A Novel Framework for Optimised Ensemble Classifiers

by

Muhammad Zohaib Jan

Thesis
Submitted in fulfillment of the requirements for the degree of

Doctor of Philosophy

School of Engineering and Technology

Central Queensland University

April 2020

RHD THESIS DECLARATION

CANDIDATE'S STATEMENT

By submitting this thesis for formal examination at CQUniversity Australia, I declare that it meets all requirements as outlined in the Research Higher Degree Theses Policy and Procedure.

STATEMENT AUTHORSHIP AND ORIGINALITY

By submitting this thesis for formal examination at CQUniversity Australia, I declare that all the research and discussion presented in this thesis is original work performed by the author. No content of this thesis has been submitted or considered either in whole or in part, at any tertiary institute or university for a degree or any other category of award. I also declare that any material presented in this thesis performed by another person or institute has been referenced and listed in the reference Section.

COPYRIGHT STATEMENT

ACKNOWLEDGEMENT OF SUPPORT PROVIDED BY THE AUSTRALIAN GOVERNMENT

This RHD candidature was supported under the Commonwealth Government's Research Training Program/Research Training Scheme. I gratefully acknowledge the financial support provided by the Australian Government.

ACKNOWLEDGEMENT OF FINANCIAL SUPPORT

I gratefully acknowledge the funding received from Australian Research Council (ARC) through the ARC-Discovery Project # DP160102369 which has supported this research.

ACKNOWLEDGEMENT OF OTHER SUPPORT

This research was undertaken with in-kind support of CQU High Performance Computing provided by Jason Bell

ACKNOWLEDGEMENT OF PROFESSIONAL SERVICES

Professional editor, Mr. Adrian Taylor, provided copyediting and proof-reading services, according to the guidelines laid out in the University-endorsed national guidelines, ‘The editing of research theses by professional editors’.

DECLARATION OF CO-AUTHORSHIP AND CO-CONTRIBUTION

Title of Paper	Multi-Cluster Class Balanced Ensemble
Full bibliographic reference	Z. Jan and B. Verma, ”Multi-Cluster Class Balanced Ensemble,” IEEE Transactions on Neural Networks and Learning Systems, 2020.
Status	Accepted on: 6th march
Nature of Candidate’s Contribution	I developed the methodology, ran experiments, compiled the results and wrote the paper. Contribution: 60%
Nature of Co-Authors’ Contributions	The co-author provided guidance in developing the methodology, conducting experiments, analysing results and writing the paper. Contribution: 40%

Title of Paper	Evolutionary Classifier and Cluster Selection Approach for Ensemble Classification
Full bibliographic reference	Z. Jan and B. Verma, "Evolutionary Classifier and Cluster Selection Approach for Ensemble Classification," ACM Transactions on Knowledge Discovery in Data, vol. 14, no. 1, pp. 1-8, 2019.
Status	Published
Nature of Candidate’s Contribution	I developed the methodology, ran experiments, compiled the results and wrote the paper. Contribution: 60%
Nature of Co-Authors’ Contributions	The co-author provided guidance in developing the methodology, conducting experiments, analysing results and writing the paper. Contribution: 40%

Title of Paper	A Novel Diversity Measure and Classifier Selection Approach for Generating Ensemble Classifiers
Full bibliographic reference	Z. Jan and B. Verma, "A Novel Diversity Measure and Classifier Selection Approach for Generating Ensemble Classifiers," IEEE Access, vol. 7, pp. 156360-156373, 2019.
Status	Published
Nature of Candidate's Contribution	I developed the methodology, ran experiments, compiled the results and wrote the paper. Contribution: 60%
Nature of Co-Authors' Contributions	The co-author provided guidance in developing the methodology, conducting experiments, analysing results and writing the paper. Contribution: 40%

Title of Paper	Optimal Clusters Generation for Maximising Ensemble Classifier Performance
Full bibliographic reference	Z. Jan and B. Verma, ”Optimal Clusters Generation for Maximising Ensemble Classifier Performance,” International Joint Conference on Neural Networks, 2020.
Status	Accepted on: 20th march
Nature of Candidate's Contribution	I developed the methodology, ran experiments, compiled the results and wrote the paper. Contribution: 60%
Nature of Co-Authors' Contributions	The co-author provided guidance in developing the methodology, conducting experiments, analysing results and writing the paper. Contribution: 40%

Title of Paper	Ensemble Classifier Generation Using Class-Pure Cluster Balancing
Full bibliographic reference	Z. Jan and B. Verma, “Ensemble Classifier Generation Using Class-Pure Cluster Balancing,” International Conference on Neural Information Processing, 2019, pp. 761-769.
Status	Published
Nature of Candidate's Contribution	I developed the methodology, ran experiments, compiled the results and wrote the paper. Contribution: 60%
Nature of Co-Authors' Contributions	The co-author provided guidance in developing the methodology, conducting experiments, analysing results and writing the paper. Contribution: 40%

Title of Paper	Ensemble Classifier Optimisation by Reducing Input Features and Base Classifiers
Full bibliographic reference	Z. Jan and B. Verma, "Ensemble Classifier Optimisation by Reducing Input Features and Base Classifiers," Proceedings of the Congress on Evolutionary Computation, 2019, pp. 1580-1587.
Status	Published
Nature of Candidate’s Contribution	I developed the methodology, ran experiments, compiled the results and wrote the paper. Contribution: 60%
Nature of Co-Authors’ Contributions	The co-author provided guidance in developing the methodology, conducting experiments, analysing results and writing the paper. Contribution: 40%

Title of Paper	Balanced Learning with Ensemble of Convolutional Neural Networks for Image Classification
Full bibliographic reference	Z. Jan and B. Verma, "Balanced Learning with Ensemble of Convolutional Neural Networks for Image Classification," IEEE Symposium Series on Computational Intelligence, 2019, pp. 1-8.
Status	Published
Nature of Candidate’s Contribution	I developed the methodology, ran experiments, compiled the results and wrote the paper. Contribution: 60%
Nature of Co-Authors’ Contributions	The co-author provided guidance in developing the methodology, conducting experiments, analysing results and writing the paper. Contribution: 40%

Title of Paper	Evolving One-Dimensional Deep Convolutional Neural Network: A Swarm based Approach
Full bibliographic reference	A. Haidar, Z. Jan, and B. Verma, “Evolving One-Dimensional Deep Convolutional Neural Network: A Swarm based Approach,” IEEE Congress on Evolutionary Computation, 2019, pp. 1299-1305.
Status	Published
Nature of Candidate’s Contribution	I helped with the development of the methodology and wrote few Sections of the paper. Contribution: 20%
Nature of Co-Authors’ Contributions	The co-authors implemented the methodology, conducted experiments and compiled results, and wrote the paper. Contribution: 80%

Title of Paper	The Optimized Selection of Base-Classifiers for Ensemble Classification Using a Multi-Objective Genetic Algorithm
Full bibliographic reference	S. Fletcher, B. Verma, Z. Jan, and M. Zhang, "The Optimised Selection of Base-Classifiers for Ensemble Classification Using a Multi-Objective Genetic Algorithm," International Joint Conference on Neural Networks, 2018, pp. 1-8.
Status	Published
Nature of Candidate's Contribution	I helped with the development of the methodology and ran some experiments. Contribution: 20%
Nature of Co-Authors' Contributions	The co-authors implemented the methodology, conducted experiments, compiled results, and wrote the paper. Contribution: 80%

Title of Paper	Optimising Clustering to Promote Data Diversity When Generating an Ensemble Classifier
Full bibliographic reference	Z. Jan, B. Verma, and S. Fletcher, "Optimising Clustering to Promote Data Diversity When Generating an Ensemble Classifier," Proceedings of the Genetic and Evolutionary Computation Conference Companion, 2018, pp. 1402-1409
Status	Published
Nature of Candidate's Contribution	I implemented the methodology, ran experiments, compiled the results and wrote the paper. Contribution: 60%
Nature of Co-Authors' Contributions	The co-authors provided guidance in developing the methodology, conducting experiments, analysing results and writing the paper. Contribution: 40%

LIST OF PUBLICATIONS

Journals

- Z. Jan and B. Verma, "Multi-Cluster Class Balanced Ensemble," IEEE Transactions on Neural Networks and Learning Systems, 2020. (Accepted on 6th March)
- Z. Jan and B. Verma, "A Novel Diversity Measure and Classifier Selection Approach for Generating Ensemble Classifiers," IEEE Access, vol. 7, pp. 156360-156373, 2019.
- Z. Jan and B. Verma, "Evolutionary Classifier and Cluster Selection Approach for Ensemble Classification," ACM Transactions on Knowledge Discovery in Data, vol. 14, no. 1, pp. 1-8, 2019.

Conferences

- Z. Jan and B. Verma, "Optimal Clusters Generation for Maximising Ensemble Classifier Performance," International Joint Conference on Neural Networks, 2020. (Accepted on 20th March)
- Z. Jan and B. Verma, "Ensemble Classifier Generation Using Class-Pure Cluster Balancing," International Conference on Neural Information Processing, pp. 761-769, 2019.
- Z. Jan and B. Verma, "Ensemble Classifier Optimisation by Reducing Input Features and Base Classifiers," IEEE Congress on Evolutionary Computation, pp. 1580-1587, 2019.
- Z. Jan and B. Verma, "Balanced Learning with Ensemble of Convolutional Neural Networks for Image Classification," IEEE Symposium Series on Computational Intelligence, pp. 1-8, 2019.
- A. Haidar, Z. Jan, and B. Verma, "Evolving One-Dimensional Deep Convolutional Neural Network: A Swarm based Approach," IEEE Congress on Evolutionary Computation, pp. 1299-1305, 2019.
- S. Fletcher, B. Verma, Z. Jan and M. Zhang, "The Optimised Selection of Base-Classifiers for Ensemble Classification Using A Multi-Objective Genetic Algorithm," International Joint Conference on Neural Networks, pp. 1-8, 2018.
- Z. Jan, B. Verma and S. Fletcher, "Optimising Clustering to Promote Data Diversity When Generating an Ensemble Classifier," Genetic and Evolutionary Computation Conference, pp. 1402-1409, 2018.

ABSTRACT

Ensemble classifiers are created by combining multiple single classifiers to achieve higher classification accuracy. Ensemble classifiers benefit from the 'perturb and combine' strategy, where an input data is perturbed to generate sub-samples and base classifiers are trained on generated sub-samples. All trained base classifiers are then suitably combined, and an ensemble decision is formed. One common strategy of perturbing input data is through clustering. Data clusters are generated from the input, and base classifiers are trained on generated data clusters. Such ensemble classifiers are also called clustering-based ensemble classifiers as they utilise clustering algorithms to generate a perturbed input training space.

Clustering has been very applicable when it comes to generating ensemble classifiers, however it has certain limitations. One key limitation is that clustering algorithms require the number of data clusters in advance. Most of the existing ensemble approaches use a fixed number of data clusters, that are generated for various datasets, and normally searched through a process of trial and error. Additionally, since clustering works independently of data classes, class imbalances may occur in the data clusters, and data clusters may miss data samples from certain classes. Therefore, not all data clusters are suitable for the training of base classifiers, and redundant or imbalanced data clusters, should be dealt with appropriately. Besides the number of data clusters problem, the choice and type of base classifiers utilised to train on generated data clusters also have significant impact on the ensemble classifier's performance. The use of all base classifiers to generate an ensemble classifier is not an ideal strategy, so an appropriate classifier selection methodology must be adopted to select the subset of base classifiers that can maximise the ensemble classifier's accuracy.

