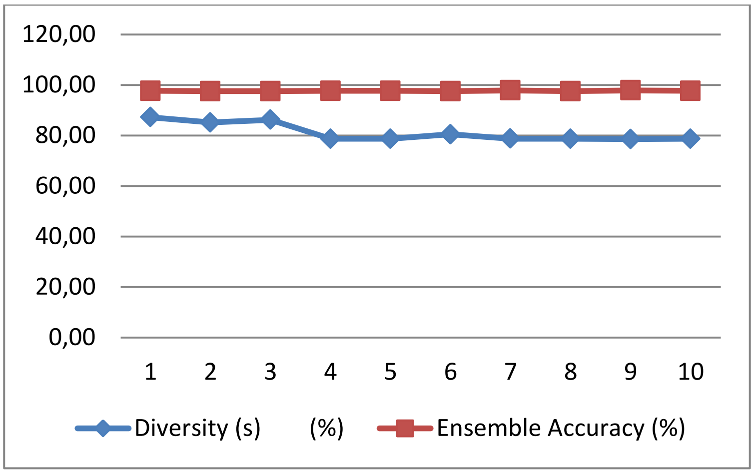

Figure 4.18 Compactness vs NBC ensembles accuracy on breast cancer dataset

Based on the experimental results, it can be seen that the ensemble accuracy follow compactness in NBC ensembles. In other words, the greater value of compactness in the NBC ensembles will provide higher accuracy.

4.5.3 Calculating Compactness in *k*-NN ensembles

Experiments were conducted to calculate the compactness value versus *k*-NN ensembles accuracy on created partition. Table 4.33 – Table 4.36 and Figure 4.19 – Figure 4.22 shows the partition, compactness value and *k*-NN ensemble accuracy on related datasets.

Table 4.33

Compactness vs k-NN Ensembles Accuracy on Haberman Dataset

Experiment #	Partition Based on ACO	Compactness (s)	(%)	Ensemble Accuracy (%)
1	[1,3][2]	33.34		67,65
2	[1,3][2]	34.86		68,95
3	[1][2,3]	36.38		67,97
4	[1,3][2]	32.85		68,30
5	[1][2,3]	35.32		69,61
6	[13][2]	31.88		68,63
7	[1,3][2]	33.96		67,65
8	[1,3][2]	29.92		69,93
9	[13][2]	32.97		67,97
10	[1,3][2]	29.01		68,63
	Average	**33.05**		**68.53**

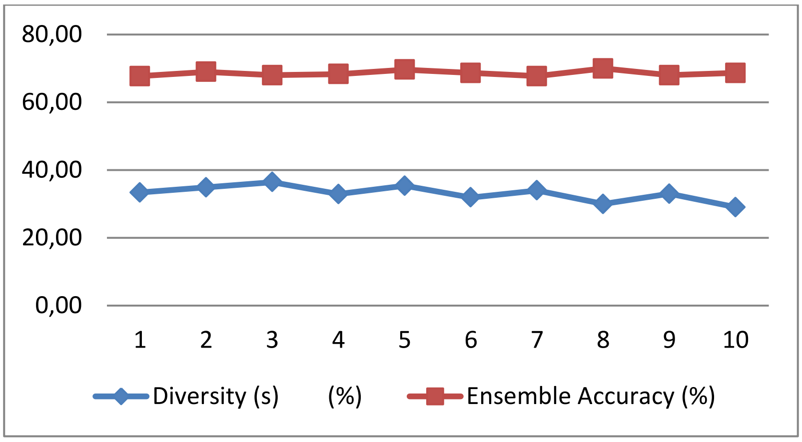

Figure 4.19 Compactness vs *k*-NN ensembles accuracy on haberman dataset

Table 4.34

Compactness vs k-NN Ensembles Accuracy on Liver Dataset

Experiment #	Partition Based on ACO			Compactness (s)	(%)	Ensemble Accuracy (%)
1	[1,4,6]	[3,5]	[2]	15.19		65.80
2	[1,2,3,4,5,6]			15.11		62.61
3	[1,4,6]	[3,5]	[2]	15.25		65.51
4	[1,3]	[4,5,6]	[2]	15.67		64.06
5	[1,2,3,4,5,6]			15.11		62.61
6	[1,2,6]	[3,5]	[4]	15.17		62.61
7	[1,2,3,4,5,6]			15.67		64.06
8	[1,4,6]	[3,5]	[2]	15.25		65.51
9	[3,5]	[1,4,6]	[2]	14.87		66.25
10	[1,3]	[2,4]	[5,6]	15.11		62.61
			Average	15.24		64.16

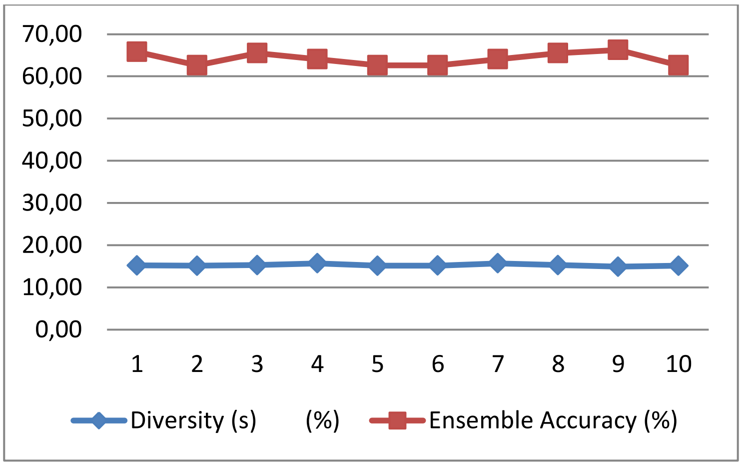

Figure 4.20 Compactness vs *k*-NN ensembles accuracy on liver dataset

Table 4.35

Compactness vs k-NN Ensembles Accuracy on Pima Dataset

Experiment #	Partition Based on ACO				Compactness (s) (%)	Ensemble Accuracy (%)
1	[6,7,8]	[1,3,4]	[5]	[2]	28.77	71.48
2	[1,3,4]	[2,6]	[5,7,8]		31.26	71.74
3	[1,3,4,7]	[5,6,8]	[2]		31.77	70.18
4	[5,6,7,8]	[1,3,4]	[2]		31.78	70.44
5	[6,7,8]	[1,3,4]	[5]	[2]	28.77	71.48
6	[1,3,4,7]	[5,6,8]	[2]		31.77	70.18
7	[1,3,4,5]	[6,7,8]	[2]		32.94	71.22
8	[4,5,7]	[1,2,3]	[6,8]		32.85	70.44
9	[1,6,7]	[2,3]	[5,8]	[4]	29.40	72.14
10	[1,4,6,7]	[2,3,5,8]			32.00	70.83
				Average	**31.13**	**71.01**

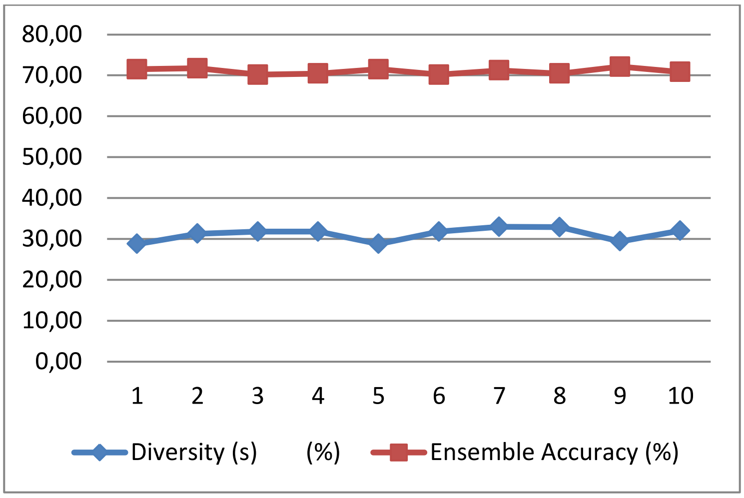

Figure 4.21 Compactness vs *k*-NN ensembles accuracy on pima dataset

Table 4.36

Compactness vs k-NN Ensembles Accuracy on Breast Cancer Dataset

Experiment #	Partition Based on ACO				Compactness (s) (%)	Ensemble Accuracy (%)
1	[1,4,7,9]	[2,3,5,8]	[6]		81.96	97.95
2	[2,3,5,9]	[1,4,7]	[6]	[8]	76.14	97.51
3	[1,2,4,7,9]	[3,5]	[6]	[8]	75.69	97.80
4	[3,4,8,9]	[2,5]	[1,7]	[6]	78.22	97.51
5	[3,5,9]	[1,2,8]	[4,6]	[7]	78.77	97.51
6	[2,3,9]	[1,4,5,7]	[6]	[8]	76.32	97.51
7	[1,3,5,9]	[2,7,8]	[4]	[6]	76.53	97.36
8	[1,3,4,6]	[2,5,9]	[7,8]		81.80	97.36
9	[1,2,4,7,9]	[3,5]	[6]	[8]	75.69	97.80
10	[1,3,4,5]	[2,7,8,9]	[6]		82.19	97.66
				Average	**78.33**	**97.60**

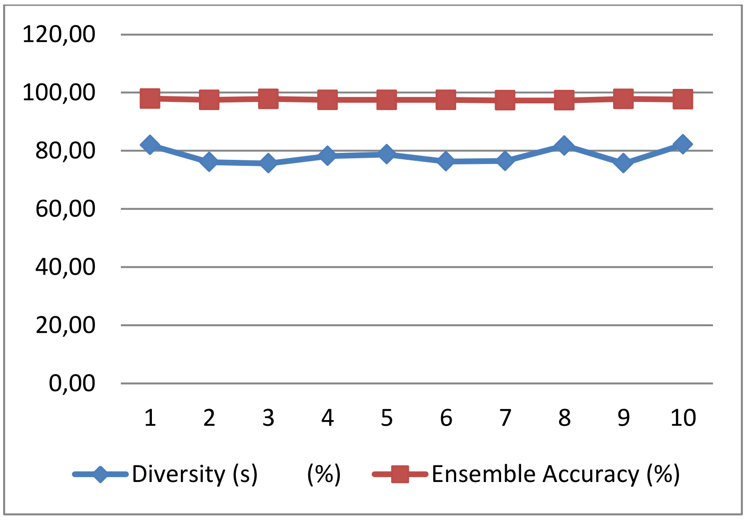

Figure 4.22 Compactness vs *k*-NN ensembles accuracy on breast cancer dataset

Based on the experimental results it can be seen that the compactness in *k*-NN ensembles is directly proportional to the accuracy of the ensemble. The larger value of compactness will provide high accuracy and vice versa.

4.5.4 Calculating Compactness in LDA ensembles

Experiments were conducted to calculate the compactness value versus LDA ensembles accuracy on created partition. Table 4.37 – Table 4.42 and Figure 4.23 – Figure 4.28 show the partition, compactness value and LDA ensembles accuracy on related datasets.

Table 4.37

Compactness vs LDA Ensembles Accuracy on Haberman Dataset

Experiment #	Partition Based on ACO	Compactness (s) (%)	Ensemble Accuracy (%)
1	[1][2,3]	38.21	74,84
2	[1][2,3]	38.36	74,51
3	[1][2,3]	38.08	74,51
4	[1,2][3]	37.93	74,18
5	[1][2,3]	38.44	74,51
6	[1][2,3]	38.94	74,51
7	[1][2,3]	38.25	73,53
8	[1,2][3]	38.20	74,18
9	[1][2,3]	38.56	74,51
10	[1][2,3]	38.35	74,18
	Average	**38.33**	**74.35**

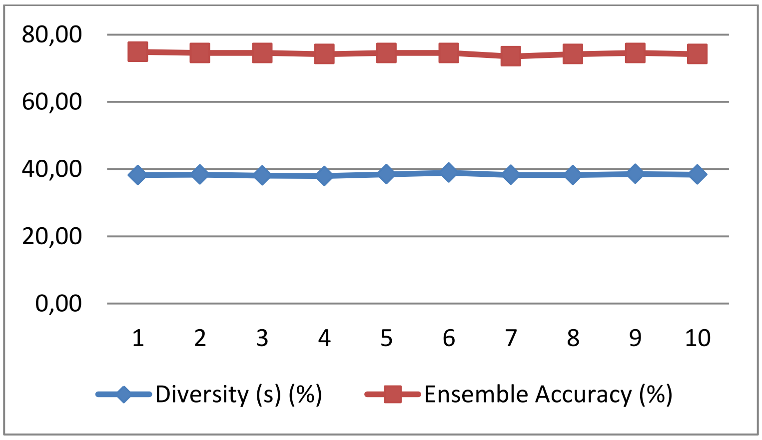

Figure 4.23 Compactness vs LDA ensembles accuracy on haberman dataset

Table 4.38

Compactness vs LDA Ensembles Accuracy on Liver Dataset

Experiment #	Partition Based on ACO			Compactness (s) (%)	Ensemble Accuracy (%)
1	[1]	[3,4,6]	[2,5]	14.59	63.48
2	[2,3,4,6]	[1]	[5]	12.37	63.77
3	[1,2,6]	[3,4]	[5]	13.81	63.77
4	[1,3,4,6]	[2]	[5]	14.28	63.77
5	[2,5]	[3,4,6]	[1]	14.50	64.06
6	[1,3,4]	[2,6]	[5]	12.95	64.06
7	[2,3,4]	[1,6]	[5]	12.64	64.06
8	[1,3,4,6]	[2]	[5]	13.63	63.48
9	[5,6]	[2]	[1,3,4]	12.67	64.06
10	[2,5,6]	[3,4]	[1]	14.66	64.06
			Average	**13.61**	**63.86**

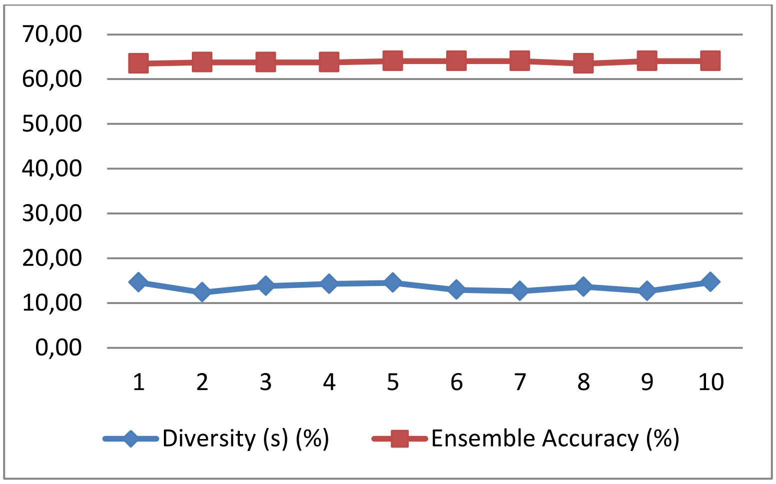

Figure 4.24 Compactness vs LDA ensembles accuracy on liver dataset

Table 4.39

Compactness vs LDA Ensembles Accuracy on Ecoli Dataset

Experiment #	**Partition Based on ACO**			**Compactness (s) (%)**	**Ensemble Accuracy (%)**
1	[1 3 5]	[4 6]	[7]	25.31	75.96
2	[1 2]	[3 7]	[4 5 6]	25.00	75.90
3	[1 3 5]	[4 6]	[7]	25.17	75.96
4	[1 2 6 7]	[5]	[3 4]	25.05	76.00
5	[1 3 5]	[4 6]	[7]	25.21	75.89
6	[1 2 6 7]	[5]	[3 4]	25.11	75.96
7	[1 2]	[3 7]	[4 5 6]	25.17	75.96
8	[1 3 5]	[4 6]	[7]	25.19	75.79
9	[1 2]	[3 7]	[4 5 6]	25.14	75.96
10	[1 3 5]	[4 6]	[7]	25.12	75.96
			Average	**25.15**	**75.93**

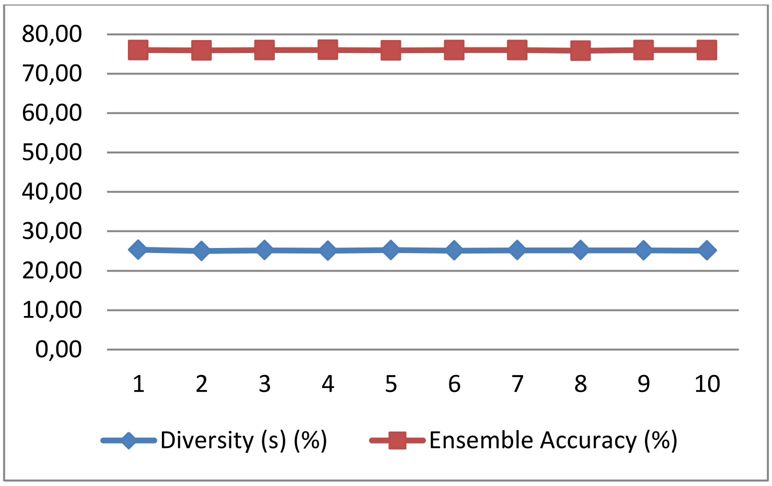

Figure 4.25 Compactness vs LDA ensembles accuracy on ecoli dataset

Table 4.40

Compactness vs LDA Ensembles Accuracy on Tic-Tac-Toe Dataset

Experiment #	**Partition Based on ACO**					**Compactness (s) (%)**	**Ensemble Accuracy (%)**
1	[2,4,5,6]	7	[3,8]	9	1	25.01	73.05
2	[6,7]	[2,8,9]	1	3	[4,5]	25.09	72.79
3	[2,6,7,8]	[3,4]	5	9	1	25.17	72.53
4	[2,4,8,9]	[3,6]	5	1	7	24.98	72.92
5	[2,4,5,8]	[1,6]	3	9	7	25.71	73.70
6	[3,8]	7	[2,5]	[4,9]	[1,6]	24.71	73.18
7	[2,4,5,6,8]	1	3	7	9	24.77	72.53
8	[2,4,5,6,8]	9	7	1	3	25.71	73.96
9	[4,5]	9	[1,2,6]	[7,8]	3	24.98	72.79
10	[2,4,5,8]	9	[1,6]	3	7	24.98	72.79
					Average	**25.11**	**73.02**

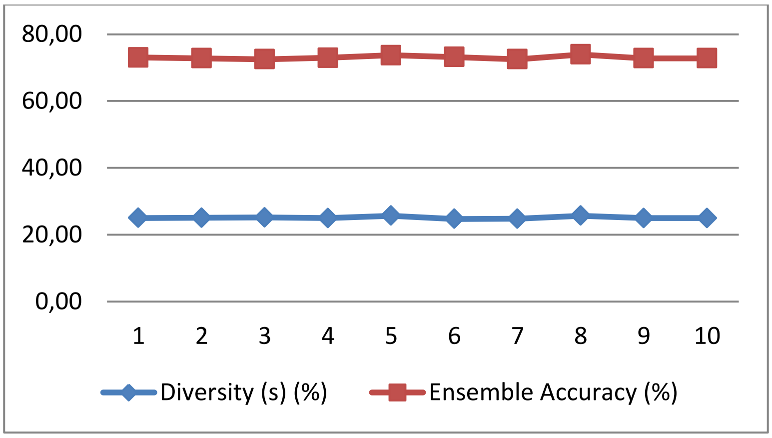

Figure 4.26 Compactness vs LDA ensembles accuracy on tic-tac-toe dataset

Table 4.41

Compactness vs LDA Ensembles Accuracy on Glass Dataset

Experiment #	Partition Based on ACO			Compactness (s) (%)	Ensemble Accuracy (%)
1	[2,3,5,7]	[4,8,9]	[1,6]	15,34	62,66
2	[2,3,5,7]	[4,8,9]	[1,6]	15,67	62,66
3	[1 3 4]	[2 5 9]	[7 8]	15,54	62,76
4	[2,3,5,7]	[4,8,9]	[1,6]	15,77	62,66
5	[1 3 4]	[2 5 9]	[7 8]	14,91	61,79
6	[2,3,5,7]	[4,8,9]	[1,6]	15,17	62,66
7	[2,3,5,7]	[4,8,9]	[1,6]	15,76	62,66
8	[1 3 4]	[2 5 9]	[7 8]	15,30	62,03
9	[2,3,5,7]	[4,8,9]	[1,6]	14,88	62,66
10	[2,3,5,7]	[4,8,9]	[1,6]	15,09	62,66
			Average	**15.34**	**62.52**

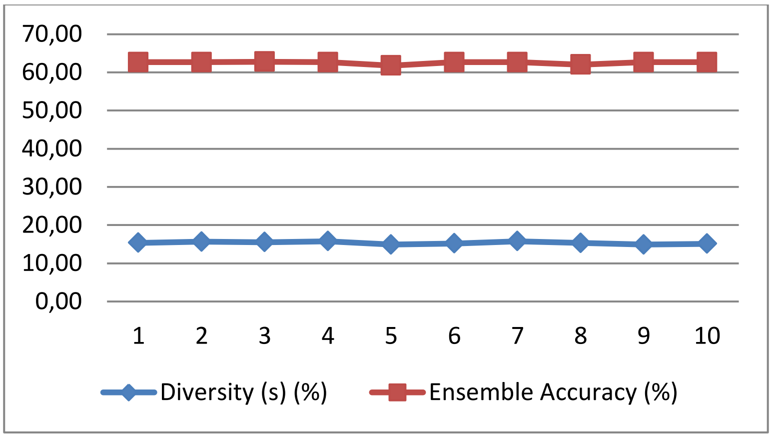

Figure 4.27 Compactness vs LDA ensembles accuracy on glass dataset

Table 4.42

Compactness vs LDA Ensembles Accuracy on Breast Cancer Dataset

Experiment #	**Partition Based on ACO**				**Compactness (s) (%)**	**Ensemble Accuracy (%)**
1	[2,4,6]	[5,7,9]	[1,8]	3	80.60	97.07
2	[2,4,6]	[5,7,9]	[1,8]	3	80.60	97.07
3	[1,3,5,9]	[2,6,8]	7	4	77.32	97.22
4	[2,4,8]	[7,9]	3	[1,5,6]	79.65	97.22
5	[1,3,5,9]	[2,6,8]	7	4	77.32	97.51
6	[1,3,5,9]	[2,6,8]	7	4	77.32	97.22
7	[2,4,8]	[7,9]	3	[1,5,6]	79.65	97.22
8	[1,3,5,9]	[2,6,8]	7	4	77.32	97.22
9	[2,4,8]	[7,9]	3	[1,5,6]	79.65	97.22
10	[3,5,7,9]	[2,6,8]	4	1	76.24	97.07
				Average	**78.57**	**97.20**

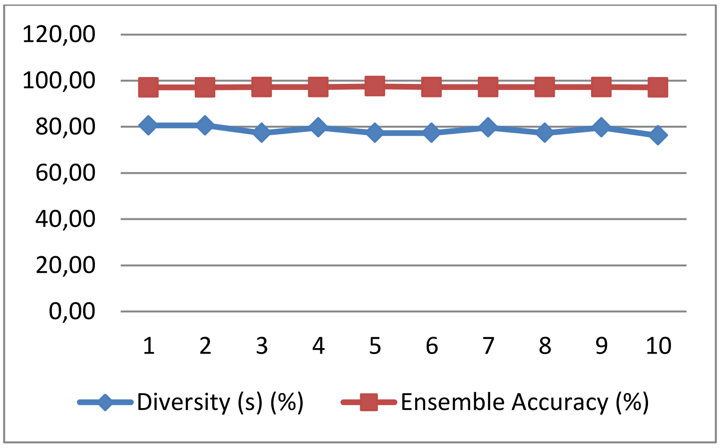

Figure 4.28 Compactness vs LDA ensembles accuracy on breast cancer dataset

Based on the experimental results, it can be seen that the compactness in LDA ensembles is directly proportional to the accuracy of the ensemble. The larger value of compactness will provide higher accuracy of ensemble.

4.5.5 The Relationship between Compactness and Ensemble Accuracy

In this section, testing the relationship between compactness measure and ensemble accuracy is conducted. Testing is conducted using four constructed classifier ensembles on datasets which form partition. The regression test is conducted using SPSS. Simple linear regression test was performed to determine the relationship between them. Summaries of simple linear regression results of SPSS are shown in Table 4.43 – Table 4.45. Here R is the correlation coefficient, R^2 is the determination coefficient and p is the significance value (sig). Based on the results of linear regression, hence a significant relationship between this proposed compactness measure with the ensemble accuracy was identified. There was a positive linear relationship between them.

Tabel 4.43

Model Summary

R	***R^2***	**Adjusted R Square**	**Std. Error of the Estimate**
.971	.943	.943	3.28430

Tabel 4.44

ANOVA

	Sum of Squares	**df**	**Mean Square**	**F**	**Sig *(p)***
Regression	35506.280	1	35506.280	3291.689	.000
Residual	2135.756	198	10.787		
Total	37642.036	198			

Tabel 4.45

Coefficients

Model		Unstandardized Coefficient		Stdandardized Coefficient	t	Sig (*p*)
		B	Std. Error	Beta		
1.	(Constant)	57.949	.375		154.355	.000
	Compactness	.477	.008	.971	57.373	.000

The scatter plot and regression of proposed parameter and ensemble accuracy depicted in Figure 4.29.

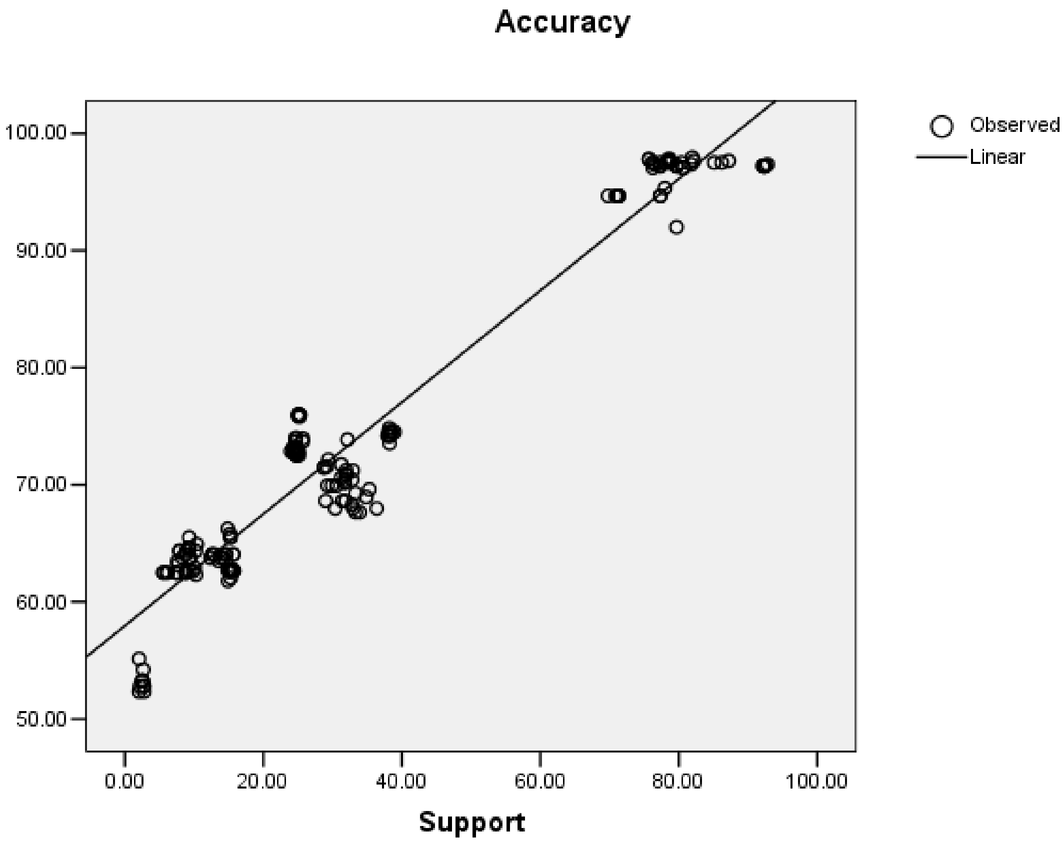

Figure 4.29 Scatter plot of proposed parameter and ensemble accuracy

4.6 Summary

Ant system-based feature set partitioning algorithm for classifier ensembles construction has been presented. Each classifier is trained on different partitions of features where the feature partition is performed through ant colony optimisation. The proposed ant system-based feature set partitioning has been evaluated and compared to the random subspace. The classification results were validated by the 10-fold cross-validation approach. Based on experiment results of classification using 4 (four) homogeneous classifier ensembles and nine (9) UCI datasets, where the individual classifier outputs were combined by majority voting, it can be concluded that the use of the proposed technique can produce better classification results than the random subspace. However, there is a need to formulate an appropriate combination technique which considers the accuracy of each classifier.