In this thesis several novel ensemble classifier methods have been proposed to mitigate the limitations and improve accuracy of ensemble classifiers. The first ensemble method is based on a novel strategy of incorporating an evolutionary algorithm to dynamically search for the upper bound of clustering. The second ensemble classifier method incorporates an evolutionary algorithm in two phases by optimising the pool of data clusters rather than a single upper bound and optimising the pool of base classifiers. The third ensemble classifier method is based on a hybrid approach that solves the problem of dimensionality and uses reduced dimensions data to generate an optimised ensemble classifier. The fourth ensemble classifier method is based on a novel cluster balancing strategy that solves the problem of class imbalances by balancing data clusters. The fifth ensemble classifier method contains a novel strategy to find the optimal value of clusters for each data class through the incorporation of cluster validation strategies. The sixth ensemble classifier method is based on a novel classifier selection strategy that selects classifiers from the pool based on accuracy and diversity comparisons. The seventh, and final ensemble classifier method, uses a novel pairwise diversity measure to select classifiers from the pool based on increasing accuracy and diversity. The proposed ensemble methods were evaluated on several benchmark datasets. These datasets are used by other researchers and allow a comparative analysis. In most cases an ensemble classifier's accuracy was used as a metric to measure the performance, and in other cases different diversity measures were used. Statistical significance testing was also conducted to further validate the efficacy of the results and p-values were reported.

The results and analysis presented in this thesis show that the proposed ensemble methods not only achieved classification accuracy better than existing state-of-the-art ensemble methods, but also provide a platform for future research. It was found through experimentation that upper bounds of clustering follow a logarithmic relation with the number of data samples each dataset has. Moreover, through extensive experimentation, it was proved that not all base classifiers should be selected to generate the ensemble, and only a subset of base classifiers is required to generate an ensemble classifier that can achieve the highest classification accuracy. Through the incorporation of optimisation, it was also proved that no preference is given to a specific base classifier and the type of base classifier is dependent on the characteristics of the dataset. Silhouette analysis proved to be an effective cluster validation metric to determine the optimal number of data clusters. Finally, balancing data clusters proved to be effective not only in terms of classification accuracy, but also confirmed that each dataset has different spatial characteristics which, when exploited appropriately, can contribute to overall ensemble classifier accuracy.

ACKNOWLEDGEMENTS

This thesis could not have been completed without the help and involvement of many individuals.

First and foremost, I would like to express a deep gratitude towards my supervisor, Professor Brijesh Verma, without whose guidance and support I could not have completed this thesis. He has devoted a lot of his time to help me to develop novel concepts, conduct experimental analysis and write papers for high quality conferences and journals, leading to the completion of this thesis.

I would like to thank the School of Engineering & Technology and the RHD team for their constant support and encouragement in all matters during the pursuit of my PhD. I would also like to gratefully acknowledge the financial support received from CQU and Australian Research Council.

I would like to thank my parents for their constant encouragement and finally, my wife, Mrs. Houzaifa Jan, for bearing with me on the three-year PhD journey and its inevitable struggles. Her motivation and support were crucial, as was the effort she expended in looking after the family all by herself.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ANN	Artificial Neural Networks
ARPSO	Attractive and Repulsive Particle Swarm Optimisation
CBCA	Classification by Cluster Analysis
CCBM	Class and Cluster Balanced Method
CIFAR	Canadian Institute for Advanced Research
CL	Clustering
CPES	Complementary Ensemble Selection method
CSMEM	Classifier Selection by Multiple Elimination Method
CSO	Competitive Swarm Optimiser
DES	Dynamic Ensemble Selection
DF	Double Fault
DGEELM	Diversity Guided Ensemble of Extreme Learning Machines
DM	Disagreement Measure
DT	Decision Trees
EBAGTS	Ensemble-based Artificially Generated Training Samples
ECI	Ensemble Clustering Index
EDSVC	Ensemble Driven Support Vector Clustering
ELM	Extreme Learning Machine
EMU	Expected Margin Utility
EPSMS-BI	Ensemble Particle Swam Model Selection–Best Iteration
FSOM	Feature Selection and Optimisation Method
GMB	Global Mapping Block
HIEL	Hybrid Incremental Ensemble Learning
IC	Inverse Correlation Coefficient
IDAFSEN	Improved Discrete Artificial Fish Swarm Algorithm Ensemble
IK	Interrater K
KNN	k-Nearest Neighbour
LDA	Linear Discriminant Analysis
LDC	Linear Discriminant Classifier

LMB	Local Mapping Block
LSVM	Linear Support Vector Machines
MCS	Multi Classifier Systems
MDM	Misclassification Diversity Method
MNIST	Modified National Institute of Standards and Technology
MOGP	Multi Objective Genetic Programming
MOSEL	Multi-Objective Sparse Ensemble Learning
MPRaF-T	Multi Proximal Random Forest with Tikhonov
MPSVM	Multi Surface Proximal Support Vector Machine
MT	Meta Heuristics
MV	Majority Voting
NB	Naïve Bayes
NCA	Neighbourhood Component Analysis
NMC	Nearest Mean Classifier
NULCOEC	Novel non-Uniform Layered Cluster Oriented Ensemble Classifier
OCCM	Optimized Clusters and Classifiers Method
OCGM	Optimal Class-Pure Cluster Generation Method
OEC-ILC	Optimal Ensemble Classifier – Incremental Layered Clustering
PCA	Principle Component Analysis
PSEMISEL	Progressive Semi Supervised Ensemble Learning
PSO	Particle Swarm Optimisation
QBC	Query by Committee
QDC	Quadratic Discriminant Classifier
QFWEC	Quadratic Form Weighted Ensemble Classifier
QT	Q test
RaF	Random Forest
REC	Recall
RL	Reinforcement Learning
RoF	Rotation Forest
SA	Simulated Annealing
SOI	Sparse Solutions of Interests

SSL	Semi Supervised Learning
STJ48	Standard Classification with Clustering
STLR	Stacking with Logistic Regression
SVM	Support Vector Machines
UBOM	Upper Bounds Optimisation Model
UCI	University of California Irvine
WMV	Weighted Majority Voting

LIST OF SYMBOLS

f_1	Error of the ensemble
f_2	Size of the ensemble
y^o	Vector representing wrongly classified samples
φ^n	Indicator function converting particle positions to 1s and 0s
C	A data cluster
D	Dissimilarity
$DF(D_i, D_j)$	Double fault measure of two dissimilarity of classifiers
$DM(D_i, D_j)$	Disagreement measure of two dissimilarity of classifiers
$I(x, y)$	Indicator function
$J(\mathrm{C}_i, \mathrm{C}_j)$	Similarity of two data clusters Jaccard Index
K	Upper bound of clustering
N	Number of clusters for each class
$P(y'\|x)$	Posterior probability y given x
$Q(D_i, D_j)$	Q test of two dissimilarity of classifiers
R	Number of chances each base classifier has
T	Testing dataset
V	Validation dataset
X	Training dataset
X'	A subset of X consisting of samples from a single class
a	Accuracy of ensemble
$a(i)$	Similarity score of a point i
b	Row vector representing chances of each base classifier
$b(i)$	Dissimilarity score of a point i
bcp	Base classifier pool
c	Cluster centroid
$f(sp)$	Cost/Objective function for finding the best value of K
$f(\xi)$	Cost/Objective function for finding the minimum RMSE and component size of a possible ensemble solution

k	Number of data clusters to generate
l	Loss threshold of each feature
m	Number of features
n	Number of samples
pop	Population vector representing base classifiers as particles
r	Row vector representing each data class
$s(i)$	Silhouette score of a point i
sp	Pool of data clusters
sp'	Pool of balanced/optimised data clusters
w	Weight vector of each feature
x	A feature vector
y	A vector of class labels
y'	A vector of predicted class labels
β	A set of base classifiers
ζ	A base classifier
κ	Number of data classes
ξ	An ensemble solution
ς	Number of base classifiers

Chapter 1: **Introduction**

1.1 Background

Humanity has always relied on groups when it comes to making decisions about new ideas, rules, laws and other important matters. While individual decisions are prone to error they can be mitigated if a group of individuals of differing opinions is formed to take a common decision. An example is the famous television game show ‘Who wants to be a millionaire’, in which a player is given a choice of whether to call an expert to help with a question or ask the crowd. Statistics have shown that asking the crowd to come up with an answer is always better than asking a single expert and is commonly referred to as the “wisdom of the crowd”. The notion was formally explored by the Marquis de Condorcet in 1784 in the context of democratic systems [1]. The idea was further explored by Laplace in 1818, and it was suggested that when decisions of different probabilistic methods are suitably combined, the combined system can outperform the individual. Thus the concept of an ‘ensemble’ (mixture of experts) was born and was officially introduced in the statistical mechanics by Gibbs in 1878 [2].

Ensemble classifier is a machine learning classification strategy that is utilised for improving the classification accuracy of single classifier models by suitably combining the class label estimates of multiple base classifiers. An individual classifier is deemed to be accurate if it performs better than random guessing, and diverse if the error it makes is uncorrelated to the errors of other classifiers. Combinations of diverse and accurate classifiers have shown improved performance compared to a system where only accurate classifiers are selected [3-5]. The process of ensemble classifier learning can be categorised into three stages (**Generation −> Selection −> Combination**).

First is the generation stage, where given input data is “perturbed” by generating a random subspace from it. This is done to control the bias and variance of base classifiers that are trained on a given input. If all base classifiers are trained on the same input samples, their classification capabilities are correlated even though their internal structures may differ. This concept is also known as the ‘bias variance co-variance decomposition’ [6] theory. Base classifiers are trained on all generated sub-samples to generate the base classifier pool. A mixture of base classifiers such as Artificial Neural Networks (ANN), Support Vector Machines (SVM), *k*-Nearest Neighbour (KNN) etc. are trained. If same base classifiers are trained on various generate sub-sample the ensemble is considered a homogeneous ensemble. If, however, different base classifiers are trained then the ensemble is considered a heterogeneous ensemble.

Secondly, base classifiers from the pool are selected by using a suitable base classifier selection methodology that can maximise ensemble classifier accuracy. Lastly, all suited base classifiers are combined through a suitable decision fusion methodology, typically on a single dataset, and combined using a suitable fusion methodology such as majority voting [7] to generate the class predictions of the ensemble. The three stages of ensemble classifier training are illustrated in Figure 1.

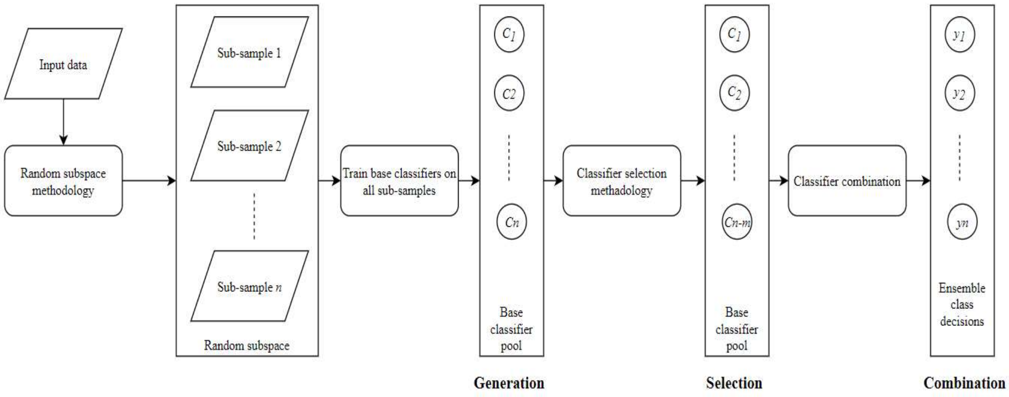


Figure 1: Ensemble classifier method.