A support formulation for compactness measurement in classifier ensemble has also been introduced. Compactness in classifier ensemble is measured simultaneously. The category of the support formulation was non-pairwise, in which the compactness in classifier ensemble is measured directly. The compactness measure has been shown to be the factor that significantly influences the ensemble accuracy. This support formulation has been shown to be able to predict ensemble accuracy and there is a positive relationship between the support formulation and the ensemble accuracy.

CHAPTER FIVE
WEIGHTED VOTING-BASED COMBINER

5.1 Introduction

This chapter presents the weighted voting-based technique to combine the classifier outputs. The weighted voting technique is proposed to be integrated and synergized with classifier ensemble through feedback for ACO. Figure 5.1 shows the block diagram of the proposed multiple classifier combination scheme which shows the position of the proposed weighted voting-based technique.

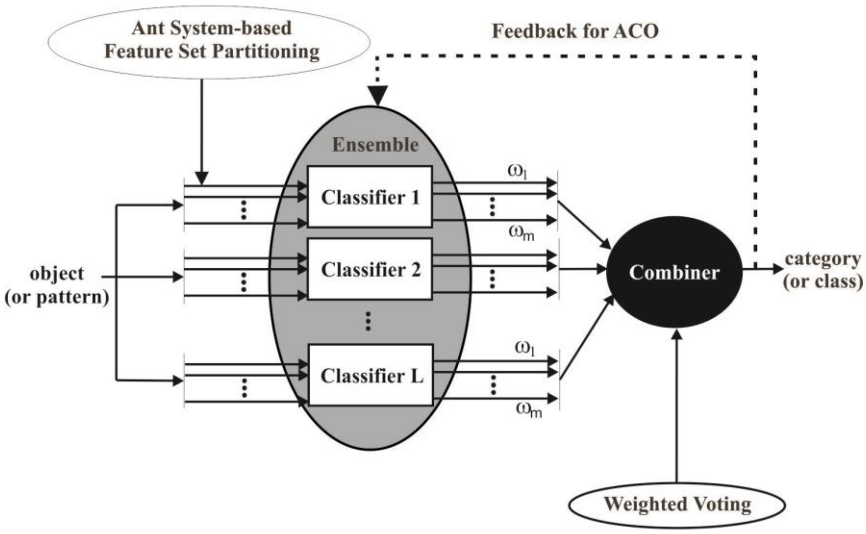

Figure 5.1 Block diagram of proposed multiple classifier combination scheme

There are L different feature partitions, $f_1, \dots, f_L$obtained as the results of ant system-based feature set partitioning algorithm. There will be L classifiers, $D_1, \dots, D_L$ constructed as classifier ensemble. The number of classifiers in the ensemble is determined by the number of partitions obtained by the ant system-based feature partitioning algorithm. Each feature set partition is used to train classifiers to induce diversity. Therefore, L outputs will be produced to combine classifiers using the weighted voting. This proposed multiple classifier combination scheme can be used for any classifier, in other words, not limited for a particular classifier.

Experiments were conducted for the decomposition of features and weights simultaneously instead of each one separately. The output of the combiner is the feedback for ACO based on ant system algorithm, while the weighted voting technique is based on the majority voting technique.

The feedback for ACO in the proposed combination scheme will make each single classifier trained on a different subset of features. This leads to the different performance of each classifier in each iteration. Weighted voting technique considers the performance of each classifier (Kim et al., 2011). For this justification the weigted voting technique is proposed to be included as combiner in this combination scheme. Furthermore, this approach has the potential to make the multiple classifier combination more robust to the choice of the number of individual classifiers (Valdovinos & Sánchez, 2009). Moreover, several combination strategy studies concentrated on the weighted voting approach (Wozniak, 2009; Huang & Wang, 2009; Zarafshan et al., 2010; Nabatchian et al., 2010). Moreover, the weighted voting approach has been used to combine classifiers in order to solve real world problems such as face and voice recognitions (Mu et al., 2009) and listed companies' financial distress prediction (Sun & Li, 2008).

The rest of this chapter is organised as follows. In section 5.2, the weighted voting technique for classifier combiner is proposed. In section 5.3, several experiments are performed to test the weighted voting approach on several standard datasets. In section 5.4, the constructed multiple classifier combination by this proposed combination scheme is evaluated. In section 5.5, some conclusions are outlined.

5.2 The Proposed Weighted Voting-based Technique

There are many approaches that have been developed for combination classifier as those described in chapter two. One of the approaches uses the classifier fusion approach scheme. The classifier fusion scheme assumes all of the classification algorithm are equally experienced, and the output of classifiers are considered. In the fixed classifier fusion scheme, the weight of each classifier for combining classifiers is fixed. There is no training process to determine the weight of each classifier. In general, the gain of the fixed classifier fusion scheme is its simplicity and reduce computational cost (Chen, 2007). One of the most famous and frequently used combination techniques in fixed classifier fusion scheme is majority voting. In several methods for constructing classifier ensemble, the majority voting is the optimal combiner (Ponti, 2011). The weakness of this combiner is it only considers the first rank class and does not consider the strength of the classifier, thus the strength of each classifier is considered equal in vote. If the accuracies of the classifiers can be reliably estimated, then the weighted voting approach may be considered (Polikar, 2006). Therefore, the weighted voting technique is proposed to be included in proposed combination scheme which considers the performance of each classifier.

The focus of this study is on how to effectively combine the output of classifiers at the abstract level. At the abstract-level, each classifier produces a class label (crisp output). Although the classifier provides the least amount of information on this level, the output of any classifier can be tranformed to abstract-level (Ahmadzadeh & Petrou, 2003; Kuncheva, 2004). Integrating ant system-based and the weighted voting will also be able to operate at this level to combine the output of ensemble member regardless of what classifier is used. It is expected that the proposed combination scheme can be used as a guideline in combining multiple classifier for pattern classification task. This is the reason on why the abstract level is focused in this study.

The goal to propose the weighted voting in this cobination sceme is not to compete with the best combiner that has been reported in the literature. Instead, the goal is to demonstrate that an effective classifier combination approach must address two main phases, which are the ensemble construction and the selection of appropriate combiner. Therefore, classifier combination scheme which involves two steps simultaneously is needed. The use of weighted voting in the proposed combination scheme is compared to the use of standard majority voting in the same combination scheme.

The problem of combining multiple classifiers can be defined as follows: Let D= $\{D_1, \ldots, D_L\}$ be a set of individual classifier (or ensemble) where L is the number of individual classifiers. Let Ω=$\{\omega_1, \omega_2, \omega_3, \ldots, \omega_c\}$ be a set of class labels where c is the number of classes. Let $T = \{x_i, y_i\}$ be a training set (a labeled dataset) where $i = 1 \ldots N$, N is the number of instances, $x_i \in \Re^n$ is the n dimensional feature vector of *i-th* instance and $y_i \in \{\omega_1, \dots, \omega_c\}$ is the class label of the *i-th* instance. Each classifier D_j assigns an input feature vector to one of the class label i.e. $D_j: \Re^n \to \Omega$. The outputs of a classifier ensemble is an L dimensional class label vector $[D_1(x), \dots, D_L(x)]^T$. The task is to combine L of individual classifier outputs to predict the class label from a set of possible class labels that make the best classification of the unknown pattern.

Let us assume that only the class labels are available from the classifier outputs, and define the decision of the j-th classifier as $d_{j,k} \in \{0,1\}, j = 1, \dots, L \; and \; k = 1, \dots, C$ where L is the number of classifiers and C is the number of classes. If j-th classifier D_j chooses class ω_k, then $d_{j,k} = 1$ and 0 otherwise.

The classifier outputs combination by the weighted voting can be described as follows: choose class ω_{k*} if

$$\sum_{j=1}^{L} acc_j d_{j,k*}(x) = \; max_k \sum_{j=1}^{L} acc_j d_{j,k}(x) \tag{5.1}$$

where acc_j is the accuracy (or weight) of classifier D_j. The votes are multiplied by a weight before the actual voting. The weight is obtained by estimating the classification accuracy on a validation set.

5.3 Experimental Results in Combiner Construction

This section presents the experiment conducted in evaluating the performance of the weighted voting to combine classifier outputs. For a fair analysis, the performance of multiple classifier combination that is generated by the proposed combination schema is compared with the performance of several previous multiple classifier combination. Four homogeneous ensembles which are NMC ensembles, NBC ensembles, *k*-NN ensembles and LDA ensembles, were constructed in the experiment. The goal is to empirically evaluate the suitability of weighted voting to combine their classifier outputs. For this pupose, the four sets of experiments proceed in two treatments. In the first treatment, the standard majority voting technique was applied to combine classifier outputs while in the second treatment, the weighted voting was applied to combine classifier outputs. The majority voting and weighted voting respectively were tested to combine the output of ensemble classifiers which is constructed by the ant system-based feature set partitioning algorithm. Ten (10) experiments were performed on nine (9) datasets from UCI machine learning repository to test the performance of combined classifier outputs. During the experiment, the performance of multiple classifiers which is combined by the weighted voting is compared to the performance of multiple classifier which is combined by the majority voting. The experimental code was written in MATLAB. The 10-fold cross validation method was used on both treatments for prediction performance assessment. The following subsections present the experimental results.

5.3.1 Experiments in Combining NMC Ensembles

Experiments were conducted to test the weighted voting in combining homogeneous NMC ensembles outputs. Tables 5.1 and 5.2 present the average and standard deviations of the classification accuracies of combined NMC outputs using majority voting and weighted voting respectively. Based on the experimental results, it can be seen that a small deviation of the classification accuracies was obtained which showed that the experiments were accurate and good.

Table 5.1

Accuracy of Combining NMC using Majority Voting

Experiment #	Haberman	Iris	Lenses	Liver	Ecoli	Pima	Tic-Tac-Toe	Glass	Breast Cancer
1	69.93	94.67	70.83	64.06	81.25	73.05	73.17	52.80	97.22
2	71.57	94.67	66.67	62.61	82.14	72.79	72.76	53.27	97.22
3	71.24	94.67	62.50	64.35	81.25	72.53	72.65	55.14	97.22
4	73.86	94.67	70.83	65.51	82.74	72.92	74.01	52.80	97.22
5	67.97	95.33	70.83	64.64	81.25	73.70	72.86	52.34	97.22
6	68.63	94.67	62.50	64.35	82.74	73.18	72.86	52.34	97.36
7	69.93	94.67	70.83	63.77	82.14	72.53	72.86	52.80	97.22
8	70.92	94.67	58.33	64.35	82.14	73.96	72.65	53.27	97.22
9	69.28	92.00	66.67	64.64	81.25	72.79	73.28	54.21	97.22
10	70.59	94.67	66.67	64.64	81.25	72.79	72.96	53.27	97.22
Average	**70.39**	**94.47**	**66.67**	**64.29**	**81.82**	**73.02**	**73.01**	**53.22**	**97.23**
Standard deviation	1.67	0.89	4.39	0.75	0.63	0.47	0.41	0.87	0.04

Table 5.2

Accuracy of Combining NMC using Weighted Voting

Experiment #	Haberman	Iris	Lenses	Liver	Ecoli	Pima	Tic-Tac-Toe	Glass	Breast Cancer
1	69.93	96.00	87.50	64.64	82.44	75.13	74.22	54.00	96.78
2	69.93	96.67	87.50	64.06	82.14	75.39	74.88	54.34	98.03
3	70.26	96.00	87.50	64.64	81.55	75.39	74.23	54.34	98.03
4	69.61	96.00	87.50	63.77	82.44	75.26	74.45	54.34	98.03
5	70.92	96.00	87.50	65.22	82.07	75.13	74.77	54.67	98.03
6	70.59	96.00	87.50	64.35	82.51	75.26	74.23	54.34	98.03
7	71.57	96.00	87.50	65.22	82.44	74.61	74.20	54.34	98.03
8	71.90	96.00	87.50	64.06	82.07	74.48	74.23	54.67	98.03
9	68.95	96.00	87.50	64.64	82.22	75.13	74.45	54.34	98.03
10	72.88	96.00	87.50	64.64	82.34	75.13	74.77	54.00	98.03
Average	**70.65**	**96.07**	**87.50**	**64.52**	**82.22**	**75.09**	**74.44**	**54.34**	**97.91**
Standard deviation	1.18	0.21	0.00	0.48	0.29	0.31	0.27	0.22	0.40

The comparison between the majority voting and weighted voting in combining NMC outputs are shown in Table 5.3 and in Figure 5.2. Based on the results of this comparison, it can be seen that the weighted voting method provides better results than majority voting on all datasets.

Table 5.3

Comparison of Majority Voting and Weighted Voting in Combining NMC

Dataset	Single	Majority Voting	Weighted Voting
Haberman	69.97	70.39	**70.65**
Iris	92.07	94.47	**96.07**
Lenses	65.83	66.67	**87.50**
Liver	55.19	64.29	**64.52**
Ecoli	81.55	81.82	**82.22**
Pima	63.29	73.02	**75.09**
Tic-Tac-Toe	63.19	73.01	**74.44**
Glass	44.16	53.22	**54.34**
Breast Cancer	96.49	97.23	**97.91**

The average accuracy of both combiner in integrating NMC is again depicted in Figure 5.2.

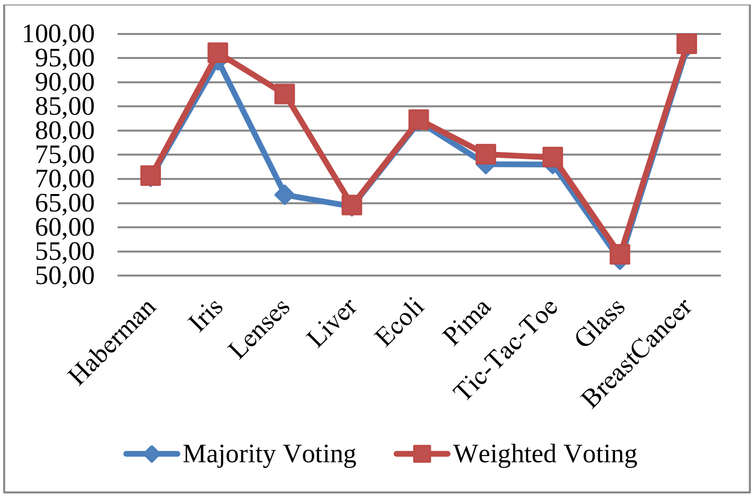

Figure 5.2 Comparison of majority voting and weighted voting in combining NMC

Based on the experimental results, it can be seen that the use of weighted voting to combine the NMC outputs exceeds the majority voting technique on whole datasets. However, a significant increase occurs in lenses, pima and tic-tac-toe datasets.