Two pioneering works in ensemble classifiers were presented by Zhou [3] and Schapire [8]. Zhou presented the idea of ensemble classifiers by empirically proving that classification accuracy can be increased if, in a neural network ensemble, a set of homogeneous neural network classifiers is suitably combined. Schapire, on the other hand, suggested that retraining a weak learner on data patterns where the learner performed poorly can boost the performance of the learner. Ensemble classifiers perform better than single classifier systems because they are not biased by the decision of a single learner, rather class label estimates of different learners are combined to form a unanimous decision which has lower generalisation error. According to the *'no free lunch'* theory [9] it is practically difficult to find a single classifier that performs well on different datasets but ensemble classifiers overcome this by having multiple learners form a common consensus.

Ensemble classifiers have been applied to diverse real word tasks including object detection, credit analysis, weather forecasting, medical data analysis, *etc.* and they have been found useful. Two famous real-world ensemble classifier applications are found in KDD-Cup and Netflix Prize. KDD-Cup [10] is a famous data mining competition and is held every year since 1997. It covers a variety of real-world tasks in many fields including molecular biology, customer relationship management, computer networks, and education data mining. Since 2009 many first place and second place winners have used ensemble methods. Another example is the Netflix Prize, [11] an online DVD rental service. Netflix uses a learning algorithm which gives recommendations to viewers based on their past movie preferences and seeks participants who can improve Netflix's own algorithm by 10% accuracy. A grand prize of $1M was awarded to the team *BellKor's Pragmatic Chaos* in September 2009.

1.2 Problems and motivation

The main aim of this thesis is to develop a novel ensemble classifier framework for generating and optimising ensemble classifiers. The idea is to create many clusters from input data, train base classifiers, fuse them and finally generate an optimal ensemble classifier. Clustering-based [12] ensemble classifier methods have been the focus of extensive research predominantly because of their ability to achieve good performances on real-world data.

As effective as it may be, a key parameter to clustering-based ensemble classifiers is the number of clusters that should be generated. This parameter is required a-priori and should be passed as an argument to a clustering-based ensemble classifier. This is a key limitation of clustering-based ensembles, because knowing the optimal number of data clusters beforehand is not easy. Moreover, one value that may work for one data may not work well for another. Therefore, an ensemble classifier method that is geared towards a specific dataset may not be very practical for real-world applications.

Another key component of an ensemble classifier is the choice of base classifiers. Initially, ensemble classifier methods such as Bagging, Random Forest, *etc.* used decision trees as their choice of base classifier. These methods trained a multitude of decision trees on either feature sub-samples or input sample sub-samples. However, recent ensemble methods use diverse base classifiers such as Artificial Neural Networks (ANNs), Support Vector Machines (SVMs), Decision Trees (DTs), K- Nearest Neighbours (KNNs), *etc.* This is an effective strategy to generate an ensemble as the more diverse base classifiers are chosen the more accurate the ensemble becomes. [13]. This leads to a new domain of the choice of base classifier that can or should maximise ensemble classifier accuracy.

This thesis explores the potential effects of dynamically searching for an optimal value of clustering to generate a random subspace of data clusters and investigates the effect of clustering in relation to different datasets. It also examines the effect of dynamically searching for the best subset of base classifiers from the base classifier pool, and whether there is a preference for any type of base classifiers.

1.3 Research questions

This thesis aims to answer the following research questions:

- How can evolutionary algorithms be utilised to optimise an ensemble classifier?
- What is the optimal number of clusters to generate from training data in order to train diverse classifiers?
- Which set of classifiers or base classifiers act as the optimal set for the ensemble pool?
- How well can the proposed methods perform in comparison with the existing state-of-the-art methods used for ensemble classifiers?

1.4 Original contributions

The original contributions of this thesis are as follows:

- An ensemble classifier framework is proposed that utilises clustering to create a random subspace of data clusters, trains base classifiers on all created data clusters and fuses them to generate an optimal ensemble.
- An ensemble method, namely Upper Bounds Optimisation Method, is proposed that trains an ensemble classifier by optimising the upper bound of clustering through the incorporation of an evolutionary algorithm.
- An ensemble method, namely Optimised Clusters and Classifiers Method, is proposed that optimises not only the pool of generated data clusters, but also the pool of trained base classifiers to generate an optimised ensemble classifier that can not only achieve high classification accuracy, but also has a lower component size.
- An ensemble method, namely Feature Selection and Optimisation Method, is proposed that identifies significant input features and solves the issue of the curse of dimensionality, especially in small datasets with large input features, to generate an ensemble.
- An ensemble method, namely Class and Cluster Balancing Method, is proposed that solves the class imbalance problem by generating class-pure data clusters and balancing each data cluster using the proposed clustering balancing methodology. The proposed data augmentation method works effectively on benchmark classification datasets and image datasets.
- An ensemble method, namely Optimal Class-Pure Cluster Generation Method, is proposed that computes the optimal value of data clusters for each data class in the input data.
- An ensemble method, namely Classifier Selection by Multiple Elimination, is proposed that uses the proposed classifier selection methodology to select classifiers from the pool based on accuracy and diversity comparisons.

- A diversity measure, namely Misclassification Diversity Method, is proposed that generates ensembles by selecting classifiers using the proposed diversity measure.

1.5 Thesis structure

Chapter 1 provides an overview of ensemble classifiers and lists the various research questions and original contributions.

Chapter 2 briefly discusses the recent state-of-the-art ensemble classifier methods that exist in current literature.

Chapter 3 discusses the proposed ensemble classifier framework that is utilised throughout this thesis to develop further ensemble methods. It also discusses the datasets used in experiments and various evaluation metrics that are utilized to gauge the performance of various methods.

Chapter 4 discusses the proposed ensemble classifier method that extends the original ensemble framework by utilising an evolutionary algorithm to optimise the process of searching for the optimum value of clustering to generate a random subspace.

Chapter 5 further extends the proposed method by incorporating an evolutionary algorithm to not only optimise the pool of data clusters but also the pool of trained base classifiers. The proposed method eliminates the need for knowing the upper bound of clustering a-priori as it optimises the pool of data clusters whatever value of clustering is chosen.

Chapter 6 proposes a novel cluster and class balancing method for generating an ensemble classifier. The proposed method mitigates the issue of class imbalances that not only exist in datasets but also in data clusters. Since clustering works independently of the data classes, the generated data clusters are also imbalanced.

Chapter 7 proposes an ensemble method that utilises a cluster validation analysis to generate an optimal number of data clusters for each data class and generates an ensemble.

Chapter 8 proposes a classifier selection methodology to generate an ensemble classifier by allowing each classifier from the pool of base classifiers to participate in multiple rounds of selection.

Chapter 9 proposes a pairwise diversity measure that is utilised by a classifier selection methodology to generate an ensemble.

Chapter 10 summarizes the contributions that are made in this thesis, answers the research questions, and provides future directions.

Chapter 2: **Literature Review**

This section reviews state-of-the-art ensemble classifier methodologies. A lot of work has been done to develop various methodologies for ensemble classifiers [3, 4, 14]. Previous research indicated that ensemble classifiers can outperform a single classifier system, primarily because ensemble classifiers do not depend on the performance of a single classification function. The bulk of this section critically investigates various ensemble classifier methodologies and identifies where a contribution can be made.

2.1 Legacy ensemble classifier methods

Ensemble classifiers benefit from "perturb and combine" strategy, where first a subsampling strategy is utilised to generate partitions of the input data which are uncorrelated. On the generated input partitions heterogeneous or homogeneous classifiers are trained to generate the pool. Very briefly a pool of over produced trained classifiers is generated which are uncorrelated to each other and various selection methodologies is employed to select the best subset that can maximise the generalization ability of the ensemble. Therefore, ensemble classifier methods can be placed in three categories [14]. Some methods use different subsets of training data with a single base classifier, others use different training parameters on same base classifier, and some methods use different base classifiers. Two famous ensemble classifier methods, known as 'bagging' [15], and 'boosting,' [16] belong to the first category of ensembles. Bagging works by training homogeneous classifiers on random samples of training data with replacements (stratified sampling). Boosting, in overall structure, is like Bagging. The difference is that Boosting uses the samples of training data which have not been used by the classifier to increase its accuracy on the samples which it already has trained on. It therefore boosts the performance of that classifier [15, 16].

'Random Forest'[17] belongs to the second category of ensembles and it trains base classifiers, decision trees in this case, using different training parameters. Random Forest selects a random vector of features from the training data and trains trees on that vector. Random Forest has been a very popular and successful ensemble classifier and performs very well compared to Boosting and Bagging with noisy data [17].

Wozniak *et al.* [18] conducted a survey on Multi Classifier Systems (MCS), also known as ensemble classifiers, by gathering information from three well known academic sites where the keyword MCS has been searched for. They showed that since 1990 there has been an exponential growth in MCSs, and a significant research component was devoted to MCS. They suggested that MCSs have three main benefits as follows: filter out hypothesis that is incorrect due to small training data; combining classifiers trained on different segments of data can overcome the local optima problem, and; a single expert classifier might be impossible to train but training multiple classifiers can expand the representation of the problem space. They summarised that designing MCSs can be categorized into three components;

1) classifier generation methodology which deals with the methods of training of multiple base classifiers on a single input data;
2) classifier selection methodology which deals with the methods selecting classifier from the pool of generated trained classifiers;
3) classifier fusion methodology which deals with the methods of combining multiple classifiers to generate the ensemble output.

They further discussed how different MCS design methods exist in different problem streams and how one can out-perform another for a given problem set.

2.2 Ensemble classifier component size

A key aspect when it comes to creating ensemble classifiers is the size of ensemble itself. Adding more classifiers in an ensemble classifier does increase overall accuracy, however, a recent study has proved that having more or less than an optimal number of classifiers in an ensemble decreases the ensemble accuracy. This is called the '*law of diminishing returns'*. Bonab [19] theoretically provided a framework to show that having as many classifiers as there are class labels in the dataset gives an ensemble classifier that can achieve the highest accuracy.

Although it has been shown that adding to the number of classifiers can increase accuracy of an ensemble classifier, there is still need of a conclusive work which can tell *a priori* the optimum size of an ensemble classifier that should be maintained to achieve highest accuracy without imposing any unnecessary computational cost. Oshiro *et al.* [20] proved in their research, entitled '*How many trees in a random forest,'* that adding trees does not necessarily increase the accuracy of the ensemble, rather it adds more computational complexity to the ensemble classifier. Lattinne *et al.* [21] used McNemar significant testing to prove that adding more classifiers (tree) in an ensemble classifier does not increase accuracy, and suggested that one should incrementally add classifiers and compare the performance of the current ensemble classifier with the one before. If test results are not significant then we have reached an optimal number of classifiers and adding more will not increase the performance of the ensemble classifier.

Lustosa *et al.* [22] comparatively analysed various dynamic ensemble classifier selection methodologies. In order to test the performance of ensemble classifiers they used majority voting as a fusion methodology and four diversity measures namely $q-$statistics, double fault, bad and good diversities. For ensemble selection they used *k*-NN and *k*-means algorithms and 14 datasets were used from the UCI machine learning repository. After experimentation it was concluded that many classifiers produced accurate results typically, when an initial set of 30 classifiers was used. Their work also concluded that using higher numbers of classifiers does not significantly influence performance unless only two classifiers are used initially. It was shown that diversity influences the accuracy except for good diversity and Q-statistics. Lastly, for ensemble selection there was no significant impact and either selection method can be used.

Lysiak *et al.* [23] proposed a novel method of Dynamic Ensemble Selection (DES) system. It was suggested that classifier accuracy and diversity can be considered an optimisation problem. The proposed method uses classifier accuracy as an objective function for an optimisation problem and diversity as a constraint, thus the method of DES-CD. The authors suggested using a Simulated Annealing (SA) algorithm to find the best ensemble classifier instead of exhaustively searching for the best set of classifiers which are both accurate and diverse. Moreover, SA algorithm is faster than Tabu search and Genetic algorithms allowing the model to converge at a much faster rate. They conducted experimentation on six benchmark datasets taken from UCI repository and one benchmark dataset taken from Statlab. They concluded that the hybrid method DES-CD performed very well compared to other methods not only in terms of classification accuracy but also ensemble component size.

2.3 Clustering-based ensemble classifier methods

Besides researching an optimal number of classifiers to generate an ensemble classifier, various classifier fusion methods have been developed over the years. Some methods use diversity as a measure when combining classifiers, others use accuracy; although there is a debate between accuracy and diversity, accuracy is given precedence over diversity in most cases [24]. One way of incorporating diversity whilst maintaining accuracy is to generate a diverse input space on which base classifiers are trained. Such an input space is often referred to as the random subspace [25]. A random subspace is essentially generated from the input data and it contains smaller subsets of the input data with repeating and unique records. The main idea is to control the bias and variance of base classifiers by training them on different sub-samples of the input data. The benefits here are incorporation of diversity in two folds: firstly, since classifiers are trained on different sub-samples, each is different as it has learned on a different sub-sample; secondly, different types of classifiers are trained on different sub-samples which further increases diversity incorporation. The novelty here is that if a large enough 'perturbed' input space is generated, then a classifier selection methodology can be incorporated to select the subset able to maximise the accuracy of the ensemble. Besides Bagging, another common random-subspace methodology is clustering, and ensemble classifier approaches that utilise it are called clustering-based ensemble classifiers.

As such, Asaf [26] proposed a Novel non-Uniform Layered Cluster Oriented Ensemble Classifier (NULCOEC) method for generating an ensemble classifier. It has been proved empirically that achieving high diversity with accuracy together can be classified as a multi objective optimisation problem and using evolutionary algorithms such as genetic algorithms can be beneficial. In experiments the author used genetic algorithms to optimise the number of layers required to generate an ensemble classifier with highest accuracy and diversity.

Rahman [27] achieved diversity in ensemble classifiers by clustering datasets using *k*-means algorithm into atomic and non-atomic data clusters. An atomic cluster is a cluster with only one class label and a non-atomic cluster has multiple class labels. Every non-atomic cluster was fed into a Neural Network with two hidden layers to transform it into an atomic cluster. This process was repeated until every non-atomic cluster was converted into an atomic cluster. When all the clusters are atomic, decisions can be formed. Asafuddoula *et al.* [28] proposed a novel method of incremental ensemble learning process. In the proposed method, the input data was first partitioned into several data clusters and on each cluster a set of base classifiers was trained. Classifier accuracy and diversity was calculated for all classifiers, and, with incremental layered method, each new classifier with higher accuracy was added to the ensemble. If classifier accuracy remained unchanged, but there was an increase in diversity, then that classifier was added to the ensemble; if neither accuracy nor diversity increased then the classifier was discarded. Also, the process over each iteration increased the number of data clusters generated from the dataset. Similarly, Fletcher [29] proposed an extension to the idea and suggested, since data clusters have repeating and unique records, a measure must be introduced which can classify a data cluster as a repeating data cluster or not. The idea behind the work is that if base classifiers are trained on repeating data clusters with majority of samples in common, such base classifiers will be biased. They introduced a similarity measure known as the Jaccard Index [30], and any cluster that had more than 90% similarity was discarded.

In another research Huang *et al.* [31] suggested that most of the clustering-based ensemble classifier approaches have one limitation in common, that is they all consider clustering algorithms equally regardless of their performance and reliability. This problem is exacerbated when there is no access data or features available to validate the results of the clustering algorithm. Therefore, they proposed a clustering-based ensemble approach based on uncertainty estimation and local weighting strategy. To evaluate the reliability of a data cluster in the proposed ensemble framework entropy is used, where each cluster is given an entropy score and an ensemble clustering index (ECI) is generated. Any cluster below the ECI threshold is discarded from the pool.

2.4 Classifier selection-based ensemble classifier methods

Bagheri *et al.* [32] suggested the use of Dempster Shafer theory of fusing diverse classifiers to generate an ensemble classifier. The authors compared three fusion methods. Firstly, different feature sets were used and combined in a higher dimension to be used by a single classifier. Secondly, different random samples were used to generate diverse classifiers which were fused together. Thirdly, random subspace sampling was used to create partitions of features. They conducted experiments on UCF101[33] datasets and used SVM for classification. Experimentation showed that ensemble classifiers based on random sub sampling of feature sets to generate diverse classifiers performed very well for image classification.

Zhang [34] proposed a novel method of using oblique decision tree ensembles using geometric decision trees as base learners. Oblique decision trees were used with Random Forest and Rotation Forest. The hyperplane was optimized using existing Multi Surface Proximal Support Vector Machine (MPSVM). Two regularization strategies, Tikhonov and axis parallel regularisation, were used to optimise the hyperplane of MPSVM. Additionally, Bhattacharyya distance was used to transform a multi class problem into a binary classification problem. Experiments were conducted on 40 benchmark datasets from the UCI repository and four bio informatics datasets, including a face recognition dataset from Yale. It was concluded that MPSVM with Random Rotation Forest using Tikhonov regularisation performed well compared to other methods.

Bhattacharjee *et al.* [35] proposed an extension to the existing Expected Margin Utility (EMU) algorithm which was used to generate a knee of Pareto Optimal Front (POF). The proposed method used EMU recursively called as EMU^r, to generate unique sparse solutions of interest (SOI) at the end. The SOI generated was classified into three classes Peripheral$_E$, Peripheral, and Internal classes, and depending on the decision-maker's preference a specific solution was chosen. The proposed method was tested with multiple objective benchmark datasets DO2DK, DEB2DK and DEB3DK [36] . The test was conducted on 200 non-dominated solutions with four POF knee regions for DO2DK; DEB2DK and EMU^r was able to predict 23 SOI out of which nine were in internal classes, thus offering a higher reduction rate than EMU which offered 24 sparse solutions. Similarly, EMU^r outperformed DEB3DK in predicting a higher number of knee regions and offered higher reduction rates in proposed solutions as well. EMU^r was also tested with real world applications using Radar Wave form design, and General aviation aircraft problem. The proposed method was able to utilise the inherent property of expected marginal utility and was able to deliver diverse solutions of interest.

Qi *et al.* [37] proposed a novel method of combining deep learning and support vector machine. The proposed method takes advantage of auto encoder technique and integrates ensemble classifiers. Ensemble classifier is used to pre-train the data in deep learning for SVM. ExAdaboost is used to tune the weights of a layer before feeding it to the next layer. In this manner the proposed method tunes the parameters in the feed forward step and no back propagation is required. The outputs of SVM from each layer become the input of the SVM of the next layer; this becomes the crucial step of the deep SVM structure. They conducted experiments on eight datasets from the UCI repository using 10-fold cross validation. According to experimentation the proposed method outperformed SVM, stacked SVM, MLSVM, MKMS, and DNN in six of eight datasets from UCI repository, and lost to DNN in German, and w1a datasets by a margin of 0.44, and 0.06 respectively.

Chang [38] proposed a novel Complementary Ensemble Selection method (CPES). The proposed method selects a classifier based on diversity by overcoming the weaknesses of previous classifiers in the pool like Boosting. The method was compared with existing state of the art ensemble classifier methods namely Ensemble of Ensembles (EE) using five popular data sets LETTER.p1, LETTER.p2, COVTYPE.p1, COTTYPE.p2, and COVTYPE.p3 from the UCI repository. Experimentation proved that both EE and CPES significantly outperformed bagging, however CPES was not able to outperform EE. Although CPES and EE yielded a similar accuracy, CPES was able to do so with a significantly smaller ensemble size compared to EE. CPES is computationally less expensive than EE whilst maintaining the same accuracy.

Huang *et al.* [39] proposed a novel method known as Ensemble Driven Support Vector Clustering (EDSVC). The authors suggested that support vector clustering, though being very versatile, lacks the ability to handle kernel parameters and trade off parameters. Many have tried to solve this issue using many techniques, yet there is still a need for a technique which can automatically identify these two parameters in an unsupervised manner. In order to solve this issue, the authors suggested a novel method of EDSVC which successfully implements SVC in an unsupervised manner and automatically finds these parameters. Authors tested EDSVC on six real world data sets from UCI repository and compared EDSVC with four state-of-the-art ensemble classifier clustering methods namely COMUSA, DICLENS, COMUSAACL and COMUSAACL-DEW. They also compared EDSVC with PSVC method, and, according to experiments, EDSVC outperformed other methods significantly in terms of NMI scores and proved to be more effective.

Kilic [40] proposed a gating system to select the best set of classifiers from the pool in order to form an ensemble. It was suggested that focusing only on the expertise of the classifiers in an ensemble classifier is better than fusing all classifiers. Therefore, the gating system will select only those classifiers which have higher posterior probability over the validation set. This way the ensemble generated will not only be less biased but will also be pruned of all those classifiers that negatively affect the overall accuracy of the ensemble. Authors conducted experimentation on 20 datasets from UCI and Delve[41] repositories; used C45 decision trees, Gaussian Classifiers, *k*-NN, linear discriminant tree, linear logistic classifier, multi-layer perceptron, multivariate discriminant tree, and support vector machine with radial, linear and polynomial kernel as their base classifiers. According to experimentation it was concluded that the proposed method not only performed better than other referee-based methods, but also formed an ensemble which was significantly smaller compared to others.

Mao *et al.* [42] proposed a novel method of using weighted classifier ensembles based on quadratic forms, namely QFWEC (Quadratic Form Weighted Ensemble Classifier). The authors generated an ensemble classifier with global minimised error rate by selecting classifiers with a weighted quadratic form. They introduced a regularisation term in order to select the most optimised set of classifiers which will be included in the pool. They introduced a weight vector and found a solution by using an optimisation algorithm. The new method of combining classifiers with optimal weight vector is acquired by maximising three quadratic terms. They conducted experimentation on datasets from the UCI repository, PolSAR [43] image and an artificial dataset. They proved experimentally that combining individual classifiers can achieve higher classification accuracy than searching for the best classifier. The authors employed decision tree C4.5 with back pruning as base classifier for UCI and artificial datasets and SVM classifier as a base classifier for PolSAR image dataset. According to experimentation they concluded that QFWEC2 and QFWEC3 methods outperformed other ensemble methods including Bagging and Boosting.