5.3.2 Experiments in Combining NBC Ensembles

Experiments were carried out to test the weighted voting in combining NBC outputs. Table 5.4 describes the average and standard deviations of the classification accuracies of combined NBC outputs using majority voting. Table 5.5 shows the average and standard deviation of the classification accuracies of the combined NBC outputs using weighted voting.

Table 5.4

Accuracy of Combining NBC using Majority Voting

Experiment #	**Haberman**	**Iris**	**Lenses**	**Liver**	**Ecoli**	**Pima**	**Tic-Tac-Toe**	**Glass**	**Breast Cancer**
1	75.49	96.00	62.50	64.93	75.45	75.39	72.34	72.55	97.66
2	74.18	95.33	62.50	63.77	75.46	75.65	72.86	73.15	97.51
3	74.84	95.33	62.50	63.77	74.45	75.13	72.03	72.55	97.51
4	74.84	95.33	62.50	62.32	74.65	75.00	72.96	72.55	97.66
5	74.84	96.00	62.50	63.77	75.85	75.65	71.71	71.85	97.66
6	75.16	95.33	62.50	63.48	75.45	75.52	72.96	72.55	97.51
7	74.84	95.33	62.50	63.19	76.15	75.65	72.76	79.65	97.80
8	74.51	95.33	62.50	62.61	75.45	75.39	73.28	72.55	97.51
9	74.51	95.33	62.50	62.90	76.95	75.52	72.44	72.55	97.80
10	74.84	95.33	62.50	64.35	75.45	75.52	72.76	72.55	97.66
Average	**74.81**	**95.46**	**62.50**	**63.51**	**75.53**	**75.44**	**72.61**	**73.25**	**97.63**
Standard deviation	0.36	0.28	0.00	0.79	0.71	0.22	0.48	2.27	0.11

Table 5.5

Accuracy of Combining NBC using Weighted Voting

Experiment #	Haberman	Iris	Lenses	Liver	Ecoli	Pima	Tic-Tac-Toe	Glass	Breast Cancer
1	75.49	96.67	62.50	63.48	76.76	75.77	73.07	73.40	98.06
2	75.49	96.00	62.50	66.09	76.33	75.99	72.96	73.28	98.06
3	74.51	96.00	62.50	63.77	76.76	75.77	72.66	73.57	98.06
4	74.84	96.00	62.50	62.61	76.33	75.77	72.88	73.28	98.06
5	74.84	96.00	62.50	62.61	76.76	75.32	72.88	73.57	98.06
6	74.84	96.00	62.50	63.77	76.76	75.77	72.66	73.40	98.06
7	74.84	96.67	62.50	63.19	76.89	75.99	72.88	73.66	98.06
8	75.16	96.00	62.50	63.19	76.76	75.77	72.88	73.57	98.06
9	74.84	96.00	62.50	64.93	76.89	75.77	72.66	73.40	98.06
10	75.49	96.00	62.50	63.19	76.76	75.77	72.66	73.57	98.06
Average	**75.03**	**96.13**	**62.50**	**63.68**	**76.70**	**75.77**	**72.82**	**73.47**	**98.06**
Standard deviation	0.35	0.28	0.00	1.08	0.20	0.18	0.15	0.13	0.00

Table 5.6 and Figure 5.3 present comparison between the majority voting and weighted voting techniques in combining NBC outputs respectively. It can be seen that the weighted voting method provides better results than majority voting on all datasets.

Table 5.6

Comparison of Majority Voting and Weighted Voting in Combining NBC

Dataset	Single	Majority Voting	Weighted Voting
Haberman	74.51	74.81	**75.03**
Iris	95.47	95.46	**96.13**
Lenses	62.50	**62.50**	**62.50**
Liver	55.42	63.51	**63.68**
Ecoli	74.69	75.53	**76.70**
Pima	75.77	75.44	**75.77**
Tic-Tac-Toe	72.54	72.61	**72.82**
Glass	73.02	73.25	**73.47**
Breast Cancer	96.13	97.63	**98.06**

The average accuracy of both combiner in combining NBC outputs is again depicted in Figure 5.3.

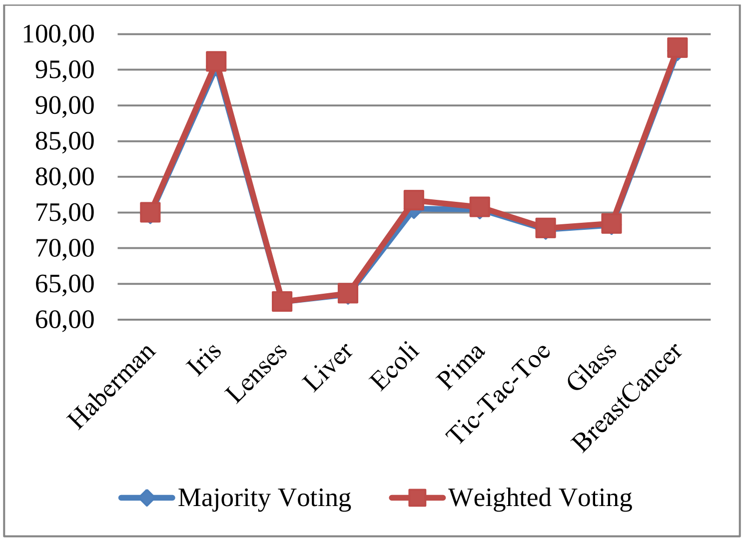

Figure 5.3 Comparison of majority voting and weighted voting in combining NBC

There is no significant improvement when the weighted voting method is used in combining NBC outputs. This proposed technique comparable with the majority voting technique on the entire dataset.

5.3.3 Experiments in Combining *k*-NN Ensembles

Experiments were conducted to test the weighted voting in combining *k*-NN outputs. Table 5.7 and Table 5.8 present the average and standard deviations of the classification accuracies of combining *k*-NN outputs using majority voting and weighted voting respectively. A small standard deviation value shows that the experimental results obtained are close to the average, so the experiment results are accurate and good.

Table 5.7

Accuracy of Combining k-NN using Majority Voting

Experiment #	Haberman	Iris	Lenses	Liver	Ecoli	Pima	Tic-Tac-Toe	Glass	Breast Cancer
1	67.65	96.00	79.17	65.80	80.95	71.48	74.74	73.36	97.95
2	68.95	96.00	79.17	62.61	81.25	71.74	74.32	72.43	97.51
3	67.97	96.00	79.17	65.51	81.25	70.18	76.83	72.90	97.80
4	68.30	96.00	79.17	64.06	80.65	70.44	75.78	72.90	97.51
5	69.61	96.00	79.17	62.61	82.14	71.48	76.10	72.90	97.51
6	68.63	96.00	79.17	62.61	81.25	70.18	75.47	74.30	97.51
7	67.65	96.00	79.17	64.06	80.36	71.22	76.10	73.83	97.36
8	69.93	96.00	79.17	65.51	80.95	70.44	76.83	71.03	97.36
9	67.97	95.33	79.17	66.25	81.55	72.14	75.05	72.90	97.80
10	68.63	96.00	79.17	62.61	81.55	70.83	76.10	72.43	97.66
Average	**68.53**	**95.93**	**79.17**	**64.16**	**81.19**	**71.01**	**75.73**	**72.90**	**97.60**
Standard deviation	0.79	0.21	79.17	1.50	0.50	0.70	0.84	0.88	0.20

Table 5.8

Accuracy of Combining k-NN using Weighted Voting

Experiment #	Haberman	Iris	Lenses	Liver	Ecoli	Pima	Tic-Tac-Toe	Glass	Breast Cancer
1	72.22	96.67	87.50	64.35	81.85	71.22	79.00	73.66	98.09
2	72.88	96.00	83.33	64.06	81.85	71.22	78.81	73.90	98.09
3	72.22	96.00	87.50	66.67	81.25	71.22	79.12	73.59	98.09
4	72.55	96.00	83.33	64.35	82.14	71.22	78.71	73.98	98.09
5	72.22	96.67	87.50	64.93	81.55	71.22	78.91	72.88	98.09
6	73.20	96.67	87.50	67.83	82.09	71.22	79.44	73.00	98.09
7	72.88	96.67	87.50	65.51	81.85	71.22	78.81	73.98	98.09
8	72.22	96.00	87.50	66.09	82.09	71.22	78.18	73.05	98.09
9	74.51	96.67	87.50	66.96	82.30	71.22	78.18	73.44	98.09
10	72.55	96.00	87.50	64.06	82.09	71.22	78.91	73.90	98.09
Average	**72.75**	**96.34**	**86.67**	**65.48**	**81.91**	**71.22**	**78.81**	**73.54**	**98.09**
Standard deviation	0.71	0.35	1.76	1.35	0.31	0.00	0.39	0.43	0.00

The comparison between the majority voting and weighted voting techniques in combining *k*-NN outputs are shown in Table 5.9 and Figure 5.4. Although the weighted voting method is not always better than majority voting, it provides good results on various datasets.

Table 5.9

Comparison of Majority Voting and Weighted Voting in Combining k-NN

Dataset	Single	Majority Voting	Weighted Voting
Haberman	66.83	68.53	**72.75**
Iris	95.67	95.93	**96.34**
Lenses	77.92	79.17	**86.67**
Liver	62.32	64.16	**65.48**
Ecoli	81.19	81.19	**81.91**
Pima	67.37	71.01	**71.22**
Tic-Tac-Toe	75.51	75.73	**78.81**
Glass	72.71	72.90	**73.54**
Breast Cancer	95.78	97.60	**98.09**

The average accuracy of both combiners is again depicted in Figure 5.4.

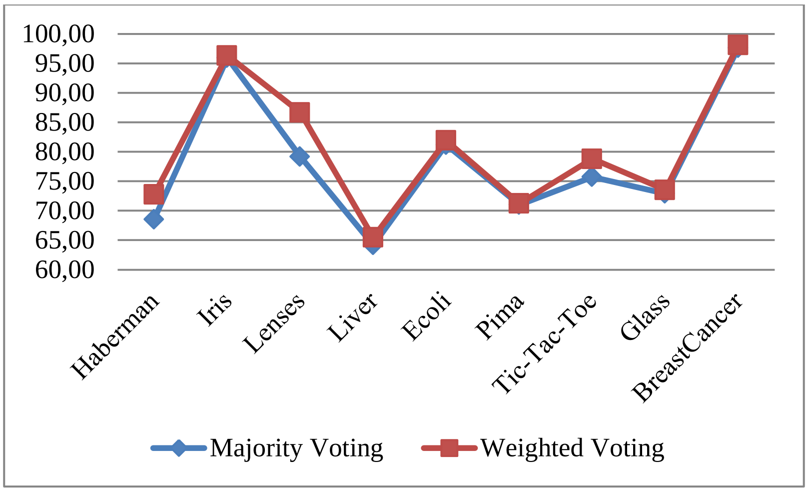

Figure 5.4 Comparison of majority voting and weighted voting in combining *k*-NN

There is a significant increase in classification accuracy when weighted voting is used to combine *k*-NN outputs. Therefore, the weighted voting method is better than the majority voting method in combining *k*-NN outputs.

5.3.4 Experiments in Combining LDA Ensembles

Experiments were performed to determine the performance of both majority voting techniques plus ant system-based feature set partitioning to combine LDA outputs. Table 5.10 and Table 5.11 present the average and standard deviations of the classification accuracies of combined LDA outputs respectively. As with previous experiments, in these experiments, a small value of standard deviation is obtained.

Table 5.10

Accuracy of Combining LDA using Majority Voting

Experiment #	Haberman	Iris	Lenses	Liver	Ecoli	Pima	Tic-Tac-Toe	Glass	Breast Cancer
1	74.84	98.00	87.50	63.48	75.96	75.52	73.05	62.66	97.07
2	74.51	98.00	87.50	63.77	75.90	75.91	72.79	62.66	97.07
3	74.51	98.00	83.33	63.77	75.96	76.17	72.53	62.76	97.22
4	74.18	98.00	87.50	63.77	76.00	75.78	72.92	62.66	97.22
5	74.51	98.00	87.50	64.06	75.89	75.65	73.70	61.79	97.51
6	74.51	98.00	87.50	64.06	75.96	76.04	73.18	62.66	97.22
7	73.53	98.00	87.50	64.06	75.96	76.17	72.53	62.66	97.22
8	74.18	98.00	83.33	63.48	75.79	76.17	73.96	62.03	97.22
9	74.51	98.00	87.50	64.06	75.96	76.04	72.79	62.66	97.22
10	74.18	98.00	87.50	64.06	75.96	76.82	72.79	62.66	97.07
Average	**74.35**	**98.00**	**86.67**	**63.86**	**75.93**	**76.03**	**73.02**	**62.52**	**97.20**
Standard deviation	0.35	0.00	1.76	0.24	0.06	0.36	0.47	0.33	0.13

Table 5.11

Accuracy of Combining LDA using Weighted Voting

# experiment	Haberman	Iris	Lenses	Liver	Ecoli	Pima	Tic-Tac-Toe	Glass	Breast Cancer
1	74.84	98.00	87.50	67.54	76.56	76.82	74.01	62.67	98.12
2	75.16	98.00	87.50	66.09	76.99	76.82	74.33	62.66	98.12
3	74.51	98.00	87.50	65.22	76.88	76.82	74.00	62.33	98.12
4	74.51	98.00	87.50	64.93	76.55	76.82	74.66	62.33	98.12
5	74.51	98.67	87.50	65.22	76.00	76.82	74.66	62.87	98.12
6	74.51	98.00	87.50	64.93	76.56	76.82	74.33	62.66	98.12
7	75.16	98.00	87.50	65.22	76.56	76.82	74.66	62.67	98.12
8	74.84	98.00	87.50	64.93	76.99	76.82	74.00	62.77	98.12
9	75.16	98.00	87.50	65.22	76.88	76.82	74.33	62.77	98.12
10	75.16	98.00	87.50	65.80	76.33	76.82	74.66	62.67	98.12
Average	**74.84**	**98.07**	**87.50**	**65.51**	**76.63**	**76.82**	**74.36**	**62.64**	**98.12**
Standard deviation	0.31	0.21	0.00	0.81	0.31	0.00	0.29	0.18	0.00

Table 5.12 and Figure 5.5 show comparison between majority voting and weighted voting on combining LDA outputs. It can be seen that the weighted voting method provides better results than majority voting in most datasets.