Gu [44] proposed an extension to tri training framework with the primary purpose of enhancing the performance of semi supervised learning (SSL). It was suggested that diversity is better achieved if heterogeneous classifiers are used, and this also benefits performance of ensemble. It was also suggested that as Linear Discriminant Analysis (LDA) and Linear Support Vector Machines (LSVM) both use linear hyperplanes, the 'views' they create are not independent of each other. A better method will be to use LDA with *k*-Nearest Neighbour (*k*-NN), as fundamentally the two classifiers are not related, and on unlabelled data it is better to apply a probabilistic method to predict the label before training a classifier on the data. The proposed method used feature manipulation with different learning methods to pair up base classifiers. In order to test the proposed method, the author conducted experimentation on 12 datasets taken from UCI repository. For experimentation purposes the author set 20% of dataset as labelled and rest 80% to be unlabelled. For base classifiers Decision Trees, Naïve Bayes, J4.8 decision trees, and *k*-NN with k=5, were used and three-fold cross validation technique was employed to split data into training and testing sets. Principle Component Analysis (PCA), and Competitive Swarm Optimiser (CSO) were utilised to employ feature manipulation with different techniques. After experimentation it was concluded that besides J48 the proposed method outperformed other Meta Heuristics (MT) based methods and the feature selection methods such as CSO and PCA increased the performance of MT based methods.

Kim *et al.* [45] proposed a novel method of using multiple feature extractors (FE) and multiple classifiers (MC); a hierarchical ensemble creation framework that integrates Bayesian network modelling and reinforcement machine learning. Authors suggested that exhaustively searching for a best feature extractor and classifier pair can be computationally expensive, therefore the problem can be divided into a hierarchy. The first stage is Local Mapping Block (LMB) where a best set of only two classifiers and two FEs are selected and the second is Global Mapping Block (GMB) where LMBs are selected and combined. In order to search for the best LMB, first all possible LMBs should be generated and to achieve this, the problem space can be divided into a graph problem. Euler trail can be used to generate all possible

LMBs which can then be optimised to find GMB. By converting the problem space into a graph problem, the complexity is reduced exponentially. Finally, to fuse classifiers, authors used Weighted Majority Voting (WMV) with Reinforcement Learning (RL). Authors conducted experimentation on seven datasets including pedestrian detection and handwritten recognition numerals from the UCI repository. They compared GMB and LMB methods with existing popular ensemble schemes such as AdaBoost, and basic FE classifier pairs. After thorough experimentation and comparison, authors concluded that the proposed hierarchical method of creating multi features and multi classifier ensembles was not only more accurate than other methods but also imposed less computational overhead.

Yin *et al.* [46] proposed a novel ensemble generation method utilising both sparsity and diversity of base classifiers. The authors suggested that base classifiers which are both sparse and diverse can be categorised as a mathematical optimisation problem and a genetic algorithm can be applied to select a classifier that is both sparse and diverse. They conducted experimentation on six benchmark datasets from the UCI repository and Pascal [47] web spam data base. They concluded empirically that the proposed method of selecting sparse and diverse (S&D) classifiers to create ensemble not only performs well in terms of accuracy, but it also selects a lesser number of classifiers in the ensemble. This means that the proposed method not only performs well but is also efficient. They proposed a mathematical framework for calculating sparsity and diversity which can be implemented by any optimisation procedure which is embedded with one diversity and semantic loss incorporating l_1-norm regularisation.

Abellan [48] empirically compared different ensemble methods performance generated using heterogeneous base classifiers using credit scoring datasets. The base classifiers used in the study were logistic regression, multilayer perception, support vector machines, C4.5 decision tree, and creedal decision trees. They used six credit scoring datasets of which two were taken from the UCI repository, namely Australian and German datasets; Iranian dataset from a small bank in Iran; Polish dataset from a company bankruptcy forecast, and the UCSD dataset from the University of California San Diego. The ensemble classifiers used were AdaBoost, Bagging, Random Subspace, DECORATE, and Rotation Forest; The Friedman test was conducted to rank each ensemble. After thorough experimentation and significance testing, it was concluded that, generally, the best base classifier overall in this scenario is creedal decision trees-based ensembles such as Rotation Forest.

Krawczyk [49] proposed a novel method of using online Query by Committee (QBC) of ensembles to actively learn on live data streams with concept drifts. It was suggested that incorporating randomised variable uncertainty strategy to adaptively learn for new instances and pattern in the presence of concept drift can be effective when generating ensembles. Originally the QBC model was intended for static data, but authors suggested that combining QBC with Bagging ensemble methods is a good way of using QBC with live data streams. An ensemble was used for selecting the most valuable instances from a drifting data stream. The author conducted experimentation on four benchmark datasets and concluded that the proposed method was better able to allocate available budget and obtained more labels from drifting data.

Santucci *et al.* [50] proposed a novel method of using randomisation of parameters of base classifiers to generate a diverse ensemble classifier. They argued that incorporating randomisation through training data can be computationally expensive; when training dataset is small, incorporating randomisation can become an exhaustive task. They suggested that distributing class parameters using multivariate Gaussian distribution and training classifiers on those set of parameters can incorporate randomness. They used three well-known classifiers, namely Nearest Mean Classifier (NMC), Linear Discriminant Classifiers (LDC), and Quadratic Discriminant Classifier (QDC), and conducted experimentation on 27 datasets of which two were generated artificially. These were Correlated Gaussian, and Uncorrelated Gaussian datasets. The remaining datasets were taken from the UCI repository. They concluded that randomisation of parameters can achieve accurate results for smaller datasets as well and proved to be a promotable area of research. They suggested that a more robust statistical model is needed in order to further investigate and utilise the proposed method to incorporate randomisation using parameters.

Bock [51] evaluated the performance of rotation based ensemble classifiers for the prediction of customer defection and proved experimentally that, in rotation-based ensemble classifiers, it is always feasible to perform dimensionality reduction using techniques such as PCA and ICA. Rotating instances and/or features of datasets for different weak learners in the pool of ensemble can also improve accuracy and shows promising results. It was also proved empirically, that although in terms of accuracy RotBoost, and Rotation Forest with ICA and PCA outperforms the same without, ICA based Rotation Forests are proved to be better in terms of decision making.

2.5 Evolutionary algorithm-based ensemble classifier methods

Kadkhodaei [52] proposed a method of adding a layer of entropy between meta classifiers in stack generalization. It was suggested that using diversity as a measure to select the best set of base classifiers would have a lesser impact on computational resources. Oracle output and entropy measure were two inputs for optimisation, and a genetic evolutionary algorithm was employed to find the best set of classifiers. Experiments on four benchmark datasets from the UCI repository were conducted. After experiments it was concluded that, although the proposed method did not have a significant impact on accuracy, the number of classifiers selected for the ensemble were significantly fewer than EBNA and this could prove useful for scenarios where there are real time datasets.

Han *et al.* [53] proposed a novel method of using Attractive and Repulsive Particle Swarm Optimisation (ARPSO) to select base classifiers from the initial pool of classifiers in Extreme Learning Machine (ELM) by considering both classification accuracy and diversity. Their method was a Diversity Guided Ensemble of ELMS built by ARPSO and is called DGEELMBARPSO. The proposed method improves generalisation performance by considering diversity of ensemble, but also adaptively selects members for ELMS. The proposed method was compared with existing ensemble methods based on PSO namely E-ELM, EOS-ELM, E-PSOELM, and ARPSOELM. Experiments were conducted on uniformly random distributed intervals on training sets and test sets. Additionally, uniform noise was also distributed in all the training samples, while test sets remained noise free. DGEELMBARPSO was tested on five benchmark classification problems from the UCI repository. From the results it was concluded that the proposed method could build more accurate ensembles that were compact and generalised and had higher diversity. DGEELMBARPSO with weight voting achieved the highest test accuracies on various datasets.

Escalante *et al.* [54] proposed a novel method of using Particle Swarm Optimisation to generate an ensemble classifier. They modified the original method of PSMS Particle Swarm Model Selection process and proposed a method called Ensemble Particle Swam Model Selection–Best Iteration (EPSMS–BI). They conducted experiments on well-known classification datasets from the UCI repository. In their experiments they used 10-fold cross validation to create training and testing datasets. After conducting experiments, it was concluded that EPSMS–BI achieved higher classification accuracy then both single classifier systems and PSMS–BEST, EPSMS–SE, and EPSMS–BS ensemble classifier methods.

Yang *et al.* [55] proposed a novel method of Clustering Ensembles using Particle Swarm Optimisation CL-POS. In the proposed method they computed the weight of each cluster of ensembles using Particle Swarm Optimisation (PSO). In PSO each cluster becomes a particle in k dimensions, which is then given a relative weight using classical PSO. They conducted experimentation on nine different datasets from the UCI laboratory and compared their algorithm with well-known ensemble clustering techniques such as NMC EM, NMSC EM, LDC EM, QDC EM, and PARZENC EM. According to the results the proposed method performed better and showed promise for future research.

Bhowan *et al.* [56] suggested a genetic programming multi-objective optimisation method to evolve a diverse ensemble classifier. The objective of the proposed method was to develop a multi-objective genetic programming framework for classification of data with unbalance majority and minority classes as learning objective. Additionally, a comparison was made between SPEA2 and NSGA2 pareto-based fitness strategies. Multi Objective Genetic Programming (MOGP) was used to evolve accurate and diverse ensembles. Experiments were conducted on six datasets from the Machine Learning UCI repository. The datasets were chosen carefully so that they have the problem of unbalanced classes. According to the results it was proved that the MOGP method evolved accurate and diverse sets of GP classifiers having higher accuracies for both majority and minority classes. Also, the generated ensemble outperformed canonical GP, NB and SVM classifiers particularly on datasets that had imbalance classes.

Gu [57] suggested that generating diverse and accurate ensemble classifiers can be categorised as a multi-objective optimisation problem. Increasing diversity will have an impact on accuracy and vice versa. Therefore, an optimum need to be achieved between the two. In order to solve this problem a famous multi-objective evolutionary algorithm NSGA-II is used. NSGA-II is famous for solving two objective problems and is used here to find an optimum between diversity and accuracy. In order to test the given method, sample datasets from the UCI machine learning laboratory were taken. Although the proposed method is model independent, for testing Linear Discriminant Hyperplane Support Vector Machine (LVSM) was used to generate an ensemble classifier. According to experimentation 31 of 36 ensembles improved their classification accuracy and, in some datasets, an increase of 30% accuracy was achieved compared to a single classifier. In the current set-up the author also suggested that comparing diversity in the input feature space, while using subsets of training datasets to achieve diverse ensemble classifiers achieved higher accuracy.

Ribeiro *et al.* [58] proposed a multi-objective optimisation design framework for ensemble generation. The authors suggested that any evolutionary ensemble generation methodology could be divided in two parts: firstly, member generation where a set of classifiers is generated; secondly, member selection where the best subset of classifiers that can maximise the generalisation ability of the ensemble is selected. In the proposed ensemble classifier framework authors used true positive rate, true negative rate, F1 score, and classifier complexity as objectives of the optimisation process to generate the pareto front. They then selected the best subset that can minimise the mentioned objective and generate an ensemble classifier. They tested the proposed ensemble framework on GECCO's SPOTSeven dataset challenge 2017 [59].

Zhao *et al.* [60] proposed that the sparseness of an ensemble can be treated as an objective of an evolutionary multi-objective ensemble learning (MOSEL). They utilised detection error trade-off (DET) as a measure to gauge the ensemble performance. They defined sparsity ratio, false positive rate, and false negative rate as the three minimisation objectives for the optimisation process. Several evolutionary algorithms are utilised, and their performance is compared to determine the best evolutionary process that can generate an ensemble with best performance. Similarly Zhang et al. [61], proposed an evolutionary feature subspace generation methodology for ensemble classification. They utilised a binary version of PSO for evolutionary multi-task feature section. They generated multiple feature subspace by training a set of classifiers (SVM, DT, and KNN) on feature subsets which can maximise their generalisation performance. All classifiers that are trained on feature subspaces are then combined through majority voting scheme to generate an optimised ensemble classifier. They conducted experiments on UCI benchmark datasets and compared the performance of their approach with existing works. In another work Yu *et al.* [62] proposed a progressive semi-supervised framework for learning of multiple classifiers (ensemble). Authors suggested that, to the best of their knowledge, no work exists that utilises evolutionary algorithms to augment the input data and, therefore, propose an evolutionary sample selection process to generate an ensemble. They suggested that proposed progressive semi-supervised ensemble learning (PSEMISEL) works very well with small datasets with many features, as it can identify the significant features and only use them for training of base classifiers.

Yen [63] proposed an ensemble to measure the performance of various Multi Objective Evolutionary Algorithms (MOEAs). The author discussed various MOEA performance metric analysis methods currently present in the literature. Yen concluded no metric alone can quantify the performance of MOEA. To overcome the deficiencies a performance metric-based ensemble is suggested which can measure the performance of various MOEAs and gives a comprehensive comparison. The proposed method allows MOEAs to generate an approximation from a given set of population. A randomly chosen approximation front is generated and, through the process of double elimination tournament, the MOEA that wins is given rank one. The already utilised set of approximation front is then removed, and the process is repeated with the remaining MOEAs until all the assigned MOEAs are given a rank. This process allows a MOEA which performs relatively weak in a specific front to still be selected. Five state-of-the-art MOEAs were chosen for experimentation, - SPEA2, NSGA-II, IBEA, PESA-II, and MOEA/D. In order to test the MOEAs five bio objective test instances were used, namely ZDT[64], DTLZ-2 [65], WFG-1, WFG-2 [66], and 10 objective DTLZ-1. After experimental analysis it was proven that various MOEAs have different rankings in different test scenarios. The result being SPEA2 being rank one in ZDT 1 and ZDT 2 tests; NSGA-II being rank one in ZDT 3 tests, MOEA/D being rank one in ZDT 4 and ZDT 6 tests, and finally IBEA being rank one in 3-objective DTLZ-2, 5 objective WFG-1 and WFG-2, and 10 objectives DTLZ-1.

Rosales-Perez *et al.* [67] proposed a novel method known as Evolutionary Multi Objective Model and Instance Selection with Pareto based Ensemble (EMOMIS-PbE). The proposed method is used with support vector machines to generate hyper parameters and instance selection. Two methods were used for instance selection, filter based, and wrapper based. Although both methods performed with the same classification accuracy, filter-based methods achieved higher reduction rates with training sets. The proposed method was tested on 43 datasets from the KEEL[68] repository. The experimentation results were compared with different Pareto based ensemble strategies and it was found that Boosting outperformed other ensemble methods and achieved a higher reduction rate as well. The proposed method was later compared with non-IS and non-MS based methods. Wilcoxon signed rank test revealed a significant *p* value in favor for EMOMIS-PbE boosting. Similarly, the model was tested with single objective models and traditional models. According to results, the proposed method out-performed both in terms of reduction rate and classification accuracy. EMOMIS-PbE was able to train SVM with different subsets of the training dataset, thus achieving diversity. The model gave good trade-off between the size of the training dataset and performance, although generally SVM performed well with higher numbers of classifiers.

Several methods were discussed in literature that used evolutionary algorithms as an optimisation technique to optimise ensemble classifiers in various ways [56, 57, 63, 67, 69-72]. Three prominent evolutionary algorithms are Genetic Algorithm (GA), Multi Objective Optimisation Algorithm (MOEA), and Particle Swarm Optimisation (PSO). Particle swarm is an optimisation algorithm, inspired by the social behaviour of flocks of birds. It was originally introduced by Kennedy, Eberhart, and Shi in 1995 as an effort to express the behaviour of flock of birds in mathematical search space. Particle swarm optimisation is a metaheuristic algorithm as it has no knowledge about the search space and makes no assumption, allowing it to search a very significant search space to find a global minimum. Everything in PSO is considered to be a particle and each particle has a personal best and a global best, and the objective is to find a global best which has a minimum impact on personal best for all particles [73]. Multi objective optimisation algorithms [74] come under the area of mathematical optimisation techniques. MOEAs are best utilized when no single solution exists between two conflicting objectives. For such a problem, several solutions exist which can form a Pareto front. A solution is deemed optimal if no such solution exists in the optimal Pareto front that can improve the current solution without negatively affecting other objectives. Genetic Algorithms [75] are inspired by the process of natural selection also known as evolution. They optimise a problem space by using cross-over, selection and mutation bio-operators.

The sample size of training data contributes greatly towards the quality of the trained classifiers. A very large dataset would train a classifier that is over-fitting and a small dataset would train a classifier that is under-fitting. Therefore, a balance must be maintained when selecting a sample/sub-sample of dataset in order to train classifiers. As such, Beleites *et al.* [76] suggested that at least 75 – 100 records must be available per training set to train a good, if not a perfect, classifier. They showed empirically that the minimum criteria are 10 records. Any training set less than 10 records would end up training a classifier that is extremely biased towards the training dataset.

2.6 Hybrid algorithm-based ensemble classifier methods

Ensemble classifier approaches either exploit the input feature space or input sample space to 'perturb' the input data so to train a pool of base classifiers on perturbed sub samples, for example bags in bagging-based ensembles, and data clusters in clustering-based ensembles. Ensemble classifier approaches that exploit both the input feature space, and input sample space are known as hybrid ensemble approaches. As such Zhiwen *et al*. [77] proposed a hybrid incremental ensemble learning (HIEL) framework for real-world noisy data classification. In the proposed approach the authors first generated bags from the input sample space and utilized an LDA classifier to identify noisy samples. Classifiers are then incrementally selected using a criterion function, and each classifier is assigned a respective weight in the selection process. This weight is later utilised in the weighted voting stage to generate the final class labels of the ensemble. Similarly, Yang *et al.* [78] proposed a hybrid sampling-based clustering ensemble with global and local constitutions. The authors incrementally generated input partitions using clustering with a novel consensus function. An ensemble is generated by training base classifiers on generated data clusters. Haque et al. [109], proposed a multi objective meta heuristic ensemble method for customer churn detection. The authors suggested that many existing ensemble works convert a multi objective problem into a single scalar value. Doing so can limit the search space of optimization. They also proposed a methodology of applying optimisation algorithms as a black box tool for business analytics. A brief summary of recent ensemble classifier methods is given below in Table 1.

Table 1: Summary of existing ensemble classifier methods

Author(s)	**Summary**
Ensemble classifier component size	
Bonab **[19]**	Theoretically provided a framework to show that having as many classifiers as there are class labels in the dataset gives an ensemble classifier that can achieve the highest accuracy
Oshiro *et al.* **[20]**	Proved in their research that adding trees does not necessarily increase the accuracy of the ensemble, rather it adds more computational complexity to the ensemble classifier
Lattinne *et al.* **[21]**	Proved that adding more classifiers in an ensemble does not increase accuracy, and suggested that one should incrementally add classifiers and compare the performance of the current ensemble classifier with the one before
Lustosa *et al.* **[22]**	Surveyed various dynamic ensemble classifier selection methodologies.
Lysiak *et al.* **[23]**	Suggested that classifier accuracy and diversity can be considered an optimisation problem
Clustering-based ensemble classifier methods	
Rahman **[26]**	Accuracy and diversity were used as inputs to a multi-objective optimisation problem and classifiers were selected by a genetic algorithm.
Rahman **[27]**	Proposed a novel cluster-oriented method for generating ensemble classifiers. Author suggested using k-means algorithm to cluster dataset into distinct clusters and train classifiers on all generated clusters.
Rahman **[27]**	Proposed a method proposed a method in which classifiers are added to the ensemble classifier based on accuracy and diversity comparisons.
Fletcher **[29]**	Proposed a methodology of training heterogeneous classifiers on distinct clusters filtered by pruning redundant clusters using Jaccard index.
Huang *et al.* **[31]**	Authors suggested that most of the clustering-based ensemble classifier approaches have one limitation in common, that is they all consider clustering algorithms equally regardless of their performance and reliability.
Classifier selection-based ensemble classifier methods	
Bagheri *et al.* **[32]**	Different classifiers were trained on different features which were combined using Dempster-Shafter methodology.
Zhang **[34]**	Proposed a new method to generate oblique decision tree ensemble by using support vector machine at each node to obtain clustering hyperplanes so that similar features are closest to each other and different are farthest.
Bhattacharjee *et al.* **[35]**	An empirical survey was conducted to bridge the gap between different multi objective optimisation problems. Different datasets explicitly designed for MOEAs were utilised and analysis was provided.
Qi *et al.* **[37]**	Authors proposed a novel method of combining deep learning and support vector machine. The proposed method takes advantage of auto encoder technique and integrates ensemble classifiers. Features were selected using a deep network of SVM called deep SVM, where each SVM was selected by ex-adaboost based on maximum accuracy and diversity. At the end features with the highest weight were chosen as inputs for a standard SVM which greatly increase accuracy.
Chang **[38]**	Proposed a complementary ensemble selection method. Proposed method uses a priority queue-based diversity measure to select classifiers that maximise diversity of each classifier that becomes part of the ensemble.