Table 5.12

Comparison of Majority Voting and Weighted Voting in Combining LDA

Dataset	Single	Majority Voting	Weighted Voting
Haberman	73.73	74.35	**74.84**
Iris	97.33	98.00	**98.07**
Lenses	86.25	86.67	**87.50**
Liver	62.35	63.86	**65.61**
Ecoli	72.91	75.93	**76.63**
Pima	75.34	76.03	**76.82**
Tic-Tac-Toe	65.62	73.02	**74.36**
Glass	58.83	62.52	**62.64**
Breast Cancer	96.18	97.20	**98.12**

The average accuracy of both techniques is again showed in Figure 5.5.

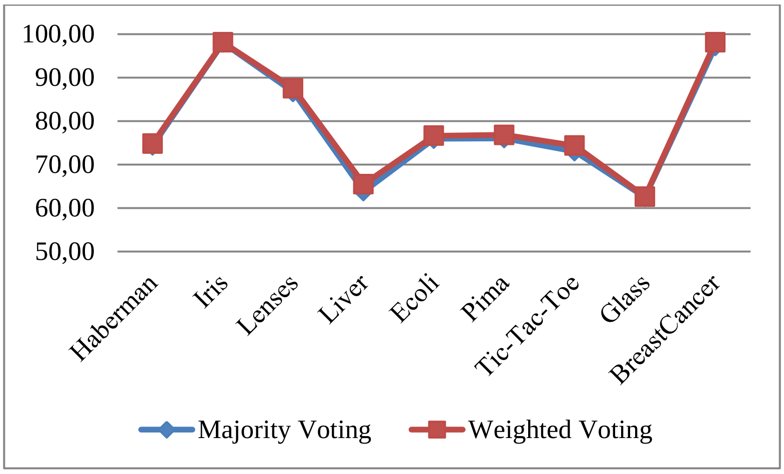

Figure 5.5 Comparison of majority voting and weighted voting in combining LDA

There is no significant improvement when the weighted voting technique is used in combining LDA outputs. This proposed technique is comparable with the majority voting technique on the entire dataset.

5.4 Summary of Results

The experimental results on applying weighted voting to combine the individual classifier outputs has been given in the previous section. In this section, a comparison of four combined classifiers for the whole datasets is given as well. It is intended to easily perform a comprehensive comparison. Table 5.13 shows a comparison of four combined classifiers on the whole dataset.

Table 5.13

Comparison of Four Constructed Homogeneous Classifier Ensembles

Dataset	**Base Classifier of Homogeneous Ensembles**	**Average of Accuracy (%)**	
		Majority Voting	**Weighted Voting**
Haberman	NMC	70.39	**70.65**
	NBC	74.81	**75.03**
	k-NN	68.53	**72.75**
	LDA	74.35	**74.84**
Iris	NMC	94.47	**96.07**
	NBC	95.46	**96.13**
	k-NN	95.93	**96.34**
	LDA	98.00	**98.07**
Lenses	NMC	66.67	**87.50**
	NBC	**62.50**	**62.50**
	k-NN	79.17	**86.67**
	LDA	86.67	**87.50**
Liver	NMC	64.29	**64.52**
	NBC	63.51	**63.68**
	k-NN	64.16	**65.48**
	LDA	63.86	**65.51**
Ecoli	NMC	81.82	**82.22**
	NBC	75.53	**76.70**
	k-NN	81.19	**81.91**
	LDA	75.93	**76.63**
Pima	NMC	73.02	**75.09**
	NBC	75.44	**75.77**
	k-NN	71.01	**71.22**
	LDA	76.03	**76.82**
Tic-Tac-Toe	NMC	73.01	**74.44**
	NBC	72.61	**72.82**
	k-NN	75.73	**78.81**
	LDA	73.02	**74.36**
Glass	NMC	53.22	**54.34**
	NBC	73.25	**73.47**
	k-NN	72.90	**73.54**
	LDA	62.52	**62.64**
Breast Cancer	NMC	97.23	**97.91**
	NBC	97.63	**98.06**
	k-NN	97.60	**98.09**
	LDA	97.20	**98.12**

The summary of results is again presented in a graph form as depicted in Figure 5.6.

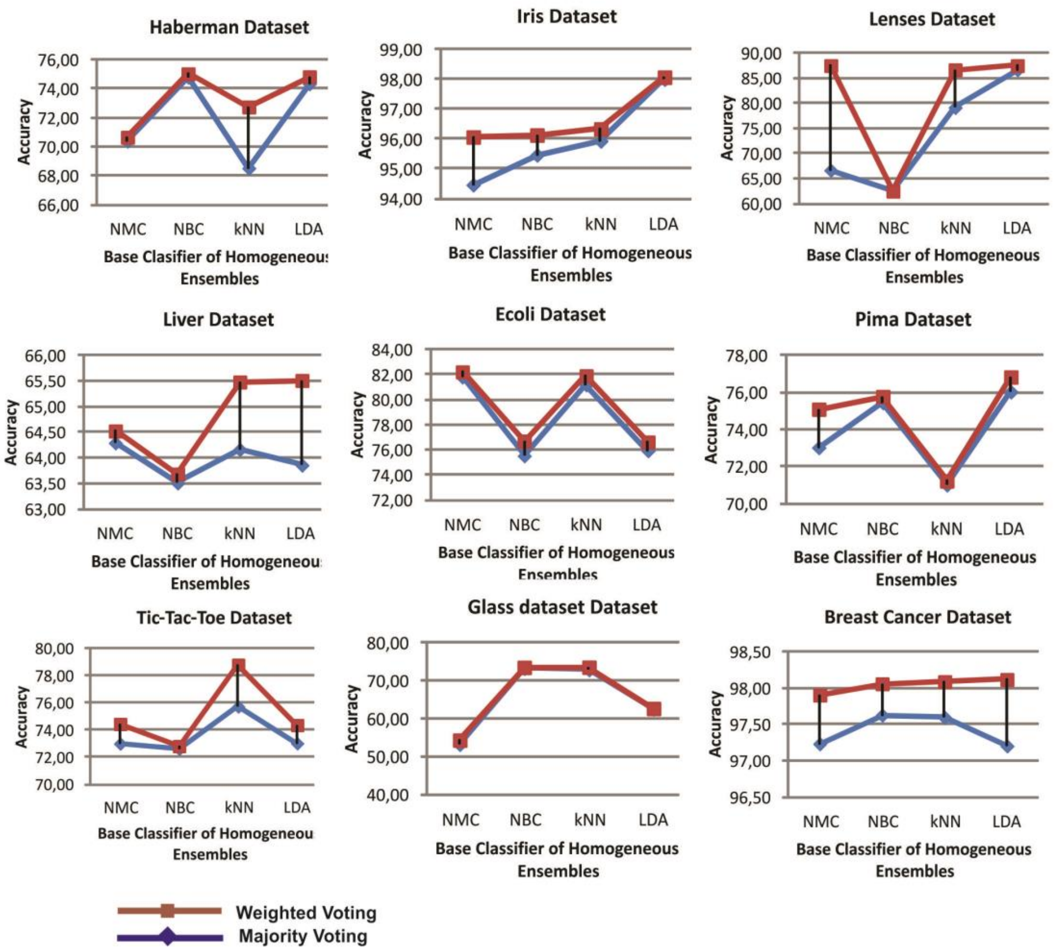

Figure 5.6 Comparison of four constucted homogeneous ensemble classifiers

Based on a comparison of the combined classifier using majority voting and weighted voting, it can be seen that in general, there is an increase in classification accuracy on almost all datasets; even though the accuracy of these improvements vary from large to small depending on the classifiers used. The performance of NMC and *k*-NN increased significantly on most datasets; meanwhile NBC and LDA's performance are not significantly increased in most of the datasets.

5.5 Comparison with other Methods

Evaluation is important to know the strengths and weaknesses of the proposed method. For evaluation purpose, an ensemble classifier is developed using the proposed multiple classifier combination scheme. The developed ensemble classifier is compared to other method. Comparing the developed ensemble classifier with single classifier and other methods is conducted in this section.

Many methods on combining multiple classifiers have been proposed. A common approach or main method on combining classifier is to use the random subspace method, such that each classifier is trained on a different feature subset of the training data. Random subspace creates ensemble diversity, by training a classifier using different random feature subsets.

A weighted voting-based technique on combining multiple classifiers which corresponds to the average distance weight has been introduced by Valdovinos and Sánchez (2009). The goal of this weighting technique was to reward (by assigning the highest weight) the individual classifier with the k-nearest neighbour to the input pattern. The effectiveness of this approach was empirically tested over a number of data sets. Experimental results with several real-problem datasets from the UCI machine learning database repository demonstrated the advantages of this weighted voting technique over the simple majority voting.

A novel method using Genetic Algorithm has been proposed by Suguna and Thanushkodi (2010). Genetic Algorithm was combined with *k*-NN to overcome the limitations of *k*-NN and enhance its classification performance called Genetic *k*-NN (GKNN). Instead of considering all the training samples and taking k-neighbours, the GA was employed to take k-neighbours straightaway and then calculate the distance to classify the test samples. Before classification, the reduced feature set was received from a novel method based on Rough set theory hybrid with Bee Colony Optimization. In the proposed method, by using GA, k-number of samples was chosen for all iterations and the classification accuracy was calculated as fitness. The highest accuracy was recorded each time. Thus, it was not required to calculate the similarities between all samples, and there was no need to consider the weight of the samples. Thus the calculation complexity of *k*-NN was reduced. The performance was compared with traditional k-NN, CART and SVM classifiers. Experimental results showed that the proposed method not only reduces the complexity of the *k*-NN, but also enhances the classification performance.

An ensemble method to improve the performance of *k*-NN which combines multiple *k*-NN classifiers, where each classifier uses a different distance metrics, and a different feature vectors has been proposed by Tahir and Smith (2010). These feature vectors were determined for each distance metric simultaneously by using a combination of Tabu search and simple local neighbourhood search to minimize the ensemble error rate. This approach selects a diverse set of classifiers, such that achieves higher performance enhancements. A simple voting scheme was adopted to obtain the final output of the ensemble. The proposed ensemble method (DF-TS3) was evaluated using several benchmark datasets from UCI machine learning repository. Experimental results have shown a significant improvement in the accuracy when compared with different well-known classifiers. Furthermore, the proposed ensemble method was also compared with ensemble classifier using different distance metrics but with same feature vector.

A novel nearest neighbour ensemble method based on weighted voting technique in classifier fusion called weighted nearest neighbour ensemble (WNNE) has been introduced by Hamzeloo et al. (2012) to enhance the accuracy of nearest neighbour classifier. WNNE is a combination of several nearest neighbour classifiers, which have different subsets of input feature set. The algorithm assigns a weight to each classifier, and uses a weighted voting technique among these classifiers to obtain the final decision. The proposed method has been evaluated on several UCI benchmark datasets. This method was compared to single nearest neighbour classifier and random subspace. The results showed that WNNE outperforms these two approaches.

A direct boosting algorithm which is an ensemble method, has been proposed by Neo and Ventura (2012) for the *k*-NN classifier that creates ensemble of classifiers with locally modified distance weighting. The weights were trained by iterating through the training set and classifying each sample against the rest of the training set. Incorrectly classified samples will update the weights of their neighbours so that they were more likely to correctly classify the instance during the following iteration. To save computation time, a modification called throttling was considered, in which the set of possible neighbours for each instance was limited. The proposed ensemble method was tested on several standard databases from the UCI machine learning repository benchmark datasets. The 10-fold cross validation approach was used to test the accuracy of algorithm. A weighted voting-based mechanism was adopted to obtain the final output of the ensemble. Tests results showed better classification accuracy and generalization ability in the majority of the datasets and never performs worse than standard *k*-NN.

In this section the performance of the proposed method has been compared to the aforementioned methods. The same base classifier and same dataset which have been used in previous experiments have also been used for comparison purposes. *k*-NN classifier is used as a base classifier for ensemble method comparison. Haberman, iris, ecoli, glass, pima and breast cancer from the UCI repository are chosen because of the availability of results from previous studies in which *k*-NN was also used as base classifier. The performance of the proposed method is evaluated by comparing the reults to: (1) Single classifier approach, (2) Dynamic weighted voting (Valdovinos & Sánchez, 2009), (3) An improved *k*-NN classification using Genetic Algorithm (Suguna & Thanushkodi, 2010), (4) Simultaneous metaheuristic feature selection (Tahir and Smith, 2010), (5) Weighted *k*-NN ensemble method (Hamzeloo et al., 2012) and (6) Direct boosting algorithm (Neo and Ventura, 2012). Table 5.14 presents the comparison of results of these methods and again depicted in Figure 5.7.

Table 5.14

Result of proposed methods compared with previous methods

Dataset	Single Classifier (1)	Dynamic Weighted Voting (2)	Genetic *k*-NN (GKNN) (3)	Simultaneous Metaheuristic Feature Selection (4)	Weighted *k*-NN Ensemble (5)	Direct Boosting Algorithm (6)	Proposed Ant System + Weighted Voting (7)
Haberman	66.83	-	-	-	71.89	-	**72.75**
Iris	95.67	**97.33**	-	-	95.20	96.70	96.34
Ecoli	81.19	-	-	-	**82.79**	-	81.91
Glass	72.71	-	-	-	**74.23**	72.50	73.54
Pima	67.37	72.68	-	71.90	-	**75.70**	71.22
Breast Cancer	95.78	96.35	97.92	97.50	-	-	**98.09**

1. Single Classifier
2. Dynamic Weighted Voting
3. Genetic *k*-NN
4. Simultaneous Metaheuristic Feature Selection
5. Weighted *k*-NN Ensemble
6. Direct Boosting Algorithm for *k*-NN
7. Proposed Ant System + Weighted Voting

The obtained results is again depicted in Figure 5.7.

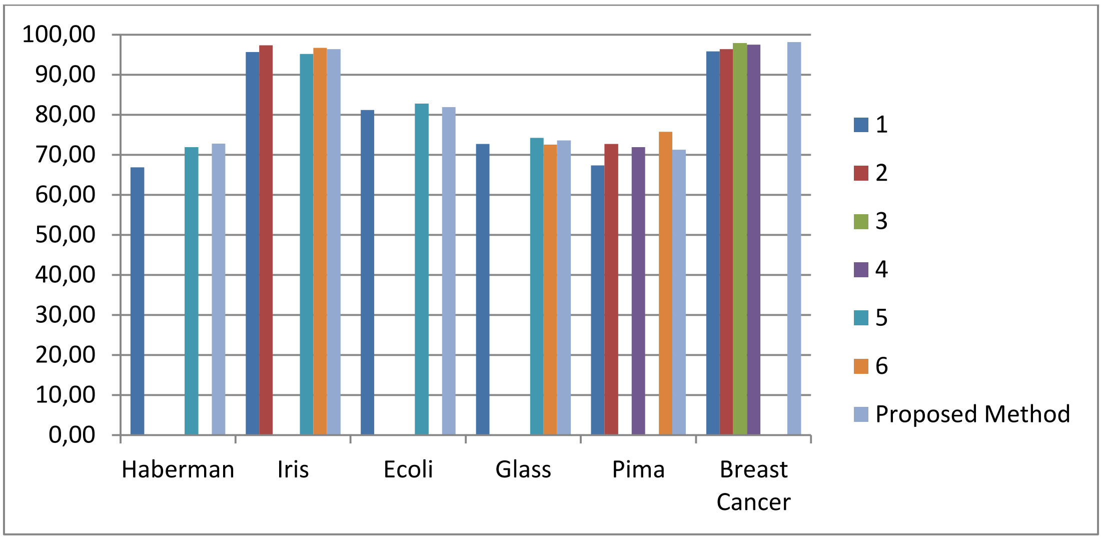

Figure 5.7 Comparison of proposed method and other previous methods

Based on the results, it can be seen that the proposed method gives the best classification accuracies as compared to the other methods on habermann and breast cancer dataset. The performance of weighted *k*-NN ensemble method exceeds the proposed method on the ecoli and glass datasets. The performances of dynamic weighted voting method and direct boosting algorithm for *k*-NN exceed the proposed method on iris and pima dataset. In general, the proposed method gives good classification results and is comparable with the previous methods.

5.6 Summary

A weighted voting combination technique was proposed to integrate the output of classifiers. This combination technique used the performance of classifiers based on the feature set partition. The weights were obtained through the feedback from the combiner. The weight is determined according to the performance of each classifier. The performance of the weighted voting technique has been evaluated to combine homogeneous classifier ensembles and compared them to majority voting on nine datasets. Based on the experimental results, it is concluded that weighted majority voting technique can produce better classification results for various datasets compared to majority voting. It can also be concluded that the performance of a multiple classifier combination scheme which combines ant system-based feature set partitioning and weighted voting method exceeds the performance of single classifiers as well as provides good classification results and is comparable to other state-of-the-art ensemble methods.

CHAPTER SIX
CONCLUSION

The main objective of this research is to develop a method in combining multiple classifier combination. This study was able to achieved the main objective and the specific objectives. Section 6.1 highlights the research contribution and Section 6.2 describes future research that can be pursued as the result of this study.

6.1 Research Contribution

There are three contributions of the research. The first two contributions can be used separately in dealing with multiple classifier combination.

The first contribution is an algorithm for classifier ensemble construction. The algorithm was developed based on ant system which is a variant of ACO algorithm. Ant system-based feature set partitioning (vertical decomposition) algorithm has been developed for the input training process. A set of classifiers are trained by using different feature partitions to induce diversity and combined them to achieve optimal ensemble accuracy. Although there are several methods to construct a classifier ensemble, this algorithm can determine the number of classifiers which are combined to produce a diverse and accurate classifier ensemble. The number of classifiers that can be combined is determined by the number of feature partitions that are obtained. Based on the experimental results, it can be concluded that the use of the proposed algorithm can produce better classifier ensemble.

The second contribution is a compactness measure in a set of classifiers. The compactness measure is intended to address the diversity-accuracy dilemma.

Compactness is a condition in which classifiers support each other for true decision regardless of the classifiers diverse or similar conditions. The compactness in a set of classifiers is measured simultaneously. Based on the experimental results, it can be concluded that there is a positive relationship between compactness and ensemble accuracy. This means that this proposed compactness measure can be used as a guide for constructing diverse and accurate classifier ensemble.

The third contribution is an improved multiple classifier combination scheme. The first contribution and weighted voting combiner are integrated in the general multiple classifier combination cheme. Based on the evaluation results, it is concluded that application of this proposed combination scheme in combining multiple classifier, exceeds single and other multiple classifiers, gives good classification results and is comparable to other state-of-the-art ensemble methods. This proposed combination scheme can be applied to develop various multiple classifier combination.

6.2 Future Work

Future research can conducted in applying the proposed ant system-based feature partitioning algorithm on other classifiers like Support Vector Machine, Neural Network and Decision Tree. This study has only focused on the use of the same algorithm over diversified datasets (homogeneous or one type of classifier). Future work can be used on different algorithms for the same data (or heterogeneous classifier). Feature partitioning does not always outperform the single classifier (or comparable). The experimental results have demonstrated that feature partitioning is superior for certain datasets. Thus, to enhance the performance of feature partitioning, feature partition-selection approach can be considered. As a consequence of the high dimensional feature vector space, it would require a large number of training samples in the training process. Ant system-based feature partitioning makes it possible to partition the feature set into several lower-dimensional feature sets, which would allow a set of classifiers to process low dimensional feature vectors simultaneously. Therefore, testing the ability of this method to overcome the high dimensional data and small training sample problems can also be considered for future work.

REFERENCES

Ahmadzadeh, M. R., & Petrou, M. (2003). Use of Dempster-Shafer theory to combine classifiers which use different class boundaries. *Pattern Analysis & Applications 6(1)*, 41-46.

Ahn, H., Moon, H., Fazzari, M. J., Lim, N., Chen, J. J., & Kodell, R. L. (2007). Classification by ensembles from random partitions of high-dimensional data. *Computational Statistics and Data Analysis, 51*, 6166-6179.

Aljanaby, A., & Ku-Mahamud, K. R. (2011). Interacted multiple ant colonies optimization approach for the single machine total weighted tardiness problem. *Proceedings of the 3rd International Conference on Computing and Informatics (ICOCI)*, (pp. 19-25).

Aslam, J. A., & Montague, M. (2001). Models for metasearch. *Proceedings of the 24th Annual International ACM SIGIR Conference on Reseach and Development in Information Retrieval*, (pp. 276-284).

Ayob, M., & Jaradat, G. (2009). Hybrid ant colony systems for course timetabling problems. *Proceedings of the 2nd Conference on Data Mining and Optimization*, (pp. 120-126).

Banfield, R. E., Hall, L. O., Bowyer, K. W., & Kegelmeyer, W. P. (2005). Ensemble diversity measures and their application to thinning. *Information Fusion*, *6(1)*, 49-62.

Basu, J. K., Bhattacharyya, D., & Kim, T. h. (2010). Use of artificial neural network in pattern recognition. *International Journal of Software Engineering and its Application*, *4(2)*, 23-34.

Bay, S. D. (1998). Combining nearest neighbour classifiers through multiple feature subsets. *Proceedings of the 17th International Conference on Machine Learning*, (pp. 37-45).

Bell, J. E., & McMullen, P. R. (2004). Ant colony optimization techniques for the vehicle routing problem. *Advanced Engineering Informatics, 18*, 41-48.

Berrichi, A., & Yalaoui, F. (2013). Efficient bi-objective ant colony approach to minimize total tardiness and system unavailability for a parallel machine scheduling problem. *The International Journal of Advanced Manufacturing Technology, 68(9-12)*, 2295-2310.

Bi, Y. (2012). The impact of diversity on the accuracy of evidential classifier ensembles. *International Journal of Approximate Reasoning, 53*, 584-607.

Bishop, C. M. (1995). *Neural networks for pattern recognition.* Oxford: Clarendon Press.

Blum, C. (2005). Ant colony optimization: Introduction and recent trends. *Physics of life reviews, 2*, 353-373.

Blum, C., & Dorigo, M. (2004). The hyper-cube framework for ant colony optimization. *IEEE Transactions on Systems, Man and Cybernetics-Part B: Cybernetics, 34(2)*, 1161-1172.

Bolon-Canedo, V., Sanchez-Marono, N., & Alonso-Betanzros, A. (2012). An ensemble of liters and classifiers for microarray data classification. *Pattern Recognition, 45*, 531-539.

Breiman, L. (1996). Bagging predictors. *Machine Learning, 24 (2)*, 123-140.

Breiman, L. (2001). Random forest. *Machine Learning, 45(1)*, 5-32.

Breve, F. A., Ponti-Junior, M. P., & Mascarenhas, N. D. A. (2006). Neural-network combination for noisy data classification. *Proceedings of the II Workshop de Visao Comptacional*, (pp. 87-92).

Brown, G. (2010). Ensemble learning. In C. Sammut, & G. I. Webb (Eds.), *Encyclopedia of Machine Learning* (pp. 312-320). New York: Springer Science+Business Media.

Brown, G., & Kuncheva, L. I. (2010). “Good” and “Bad” diversity in majority vote ensembles. In N. El-Gayar, J. Kittler, & F. Roli (Eds.), *Multiple Classifier Systems, Lecture Notes in Computer Science, Vol. 5997* (pp. 124-133). Berlin Heidelberg: Springer-Verlag.

Bryll, R., Osuna, R. G., & Quek, F. (2003). Attribute bagging: Improving accuracy of classifier ensembles by using random feature subsets. *Pattern Recognition, 36,* 1291-1302.

Bui, T. N., & Nguyen, T. H. (2008). An ant-based algorithm for coloring graphs. *Discrete Applied Mathematics, 156(2),* 190-200.

Bullnheimer, B., Hartl, R., & Strauss, C. (1999). A new rank-based version of the Ant System: A computational study. *Central European Journal of Operations Research, 7(1),* 25-38.

Burton, M., Thomassen, M., Tan, Q., & Kruse, T. A. (2012). Prediction of breast cancer metastasis by gene expression profiles: A comparison of metagenes and single genes. *Cancer Informatics, 11,* 193-217.

Canuto, A. M. P., Abreu, M. C. C., Oliveira, L. M., Xavier, C. J., & Santos, A. M. (2007). Investigating the influence of the choice of the ensemble members in accuracy and diversity of selection-based and fusion-based methods for ensembles. *Pattern Recognition Letters, 28,* 472-486.

Carpaneto, E., & Chicco, G. (2008). Distribution system minimum loss reconfiguration in the hyper-cube ant colony optimization framework. *Electric Power Systems Research, 78(12),* 2037-2045.

Cetin, E. I., Kuruuzum, A., & Irmak, S. (2008). The solution of crew scheduling problem with set partitioning model. *Aviation and Space Technology, 3(4),* 47-54.

Chang, C. F. (2008). Reconfiguration and capacitor placement for loss reduction of distribution systems by ant colony search algorithm. *IEEE Transactions on power systems, 23(4),* 1747-1755.

Chen, L. (2007). *A new design of multiple classifier system and its application to classification of time series data.* Ph.D Thesis, Department of Electrical and Computer Engineering, University of Waterloo, Ontario Canada.

Chen, L., & Kamel, M. S. (2007). A new design of multiple classifier system and its application to the classification of time series data. *Proceedings of IEEE International Conference on System, Man and Cybernetics,* (pp. 385-391).

Chen, W.-N., & Zhang, J. (2013). Ant colony optimization for software project scheduling and staffing with an event-based scheduler. *IEEE Transactions on Software Engineering, 39(1)*, 1-17.

Chharia, A., & Gupta, R. K. (2013). Email classifier: An ensemble using probability and rules. *Proceeding of the 6th International Conference on Contemporary Computing (IC3)*, (pp. 130-136).

Chicco, G. (2011). Ant colony system-based applications to electrical distribution system optimization. In A. Ostfeld (Ed.), *Ant Colony Optimization - Methods and Applications* (pp. 237-262). Rijeka, Croatia: Intech.

Crawford, B., & Castro, C. (2005). ACO with lookahead procedures for solving set partitioning and covering problems. *Proceedings of Workshop on Combination of Metaheuristic and Local Search with Constraint Programming Techniques, 2005.*

Crawford, B., & Castro, C. (2006). Ant colonies using arc consistency techniques for the set partitioning problem. *Professional Practice in Artificial Intelligence IFIP International Federation for Information Processing, 218* (pp. 295-301). Boston: Springer.

Crawford, B., Castro, C., & Monfroy, E. (2009). A new ACO transition rule for set partitioning and covering problems. *Proceedings of International Conference of Soft Computing and Pattern Recognition*, (pp. 426-429).

Crawford, B., Soto, R., Monfroy, E., Castro, C., Palma, W., & Paredes, F. (2013). A hybrid soft computing approach for subset problems. *Mathematical Problems in Engineering, 2013*, 1-12.

Dara, R. A. (2007). *Cooperative training in multiple classifier systems.* Ph.D Thesis, Department of Systems Design Engineering, University of Waterloo, Ontario Canada.

Dietterich, T. G., & Bakiri, G. (1995). Solving multiclass learning problems via error correcting output codes. *Journal Artificial Intelligence 2*, 263-286.

Dorigo, M. (1992). *Optimization, Learning and Natural Algorithm.* Ph.D thesis, Dipartimento di Elettronica Politecnico di Milano, Italy.

Dorigo, M., & Gambardella, L. M. (1996). A study of some properties of Ant-Q. In H. Voigt, W. Ebeling, I. Rechenberg, & H. Schwefel (Eds.), *Parallel Problem Solving from Nature-PPSN IV, International Conference on Evolutionary Computation, Lecture Notes in Computer Science, Vol. 1141* (pp. 656-665). Berlin Heidelberg: Springer.

Dorigo, M., & Gambardella, L. M. (1997). Ant colony system: A cooperative learning approach to the travelling salesman problem. *IEEE Transactions on Evolutionary Computation, 1(1)*, 53-66.

Dorigo, M., & Stützle, T. (2004). *Ant Colony Optimization.* USA: MIT Press.

Dorigo, M., Maniezzo, V., & Colorni, A. (1996). The ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics-Part B, 26(1)*, 29-41.

Dougherty, G. (2013). *Pattern Recognition and Classification : An Introduction.* New York: Springer Science+Business Media.

Du, P., Sun, H., & Zhang, W. (2009). Target identification from high resolution remote sensing image by combining multiple classifiers. In J. A. Benediktsson, & J. Kittler (Eds.), *Multiple Classifier Systems, Lecture Notes in Computer Science, Vol. 5519* (pp. 408-417). Berlin Heidelberg: Springer-Verlag.

Du, P., Xia, J., Zhang, W., Tan, K., Liu, Y., & Liu, S. (2012). Multiple classifier system for remote sensing image classification: A review. *Sensors*, 4764-4792.

Farid, D. M., Zhang, L., Rahman, C. M., & Strachan, R. (2013). Hybrid decision tree and naive bayes classifiers for multi-class classification task. *International Journal of Expert System with Application, 41*, 1937-1946.