Author(s)	**Summary**
Huang *et al.* **[39]**	Proposed an ensemble driven support vector clustering methodology. The proposed methodology can overcome the weaknesses in support vector clustering by performing better parameter estimations.
Kilic **[40]**	In the proposed methodology expertise of base classifiers was selected based on different datasets, therefore only those classifiers which are well-suited for a specific dataset are selected and become part of the ensemble.
Mao *et al.* **[42]**	The proposed method mitigates the dilemma problem of ensemble classifier by introducing a weight vector which uses an approximation form decomposed into a quadratic form and an error term.
Gu **[44]**	Proposed a new semi supervised ensemble learning algorithm. The proposed algorithm generates several heterogeneous classifiers and fuses them together through majority predictions.
Kim *et al.* **[45]**	Proposed a hierarchical ensemble generation methodology using feature extraction and classifiers selection. The proposed methodology selects features which can train classifiers that can achieve higher classification accuracies, and systematically selects only those classifiers that can achieve highest classification accuracy to become part of the ensemble.
Yin *et al.* **[46]**	Proposed a mathematical framework for ensemble classifier selection using sparsity and diversity learning.
Abellan **[48]**	Provided a comparative analysis of the base classifier which can be used as a component of an ensemble to get highest classification accuracy for credit risk analysis.
Krawczyk **[49]**	Query by committee active learning methodology is adopted for drifting online data streams in order to better classify the datasets.
Santucci *et al.* **[50]**	Proposed a novel methodology of randomising classifier parameters based on joint probability distribution of parameters of given classifier.
Bock **[51]**	Empirically compared the performance of two rotation-based ensemble methods for customer churn prediction. According to results rotation boost outperformed other methods.
Evolutionary algorithm-based ensemble classifier methods	
Kadkhodaei **[52]**	Authors proposed a method of adding a layer of entropy between meta classifiers in stack generalization. It was suggested that using diversity as a measure to select the best set of base classifiers would have a lesser impact on computational resources
Han *et al.* **[53]**	Authors proposed a novel method of using Attractive and Repulsive Particle Swarm Optimisation to select base classifiers from the initial pool of classifiers in Extreme Learning Machine (ELM) by considering both classification accuracy and diversity
Escalante *et al.* **[54]**	Authors proposed a novel method of using Particle Swarm Optimisation to generate an ensemble classifier
Yang *et al.* **[55]**	Authors proposed a novel method of Clustering Ensembles using Particle Swarm Optimisation CL-POS. In the proposed method they computed the weight of each cluster of ensembles using Particle Swarm Optimisation.
Bhowan *et al.* **[56]**	Authors suggested a genetic programming multi-objective optimisation method to evolve a diverse ensemble classifier. The objective of the proposed method was to develop a multi-objective genetic programming framework for classification of data with unbalance majority and minority classes as learning objective
Gu **[57]**	Proposed an approach using three diversity measures as explicit criteria along with accuracy as a multi objective optimisation problem to generate ensemble classifiers.
Ribeiro e al. **[58]**	Proposed a multi-objective design framework for member generation and member selection to generate an ensemble classifier.
Zhao *et al.* **[60]**	Used three minimisation objectives to generate an ensemble that can achieve the best DET scores over benchmark datasets.
Zhang *et al.* **[61]**	Used BPSO as a multi-task optimization toolbox to optimise the feature space to generate an ensemble classifier that can achieve the highest classification performance.
Yu *et al.* **[62]**	An evolutionary sample selection process is proposed that is typically advantageous against small datasets with large number of features.

Author(s)	**Summary**
Yen **[63]**	Proposed an ensemble method that focuses on performance metrics of different MOEAs to form an optimal pareto front. Authors empirically compared different MOEAs using different datasets, and suggested that it is inconclusive now to quantify one MOEA to be superior to others for different datasets
Rosales *et al.* **[67]**	An evolutionary multi objective optimisation method with instance selection to optimise SVMs with pareto based ensemble is proposed.
Escalante *et al.* **[69]**	Proposed a methodology of using particle swarm optimisation as a model selection methodology for selecting classifiers to generate an ensemble of classifier.
Hybrid algorithm-based ensemble classifier methods	
Yu *et al.* **[77]**	Proposed a hybrid ensemble learning method that exploits both the feature space and sample space of noisy real-world datasets to generate an accurate ensemble classifier.
Yang *et al.* **[78]**	Authors proposed a hybrid sampling-based clustering ensemble with global and local constitutions. The authors incrementally generated input partitions using clustering with a novel consensus function
Asafuddoula *et al.* **[79]**	A hierarchical ensemble generation method was proposed where classifiers were selected based on accuracy and diversity.
Verma **[80]**	Proposed a novel cluster-oriented ensemble classifier method. The proposed method classifies dataset into atomic and non-atomic clusters and a fusion classifier maps or classifies a cluster to a class label.
Yang *et al.* **[81]**	Proposed a novel methodology of using particle swarm optimisation to optimise the parameters of clusters in forming ensembles.
Kadkhodaei **[82]**	A method based on diversity of classifiers is proposed which is calculated using both an entropy measure and oracle output of classifiers.
Han *et al.* **[83]**	Proposed an extreme learning method to generate an ensemble of classifier using Attractive and Repulsive Particle Swarm Optimisation. ARPSO is used to optimise the accuracy and diversity of the ensemble.
Haque *et al.* **[109]**	Proposed a multi objective meta heuristic ensemble method for customer churn detection. The authors suggested that many existing ensemble works convert a multi objective problem into a single scalar value. Doing so can limit the search space of optimization. They also proposed a methodology of applying optimisation algorithms as a black box tool for business analytics.

It can be noted from Table I that various authors have utilized various strategies to generate an ensemble classifier. There are three areas generally when it comes to ensemble classifier generation which opens different areas of research. 1) According to the “law of large numbers” if enough subsamples are generated from a given sample the distribution of the generated subsamples will follow a uniform distribution. Based on the same principle data subsampling techniques such as bagging, or clustering is utilised in ensemble architecture to reduce the correlations of various classifiers that are trained on a single input data. If classifiers are trained on the entire input data they will be biased and there will consequently be no point in combining multiple classifiers. Therefore, to capitalize on the concept authors have utilised various techniques to reduce the correlation of trained classifiers on the input data such as bagging, boosting, and clustering. 2) A pool of classifiers known as the “base classifier pool” is generated by training classifiers on the data subsamples to "over produce" a pool of classifiers which are then utilised by a classifier selection methodology. Selecting classifiers from the pool is considered as a binary combinatorial problem therefore, authors have utilized various optimization algorithms to select the best subset of classifiers. 3) Lastly, various classifier fusion strategies have been proposed in research where the class decision of various "selected" classifiers are combined to produce a unanimous output of the ensemble. In the proposed ensemble framework clustering is utilized as a data subsampling techniques and various clustering validation techniques have been utilized to generate the optimal of data clusters which will be utilized for the training of base classifiers. Furthermore, evolutionary algorithm is utilized to dynamically select the best subset of classifiers from the pool to generate an optimised ensemble. Finally, majority voting is utilised to generate the final class decisions of the ensemble.

2.7 Summary

As can be seen in the literature review, researchers have used different methodologies to generate accurate and diverse ensemble classifiers. Two prominent areas of research are using clustering to achieve data diversity, which consequently improves ensemble classifier accuracy, and using evolutionary or genetic algorithms to optimise ensemble classifier components. It is essential to further investigate optimal clustering value, which can achieve optimal data diversity and evolutionary algorithms, to optimise the pool of classifiers to generate an accurate and diverse ensemble classifier.

Chapter 3: **Ensemble Framework, Datasets, Evaluation Metrics, And Experimental Setup**

This chapter presents the proposed ensemble framework, the datasets, the evaluation metrics, and the experimental setup used throughout this thesis. Section 3.1 details the general ensemble framework used to develop further ensemble methods to mitigate the limitations of clustering-based ensembles and answer the research questions. Section 3.2 details the benchmark classification datasets, and the benchmark image datasets, that were used to evaluate the proposed ensemble methods. Section 3.3 details various evaluation metrics which were used to compute the classification performance, diversity of classifiers, and significance of the results of the proposed ensemble methods. Lastly, Section 3.4 details various hyper-parameters and the experimental setup used to test the proposed ensemble methods.

3.1 Ensemble framework

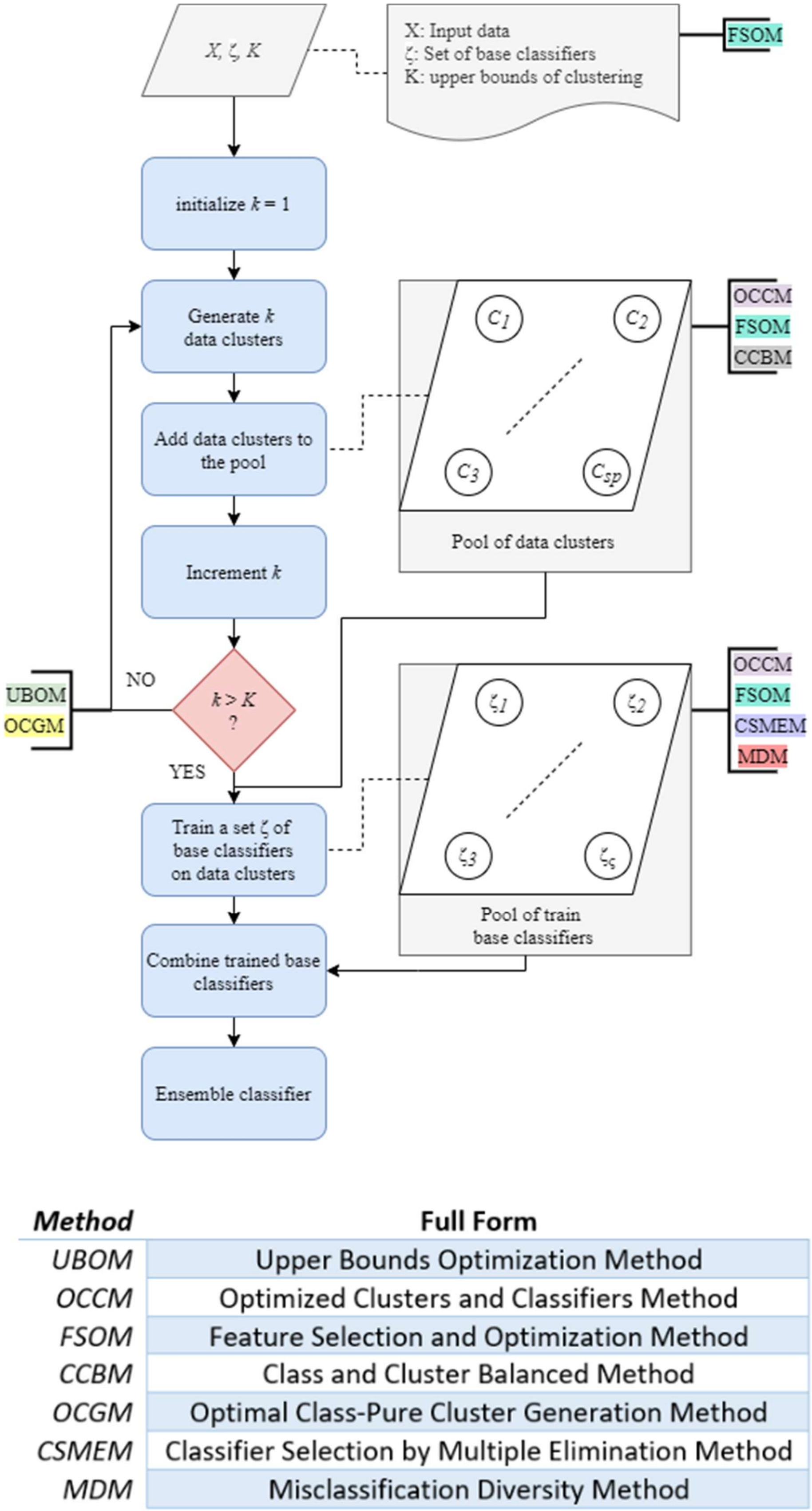
The proposed ensemble classifier framework is divided into three parts. Firstly, the input data is perturbed and a rich and diverse input space for the training of base classifiers is generated. To do so, data clusters are generated incrementally with $k = 1$, in the first iteration going up to a maximum of K data clusters. All clusters are added to the pool, to generate a pool of data clusters, which is then utilised to train a set of diverse (structurally different) base classifiers on each data cluster to generate the pool of base classifiers. Secondly, all classifiers are selected from the pool to generate the final ensemble. Thirdly, and finally, all classifiers are used to classify the unseen test set T and their class decisions are fused. The overall framework is illustrated in Figure 2 with various sections identified where different novel ensemble methods were proposed to further develop the framework and answer the research questions.