Fook, C. Y., Muthusamy, H., Chee, L. S., Yaacob, S., & Adom, A. H. (2013). Comparison of speech parameterization techniques for the classification of speech disfluencies. *Turkish Journal of Electrical Engineering & Computer Sciences, 21*, 1983-1994.

Forrester, E. C. (2006). *A process research framework.* Pittsburgh Pennsylvania: Software Engineering Institute, Carnegie Mellon University.

Frank, A., & Asuncion, A. (2010). *UCI machine learning repository Irvine, CA: University of California, School of Information and Computer Science.* Retrived from http://archive.ics.uci.edu/ml/.

Freund, Y., & Schapire, R. (1996). Experiments with a new boosting algorithm. *Proceedings of the 13th International Conference on Machine Learning*, (pp. 148-156).

Frinken, V., Fischer, A., & Bunke, H. (2010). Combining neural networks to improve performance of handwritten keyword spotting. In N. E. Gayar, J. Kittler, & F. Roli (Eds.), *Multiple Classifier Systems, Lecture Notes in Computer Science, Vol. 5997* (pp. 215-224). Berlin Heidelberg: Springer-Verlag.

G´omez, O., & Bar´an, B. (2005). Omicron ACO. A new ant colony optimization algorithm. *CLEI Electronic Journal, 8(1),* 1-8.

Gangardiwala, A., & Polikar, R. (2005). Dynamically weighted majority voting for incremental learning and comparison of three boosting based approaches. *Proceedings of International Joint Conference on Neural Networks*, (pp. 1131-1136).

Giancito, G., & Roli, F. (2000). Dynamic classifier selection. In J. Kittler, & F. Roli (Eds.), *Multiple Classifier Systems, Lecture Notes in Computer Science, Vol. 1857* (pp. 177-189). Berlin Heidelberg: Springer-Verlag.

Giancito, G., & Roli, F. (2001). Dynamic classifier selection based on multiple classifier behaviour. *Pattern Recognition, 34*, 1879-1881.

Guo, Y. H., & Shi, D. Q. (2012, November). A mutation ant colony algorithm for the generalized assignment problem under the special constraints. *Applied Mechanics and Materials*, pp. 1862-1865.

Hajdu, A., Hajdu, L., Jonas, A., Kovacs, L., & Toman, H. (2013). Generalizing the majority voting scheme to spatially constrained voting. *IEEE Transaction on Image Processing, 22(11)*, 4182-4194.

Hamzeloo, S., Shahparast, H., & Jahromi, M. Z. (2012). A novel weighted nearest neighbor Ensemble Classifier. *Proceedings of the 16th International Symposium on Artificial Intelligence and Signal Processing (AISP),* (pp. 413-416).

Han, J., Kamber, M., & Pei, J. (2011). *Data Mining Concepts and Techniques.* Massachusetts: Elsevier.

Hansen, L., & Salamon, P. (1990). Neural network ensembles. *IEEE Transaction on Pattern Analysis and Machine Intelligence, 12*, 993-1001.

He, C. L., & Suen, C. Y. (2007). A hybrid multiple classifier system of unconstrained handwritten numeral recognition. *Pattern Recognition and Image Analysis, 17(4),* 608-611.

Hernandez-Lobato, D., & Martinez-Munoz, G. (2013). How large should ensembles of classifiers be? *Pattern Recognition, 46(5),* 1323-1336.

Ho, T. K. (1998). The random subspace method for constructing decision forest. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(8),* 832-844.

Ho, T. K., Hull, J. J., & Srihari, S. N. (1992). On multiple classifier systems for pattern recognition. *Proceedings of the 11th International Conference on Pattern Recogition, 2*, (pp. 84-87).

Hong, G. (2013). A hybrid ant colony algorithm for quadratic assignment problem. *The Open Electrical & Electronic Engineering Journal, 7(1)*, 51-54.

Hongbo, S., & Yali, L. (2008). An ensemble classifier based on attribute selection and diversity measure. *Proceedings of the 5th International Conference on Fuzzy System and Knowledge Discovery, 2*, (pp. 106-110).

Huang, C., & Wang, J. (2009). Multi-weighted majority voting algorithm on support vector machine and its application. *TENCON 2009 IEEE Region 10 Conference*, (pp. 1-4).

Huang, Y. S., & Suen, C. Y. (1995). A method of combining multiple experts for the recognition of unconstraned handwritten numerals. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 17(1)*, 90-94.

Huang, Y. S., Liu, K., & Suen, C. Y. (1994). A neural network approach for multi-classifier recognition systems. *Proceedings of the 4th International Workshop on Frontiers in Handwritten Recognition*, (pp. 235-244).

Jacobs, R. A. (1995). Methods for combining experts probability assessments. *Neural Computation, 7,* 867-888.

Jain, A. K., Duin, R. P. W., & Mao, J. (2000). Statistical pattern recognition : A review. *IEEE Transaction on Pattern Analysis and Machine Intelligence, 22(1)*, 4-37.

Jevtic, A., Andina, D., James, A., Gomez, J., & Jamshidi, M. (2010). Unmanned aerial vehicle route optimization using ant system algorithm. *Proceedings of the 5th International Conference on system of systems engineering*, (pp. 1-6).

Kaur, N., & Kaur, U. (2013). Survey of pattern recognition methods. *International Journal of Advanced Research in Computer Science and Software Engineering, 3(2)*, 317-319.

Khakabimamaghani, S., Barzinpour, F., & Gholamian, M. R. (2010). A high diversity hybrid ensemble of classifiers. *Proceedings of the 2nd International Conference on Software Engineering and Data Mining (SEDM)*, (pp. 461-466).

Kim, E., Kim, W., & Lee, Y. (2002). Combination of multiple classifiers for the costumer's purchase behaviour prediction. *Decision Suppot System, 34*, 167-175.

Kim, H. C., Pang, S., Jo, H. M., Kim, D., & Bang, S. Y. (2003). Constructing support vector machine ensemble. *Pattern Recognition, 36*, 2757-2767.

Kim, H., Kim, H., Moon, H., & Ahn, H. (2011). A weight-adjusted voting algorithm for ensembles of classifiers. *Journal of the Korean Statistical Society, 40(4)*, 437-449.

Ko, A. H. R., & Sabourin, R. (2013). Single classifier-based multiple classification scheme for weak classifiers: An experimental comparison. *Expert Systems with Application, 40*, 3606-3622.

Ko, A. H. R., Sabourin, R., & Britto, A. (2008). From dinamic classifier selection to dynamic enemble selection. *Pattern Recognition, 41(5)*, 1718-1731.

Ko, A. H. R., Sabourin, R., Britto, A., & Oliveira, L. (2007). Pairwise fusion matrix for combining classifiers. *Pattern Recognition, 40*, 2198-2210.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. *Proceedings of the 14th International Joint Conference on*

Artificial Intelligence, 2 (pp. 1137-1145). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Kong, H., Li, X., Wang, J., & Kambhamettu, C. (2005). Ensemble LDA for face recognition. In D. Zhang, & A. K. Jain (Eds.), *Advances in Biometrics, Lecture Notes in Computer Science, Vol. 3832* (pp. 166-172). Berlin, Heidelberg: Springer-Verlag.

Koyuncu, H., & Ceylan, R. (2013). Artificial neural network based on rotation forest for biomedical pattern classification. *Proceedings of the 36th International Conference on Telecommunications and signal processing (TSP)*, (pp. 581-585).

Kuncheva, L. I. (2001). Combining classifiers: Soft computing solution. In S. K. Pal, & A. Pal (Eds.), *Pattern recognition: From classical to modern approaches* (pp. 427-452). Singapore: World Scientific Publishing Co.

Kuncheva, L. I. (2002a). A theoretical study on six classifier fusion strategies. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(2)*, 281-286.

Kuncheva, L. I. (2002b). Switching between selection and fusion in combining classifiers: An experiment. *IEEE Transactions on Systems, Man and Cybernetics-Part B, Cybernetisc, 32(2)*, 146-155.

Kuncheva, L. I. (2004). *Combining Pattern Classifiers: Methods and Algorithms.* Hoboken, New Jersey: John Wiley & Sons, Inc., Publication.

Kuncheva, L. I., & Jain, L. C. (2000). Designing classifier fusion systems by genetic algorithms. *IEEE Transactions on Evolutionary Computation, 4(4)*, 327-336.

Kuncheva, L. I., & Whitaker, C. J. (2001). Feature subsets for classifier combination: An enumerative experiment. In G. Goos, J. Hartmanis, & J. V. Leeuwen (Eds.), *Multiple Classifier Systems, Lecture Notes in Computer Science, 2096* (pp. 228-237). Berlin Heidelberg: Springer-Verlag.

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

Kuncheva, L. I., Skurichina, M., & Duin, R. P. W. (2002). An experimental study on diversity for bagging and boosting with linear classifiers. *Information Fusion, 3*, 245-258.

Lee, D. S., & Srihari, S. N. (1995). A theory of classifier combination: The neural network approach. *Proceedings of the 3rd International Conference on Document Analysis and Recognition, 1,* (pp. 42-45).

Lera, G., & Pinzolas, M. (2002). Neighborhood based Levenberg-Marquardt algorithm for network training. *IEEE Transactions on Neural Networks, 13(5)*, 1200-1204.

Li, H., & Sun, J. (2009). Majority voting combination of multiple case-based reasoning for financial distree prediction. *Expert Systems With Applications, 36*, 4363-4373.

Li, H., Wen, G., Yu, Z., & Zhou, T. (2013). Random subspaces evidence classifier. *Neurocomputing, 110*, 62-69.

Li, K., & Gao, H. (2010). A novel measure of diversity for support vector machine ensemble. *Proceedings of the 3rd International Sysmposium on Intelligence Information Technology and Security Informatics*, (pp. 366-370).

Liao, Y., & Moody, J. (2000). Constructing heterogeneous committees via input feature grouping. In S. A. Solla, T. K. Leen, & K. R. Muller (Eds.), *Advances in Neural Information Processing Systems, Vol. 12* (pp. 921-927). USA: MIT Press.

Lincoln, W., & Skrzypek, J. (1989). Synergy of clustering multiple back propagation networks. *Advances in Neural Information Processing Systems, 2*, 650-659.

Littlestone, J. T., & Warmuth, M. K. (1994). The weighted majority algorithm. *Information and Computation, 108(2)*, 212-261.

Liu, J., Sun, J., & Wang, S. (2006). Pattern recognition : An overview. *International Journal of Computer Science and Network Security, 6(6),* 57-61.

Lizárraga, E., Castillo, O., & Soria, J. (2013). A method to solve the traveling salesman problem using ant colony optimization variants with ant set partitioning. *Recent Andvances on Hybrid Intelligent Systems Tudies in Computational Intelligence, 451*, 237-246.

Lofstrom, T., Johansson, U., & Niklasson, L. (2007). Empirically investigating the importance of diversity. *Proceedings of the 10th International Conference on Information Fusion*, (pp. 1-8).

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

Lu, F., & Yamaoka, F. (1994). Integration of handwritten digit recognition results using evidential rasoning . *Proceedings of International Workshop on Frontiers in Handwriting Recognition, (4)*, (pp. 456-463).

Maimon, O., & Rokach, L. (2005). *Decomposition methodology for knowledge discovery and datamining.* Berlin Heidelberg: Springer-Verlag.

Maniezzo, V., & Carbonaro, A. (2000). An ANTS heuristic for the frequency assignment problem. *Future Generation Computer Systems 16(8)*, 927-935.

Maniezzo, V., & Colorni, A. (1999). The ant system applied to the quadratic assignment problem. *IEEE Transactions on Data and Knowledge Engineering, 11(5)*, 769-778.

Maniezzo, V., & Milandri, M. (2002). An ant-based framework for very strongly constrained problems. In M. Dorigo, G. D. Caro, & M. Sampels (Eds.), *Ant Algorithms, Third International Workshop on Ant Algorithms, Lecture Notes in Computer Science, Vol. 2463* (pp. 222-227). Berlin Heidelberg: Springer-Verlag.

Margoosian, A., & Abouei, J. (2013). Ensemble-based classifiers for cancer classification using human tumor microarray data. *Proceedings of IEEE International Conference on Electrical Engineering (ICEE '13)*, (pp. 1-6).

Medrano-Gracia, P., Crown, B. R., Bluemke, D. A., Finn, J. P., Lee, D. C., Lima, J. A., et al. (2013). Optimal patient classification via statistical decomposition of a 3D left-ventricular atlas. *Journal of Cardiovascular Magnetic Resonance, 15(1)*, 89-89.

Melville, P., & Mooney, R. J. (2005). Creating diversity in ensembles using artificial data. *Information Fusion 6(1)*, 99-111.

Mesquita, M., & Paias, A. (2008). Set partitioning/ covering-based approaches for the integrated vehicle and crew scheduling problem. *Computer and Operations Research, 35(5)*, 1562-1575.

Mu, X., Lu, J., Watta, P., & Hassoun, M. H. (2009). Weighted voting-based ensemble classifiers with application to human face recognition and voice recognition.

Proceedings of International Joint Conference on Neural Networks, (pp. 2168-2171).

Musehane, R., Netshiongolwe, F. V., Nelwamondo, L., Masisi, L., & Marwala, T. (2008). Relationship between diversity and perfomance of multiple classifiers for decision support. *Computing Research Repository - CORR, Vol.abs/0810.3*.

Nabatchian, A., Abdel-Raheem, E., & Ahmadi, M. (2010). A weighted voting scheme for recognition of faces with ilumination variation. *Proceedings of the 11th International Conference Control, Automation, Robotics and Vision*, (pp. 896-899).

Neelamegam, S., & Ramaraj, E. (2013). Classification algorithm in data mining: An Overview. *International Journal of P2P Network Trends and Technology (IJPTT), 4(8),* 369-374.

Neo, T. K. C., & Ventura, D. (2012). A direct boosting algorithm for the k-nearest neighbor classifier via local warping of the distance metric. *Pattern Recognition Letters*, 92-102.

Parshapoor, M., & Bilstrup, U. (2013). Ant colony optimization for channel assignment. *Proceedings of the 4th International Conference on Swarm Intelligence, ICSI 2013.*, (pp. 314-322).

Parvin, H., & Beigi, A. (2011). Clustering ensemble framework via ant colony. In I. Batyrshin, & G. Sidorov (Eds.), *Lecture Notes in Computer Science, Vol. 7095* (pp. 153-164). Berlin Heidelberg: Springer-Verlag.

Parvin, H., & Bidgoli-Minaei, B. (2011). A new space defined by ant colony algorithm to partition data. In A. Bouchachia (ed.), *Lecture Notes in Computer Science, Vol. 6943* (pp. 272-284). Berlin Heidelberg: Springer-Verlag.

Parvin, H., & Parvin, S. (2012). A robust classifier ensemble for improving the performance of classification. *Proceedings of Special Session-Revised Papers, 11th Mexican International Conference on Artificial Intelligence: Advances in Artificial Intelligence Applications*, (pp. 62-66).

Parvin, H., Alizadeh, H., & Minaei-Bidgoli, B. (2009a). A new method for constructing classifiers ensembles. *International Journal of Digital Content Technology and its applications, 3(2),* 62-66.

Parvin, H., Alizadeh, H., & Minaei-Bidgoli, B. (2009b). Using clustering for generating diversity in classifier ensemble. *International Journal of Digital Content Technology and its Application, 3(2),* (pp. 51-57).

Pasti, R., & de Castro, L. N. (2007). The Influence of Diversity in an Immune-based algorithm to Train MLP Nerworks. *Proceedings of the 6th International Conference on Artificial Immune Systems* (pp. 71-82). Berlin, Heidelberg: Springer-Verlag.

Peker, M., Sen, B., & Bayir, S. (2012). Using artificial intelligence techniques for large scale set partitioning problems. *Procedia Technology, 1,* 44-49.

Polikar, R. (2006). Ensemble based systems in decision making. *IEEE Circuits and Systems Magazine, 6(3),* 21-44.

Ponti, M. P. (2011). Combining classifier: From the creation of ensembles to the decision fusion. *Proceedings of the 24th Conference on Graphics, Patterns and Image Tutorials*, (pp. 1-10).

Ranawana, R., & Palade, V. (2005a). A neural network based multi-classifier system for gene identification in DNA sequences. *Neural Computing Application, 14(2),* 122-131.

Ranawana, R., & Palade, V. (2005b). Multiple classifier systems : A review and roadmap for developers. *International Journal of Hybrid Intelligence Systems, 3(1),* 35-61.

Randall, M., & Lewis, A. (2010). Modifications and additions to ant colony optimisation to solve the set partitioning problem. *Proceedings of the 6th IEEE International Conference on e science Workshops*, (pp. 110-116).

Rebeiro, R., & Enembreck, F. (2013). A sociologically inspired heuristic for optimization algorithm: A case study on ant systems. *Expert System with Application, 40,* 1814-1826.

Rokach, L. (2006). Decomposition methodology for classification task - A meta decomposer framework. *Pattern Analysis & Applications, 9,* 257-271.

Rokach, L. (2008). Genetic algorithm-based feature set partitioning for classification problems. *Pattern Recognition, 41(5)*, 1676-1700.

Rokach, L. (2010). *Pattern Classification using Ensemble Methods.* Singapore: World Scientific Publishing Co. Pte. Ltd.

Rokach, L., & Maimon, O. (2005). Feature set decomposition for decision trees. *Journal of Intelligent Data Analysis, 9*, 131-158.

Rokach, L., Chizi, B., & Maimon, O. (2007). A methodology for improving the performance of non-ranker feature selection filters. *Pattern Recognition and Artificial Intelligence, 21(5)*, 809-830.

Roli, F. (2009). Multiple classifier system. In S. Z. Li, & A. K. Jain (Eds), *Encyclopedia of Biometrics* (pp. 981-986). New York: Springer Science+Business Media.

Roli, F., & Giancito, G. (2002). Design of multiple classifier systems. In H. Bunke, & A. Kandel (Eds.), *Hybrid methods in pattern recognition* (pp. 199-226). Singapore: World Scientific Publishing.

Santana, L. E. A. (2010). A comparative analysis of genetic algorithm and ant colony optimization to select attributes for an heterogeneous ensemble of classifiers. *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, (pp. 1-8).

Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning, 5(2)*, 197-227.

Serpen, G., & Pathical, S. (2009). Classification in high-dimensional feature spaces: Random subsample ensemble. *Proceedings of IEEE International Conference on Machine Learning and Application*, (pp. 740-745).

Shang, F., & Wang, Y. (2010). An ant system optimization QoS routing algorithm for wireless sensor networks. *Proceedings of the 3rd International Workshop on Advanced Computational Intelligence*, (pp. 25-27).

Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal, 27*, 379-423, 623-656.

Sharkey, A. J. C. (2002). Types of multinet systems. In G. Goos, J. Hartmanis, & J. van Leeuwen (Eds.), *Multiple Classifier Systems, Lecture Notes in Computer Science, Vol. 2364* (pp. 108-117). Berlin Heidelberg: Springer-Verlag.

Shi, H., & Lv, Y. (2008). An ensemble classifier based on attribute selection and diversity measure. *Proceedings of the 5th International Conference on Fuzzy Systems and Knowledge Discovery*, (pp. 106-110).

Shilen, S. (1990). Multiple binary tree classifier. *Pattern Recognition 23(7)*, 757-763.

Shin, D., & Kim, S. (2009). Nearest mean classifier via-one class SVM. *Proceedings of the 1st International Join Conference on Computational Sciences and Optimization*, (pp. 593-596).

Shipp, C. A., & Kuncheva, L. I. (2002). Relationships between combination methods and measures. *Information Fusion, 3*, 135-148.

Sigari, M. H., Sureshjani, S. A., & Soltanian-Zadeh, H. (2011). Sprot video classification using an ensemble classifier. *Proceeding of the 7th Machine Vision and Image Processing*, (pp. 1-4).

Simpson, E. H. (1949). Measurement of diversity. *Nature, 163*, 688.

Skurichina, M., & Duin, R. P. W. (2002). Bagging, boosting and the random subspace method for linier classifiers. *Pattern Analysis and Applications (5)*, 121-135.

Skurichina, M., Kuncheva, L. I., & Duin, R. P. W. (2002). Bagging and boosting for the nearest mean classifier: Effect of sample size on diversity and accuracy. In F. Roli, & J. Kittler (Eds.), *Multiple Classifier Systems, Lecture Notes in Computer Science, Vol. 2364* (pp. 62-71). Berlin Heidelberg: Springer-Verlag.

Srimani, P. K., & Koti, M. S. (2012). Medical diagnosis using ensemble classifiers - A novel machine-learning approach. *Journal of Advanced Computing, 1*, 9-27.

Stefano, C. D., Cioppa, A. D., & Marcelli, A. (2002). An adaptive weighted majority vote rule for combining multiple classifiers. *Proceedings of the 16th International Conference on Pattern Recognition, 2*, (pp. 192-195).

Strehl, A., & Ghosh, J. (2002). Cluster ensembles - A knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research, 3*, 583-617.

Stützle, T., & Hoos, H. H. (2000). Max-Min ant system. *Future Generation Computer Systems 16(8)*, 889-914.

Su, C. T., Chang, C. F., & Chiou, J. P. (2005). Distribution network reconfiguration for loss reduction by ant colony search algorithm. *Electric Power Systems Research, 75*, 190-199.

Suguna, N., & Thanushkodi, K. (2010). An improved *k*-nearest neighbor classification using genetic algorithm. *International Journal of Computer Science, 7(4)*, 18-21.

Sun, J., & Li, H. (2008). Listed companies' financial distress prediction based on weighted majority voting combination of multiple classifiers. *Expert Systems with Applications, 35(3)*, 818-827.

Tahir, M. A., & Smith, J. (2010). Creating diverse nearest-neighbour ensembles using simultaneous metahueristic feature selection. *Pattern Recognition Letters, 31(11)*, 1470-1480.

Talha, M., & Solung, G. (2013). Reduced transformed features based breast mammograms classification. *Life Science Journal, 10(7s)*, 907-913.

Tao, D., & Tang, X. (2004). SVM-based relevance feedback using random subspace method. *Proceedings of IEEE International Conference on Multimedia and Expo (ICME), Vol.1*, (pp. 269-272).

Theodoris, S., & Koutroumbas, K. (2009). *Pattern Recognition (4th ed.).* New York: Academic.

Ting, K. M., Wells, J. R., Tan, S. C., Teng, S. W., & Webb, G. I. (2011). Feature-subspace aggregating: ensembles for stable and unstable learners. *Machine Learning 82(3)*, 375-397.

Turhal, U., Babur, S., Avci, C., & Akbas, A. (2013). Performance improvement for diagnosis of colon cancer by using ensemble classification methods. *Proceedings of International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE) 2013*, (pp. 271-275).

Ulas, A., Yildiz, O. T., & Alpaydin, E. (2012). Cost-concious comparison of supervised learning algorithms over multipe data sets. *Pattern Recognition, 45*, 1772-1781.

Valdovinos, R. M., & Sánchez, J. S. (2009). Combining multiple classifiers with dynamic weighted voting. *Proceedings of the 4th International Conference on Hybrid Artificial Intelligence Systems (HAIS '09)* (pp. 510-516). Berlin Heidelberg: Springer-Verlag.

Verma, B., & Blumenstein, M. (2008). *Pattern Recognition Technologies and Applications: Recent Advances.* United States of America: Information Science Reference IGI Global.

Wanas, N. M., & Kamel, M. S. (2002). Weighted combining of neural network ensembles. *Proceedings of International Joint Conference on Neural Network*, (pp. 1748-1752).

Wang, L. J., Wang, X. L., & Chen, Q. C. (2005). GA-based feature subset clustering for combination of multiple nearest neighbors classifiers. *Proceedings of International Conference on Machine Learning and Cybernetics, Vol.5*, (pp. 2982-2987).

Wang, P. S. (2011). *Pattern Recognition, Machine Intelligence and Biometrics.* Beijing, Berlin Heidelberg: Higher Education Press, Springer-Verlag.

Wang, Z. Q., Chen, G. H., & Yu, Z. Z. (2012). Linear discrimiminant analysis based on ensemble learning. *Proceedings of International Conference on Biological and Biomedical Sciences, Advances in Biomedical Engineering, Vol.9* (pp. 265-271). USA: Information Engineering Research Institute (IERI).

Woods, K., Kegelmeyer, W. P., & Bowyer, K. (1997). Combination of multiple classifier using local accuracy estimates. *IEEE Transaction on Pattern Analysis and Machine Intelligence, 19(4)*, 405-410.

Wozniak, M. (2008). Classifier fusion based on weighted voting-Analytical and experimental results. *Proceedings of the 8th International Conference on Intelligent Systems Design and Application*, (pp. 687-692).

Wozniak, M. (2009). Evolutionary approach to produce classifier ensemble based on weigted voting. *Proceedings of World Congress on Nature & Biologically Inspired Computing, (NaBIC),2009*, (pp. 648-653).

Wozniak, M., & Crawczyk, B. (2012). Combined classifier based on feature space partitioning. *International Journal of Applied Mathematics and Computer Science, 22(4)* , 855-866.

Wozniak, M., & Zmyslony, M. (2010). Chosen problems of designing effective multiple classifier systems. *Proceedings of International Conference on Computer Information Systems and Industrial Management Applications, 2010*, (pp. 42-47).

Wozniak, M., Grana, M., & Corchado, E. (2014). A survey of multiple classifier systems as a hybrid systems. *Information Fusion, 16*, 3-17.

Wu, C. H., & Liang, W. B. (2011). Emotion recognition of affective speech based on multiple classifier using acoustic-prosodic information and semantic labels. *IEEE Transaction on Affective Computing 2(1)*, 10-21.

Xing, L.-N., Chen, Y.-W., Wang, P., Zhao, Q.-S., & Xiong, J. (2010). A knowledge-based ant colony optimization for flexible job shop scheduling problems. *Applied Soft Computing, 10(3)*, 888-896.

Xu, L., Krzyzak, A., & Suen, C. Y. (1992). Methods of combining multiple classifiers and their application to handwriting recognition. *IEEE Transactions on System, Man and Cybernetics, 22(3)*, 418-435.

Yagmahan, B., & Yenisey, M. M. (2010). A multi-objective ant colony system algorithm for flow shop scheduling problem. *Expert Systems with Applications, 37*, 1361–1368.

Yang, L. Y., Qin, Z., & Huang, R. (2004). Design of multiple classifier system. *Proceedings of the 3th International Conference on Machine Learning and Cybernetics* (pp. 3272-3276). IEEE Press.

Yang, L. Y., Zhang, J. Y., & Wang, W. J. (2009). Selecting and combining classifiers simultaneously with particle swam optimization. *Information Technology Journal, 8(2)*, 241-245.

Yang, P., Yang, Y. H., Zhou, B., & Zomaya, Y. (2010). A review of ensemble methods in bioinformatics. *Current Bioinformatics, 5(4)*, 296-308.

Yang, S., & Browne, A. (2004). Neural network ensembles: Combining multiple models for enhanced performance using a multistage approach. *Expert System, 21(5)*, 279-288.

Zaamout, K., & Zhang, J. Z. (2012). Improving classification through ensemble neural networks. *Proceedings of the 8th International Conference on Natural Computation (ICNC)*, (pp. 256-260).

Zanda, M. (2010). *A Probabilistic Perspectif on Ensemble Diversity.* Ph.D Thesis, Faculty of Engineering and Physical Sciences, School of Computer, University of Manchester, England.

Zarafshan, G. R. F., Latif-Shabgahi, C., & Karimi, A. (2010). A novel weighted voting algorithm based on neural networks for fault tolerant system. *Proceedings of the 3rd International Conference on Computer Science and Information Technology*, (pp. 135-139).

Zhang, B. (2012). Reliable face recognition by random subspace support vector machine enemble. *Proceedings of International Conference on Machine Learning and Cybernetics (ICMLC), 2012*, (pp. 415-420).

Zhang, J., Cheng, L., & Ma, J. (2006). A new multiple classifiers combination algorithm. *Proceedings of the 1st International Multi-Symposiums on Computer and Computational Sciences (IMSCC)*, (pp. 287-291).

Zhao, C., Deshwar, A. G., & Morris, Q. (2013). Relapsing-remitting multiple sclerosis classification using elastic net logistic regression on gene expression data. *Systems Biomedicine, 1(4)*, 59-65.

Zhao, D., Luo, L., & Zhang, K. (2010). An improved ant colony optimization for the communication network routing problem. *Mathematical and Computer Modelling, 52(11-12)*, 1976-1981.

Zhao, J., & Yan, C. (2009). A bottleneck assigned binary ant system for multimodal optimization. *Proceedings of the 48th IEEE Conference on Decision and Control held jointly with 28th Chinese Control Conference*, (pp. 16-18).

Zhou, Z.-H. (2012). *Ensemble Methods: Foundation and Algorithms.* Florida, USA: CRC Press Taylor & Francis Group.