To generate a pool of data clusters for the training of base classifiers, consider an input data $X = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ where $x \in \mathbb{R}^d$ is a d-dimensional vector of features, and y is its respective class labels given as $y \in \{1, 2, \ldots, \kappa\}$, having n number of samples, and κ discrete class labels. A random subspace of data clusters is generated by incrementally clustering the input data from 1 to K data clusters. In each iteration, a total of k data clusters are generated and added to the pool. Clustering is achieved by grouping similar data samples into different groups (clusters) with a common mean. This is achieved by minimising the squared Euclidean distance from the centroid of a data cluster given as:

$$\operatorname{argmin}\left(\sum_{j_i, c_1, \dots, c_k}^{n} \left\| x_i - c_{j_i} \right\|^2\right) \ j_i \in \{1,2, \dots , k\}, \tag{1}$$

where x is a feature vector belonging to a data cluster C, k is the number of data clusters that are generated, n is the total number of samples in the dataset, and c is the centroid of a data cluster. Clustering is utilised as an alternative to Bagging to generate a random subspace for training of base classifiers. The benefits of clustering input data are two-fold: firstly, clustering input data generates mutually exclusive data clusters that help in creating and training diverse classifiers; secondly, data clusters identify dense local regions within the data, and any base classifier trained on such local regions can identify that region effectively. Essentially this is the reverse of a kernel function and each classifier has local expertise. Another added benefit to clustering is that there are several tweakable parameters when generating data clusters, this enables us to fine-tune the process and generate an input space that can maximise the classification accuracy of the ensemble.

After generating a pool of data clusters, a set of diverse base classifiers $\zeta = \{\zeta^1, \zeta^2, \dots , \zeta^\varsigma\}$ such as SVM, ANN, NB, DT, KNN, LDA, *etc.* is trained on all generated data clusters. The choice and type of base classifier is investigated in this research. Thus, a pool of trained base classifiers bcp is generated which is then utilized to generate the ensemble. A stepwise algorithm for proposed ensemble classifier framework is given below in Algorithm 1.


X, ζ, K
X: Input data
ζ: Set of base classifiers
K: upper bounds of clustering
FSOM
initialize k = 1
Generate k data clusters
Add data clusters to the pool
Increment k
$k > K$?
NO
YES
UBOM
OCGM
C_1
C_2
C_3
C_{sp}
Pool of data clusters
OCCM
FSOM
CCBM
Train a set ζ of base classifiers on data clusters
Combine trained base classifiers
Ensemble classifier
ζ_1
ζ_2
ζ_3
ζ_ς
Pool of train base classifiers
OCCM
FSOM
CSMEM
MDM

Method	Full Form
UBOM	Upper Bounds Optimization Method
OCCM	Optimized Clusters and Classifiers Method
FSOM	Feature Selection and Optimization Method
CCBM	Class and Cluster Balanced Method
OCGM	Optimal Class-Pure Cluster Generation Method
CSMEM	Classifier Selection by Multiple Elimination Method
MDM	Misclassification Diversity Method

Figure 2: Proposed ensemble framework

Algorithm 1: General ensemble framework

```
Input: Training dataset $X = \{(x_1, y_1), \dots, (x_n, y_n)\}$,  $x \in \mathbb{R}^d, y \in \{1,2, \dots, \mathcal{K}\}$,
       Validation dataset $V = \{(x_1, y_1), \dots, (x_n, y_n)\}$, Base classifier $\zeta$, upper bounds of
       clustering $K$
Output: Ensemble classifier accuracy

for $k = 1$ to $K$ do
    $sp^i$ ← partition training dataset into $k$ clusters by minimising the squared Euclidean
    distance of each sample from the cluster centroid
endfor
foreach data clusters $C$ in pool $sp$ do
    $bcp^i$ ← train a set of base classifiers $\beta = \{\zeta^1, \zeta^2, \dots, \zeta^\varsigma\}$ on the data cluster $C$ and
    add to the pool
endfor
Compute class decisions of all base classifiers in the ensemble using the test set $T$
Using eq (1), (2), and (3) fuse decisions of all classifiers to get final class labels of the
ensemble
Using eq (4) compute ensemble accuracy
```

3.2 Datasets

3.2.1 Benchmark datasets from UCI repository

To evaluate the performance of the proposed ensemble framework, 27 benchmark classification datasets from the University of California Irvine (UCI) machine learning repository [84] were used. A brief description of these datasets, such as number of records, number of features and number of class labels, is given in Table 2. These datasets are considered as structured datasets because they are tabular in nature, with the number of samples meaning the number of rows, the number of features meaning the number of columns, and the number of class labels meaning discrete class labels. These datasets were downloaded from the repository and a library was created which is utilised throughout the course of this thesis. It can be noted that a mix of datasets with different attributes are chosen. This is not only done to test the efficacy of the proposed approach thoroughly. As datasets are used in other similar works comparative analysis is enabled., Depending on the proposed ensemble method, some or all the datasets mentioned in Table 2 are used in this thesis and analysis is provided.

UCI datasets were chosen in this research because many existing recent and old ensemble works have used the same in other similar ensemble works which allows for the comparison of performance [7]. Moreover, a mixture of datasets is selected, ranging from very small datasets with large features such as *Spectfheart* dataset with 44 features and only 267 data samples, to large datasets such *Statimag* with 6435 samples and 36 features. This enables a thorough analysis of the classification performance of the proposed ensemble methods as a variety of datasets are used.

Table 2: UCI repository benchmark machine learning classification datasets used in experimentation

Dataset	**No. of Records**	**No. of Features**	**No. of Class Labels**
Appendicitis	106	7	2
Balance	625	4	3
Bank note	748	4	2
Breast cancer	699	9	2
Bupa	345	6	2
Diabetic	768	8	2
E.coli	366	7	2
Fertility	100	9	2
Glass	214	10	7
Haberman	306	3	2
Hayes-Roth	160	5	3
Heart	270	13	2
Hepatitis	80	19	2
Ionosphere	351	33	2
Iris	150	4	3
Liver	345	6	2
Plan relax	182	12	2
Sonar	208	60	2
Segment	2310	19	7
Spectfheart	267	44	2
Statimag	6435	36	6
Teaching	151	5	3
Thyroid	215	5	3
Vehicle	946	18	4
WDBC	569	30	2
Wine	178	13	3
Zoo	101	17	7

3.2.2 Benchmark image datasets

Two benchmark image classification datasets have also been used in this thesis. These image benchmark datasets are commonly used to evaluate the performance of image classification algorithms such as CNNs by many researchers and allow comparative analysis. The Modified National Institute of Standards and Technology (MNIST) [85] handwritten digit recognition dataset, and Canadian Institute For Advanced Research (CIFAR-10) [86] object recognition dataset, are standard benchmark image recognition datasets, and are used by many researchers. They not only allow for evaluating the performance of the proposed ensemble methods, but also allow comparison of results with existing methods.

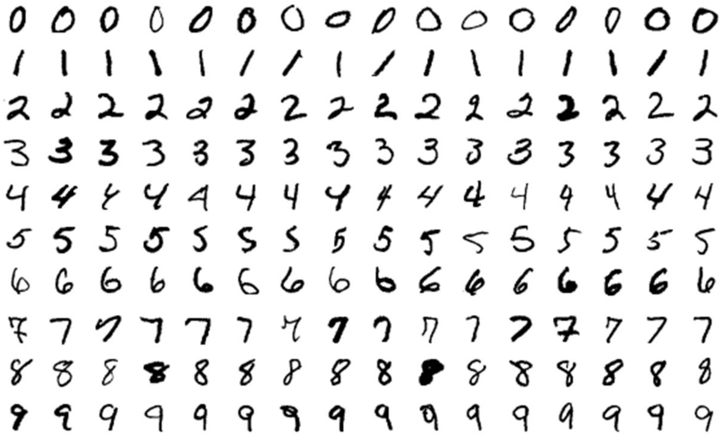


Figure 3: MNIST handwritten digit recognition dataset
REF: *MIT V*isualisation of the MNIST database
https://github.com/saradhix/mnist_visual

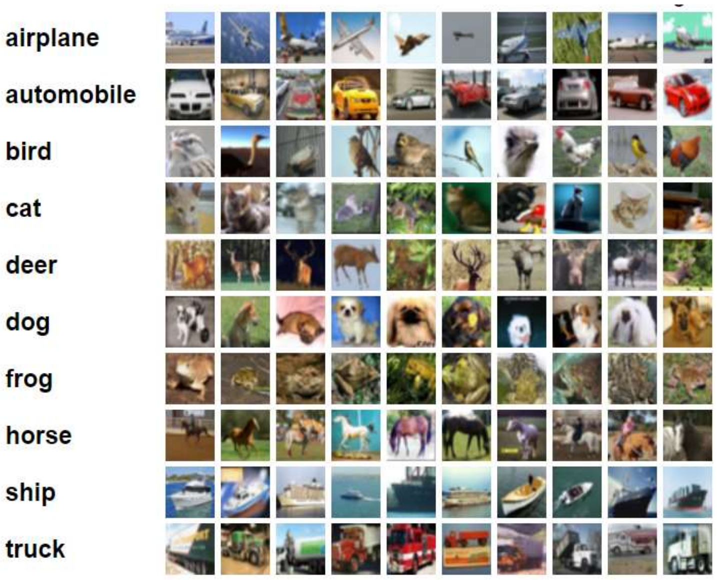


Figure 4: CIFAR-10 dataset
REF: The CIFAR-10 and CIFAR-100 are labelled subsets of the 80 million tiny images dataset
https://www.cs.toronto.edu/~kriz/cifar.html

The MNIST dataset is a database of hand-written digits utilised for training and testing of different image recognition systems, typically to evaluate the performance of convolutional neural networks. The MNIST dataset contains 70,000, 28 by 28 grey scaled in 60,000 training images and 10,000 testing images. Some example images from MNIST dataset are shown in Figure 3. Similarly, CIFAR-10 dataset is another benchmark dataset that consists of various images of 10 different objects. CIFAR-10 contains 60,000, 32 by 32 colour images with 50,000 training images and 10,000 test images. Some example images from CIFAR-10 datasets are shown in Figure 4. Although, the proposed ensemble method is mainly tested on structured datasets for evaluation purposes however, image classification has been added to evaluated whether the proposed method can be extended in the area of deep learning and whether an ensemble of deep learners outperform a single deep learner.

3.3 Evaluation metrics

3.3.1 Classification accuracy metric

To compute the classification accuracy on test data, the predicted class labels of the proposed ensemble method are generated by using the unseen test set T with feature vectors x, and class labels y. Every trained base classifier in the ensemble pool is utilised and its class decisions are produced. A classifier in the pool can be defined as:

$$y' = \zeta(x), \tag{2}$$

where y' are the predicted class labels and x is a feature vector. Therefore, an ensemble consisting of n base classifiers is given as:

$$\xi = \{\zeta(x)^1, \zeta(x)^2, \dots, \zeta(x)^n\}, \tag{3}$$

To get the predicted class labels y' of the ensemble a column wise mode is taken. This depicts majority voting and is given as:

$$y' = mode(\xi), \tag{4}$$

Therefore, utilising the predicted class labels y' and the true class labels y from the test set T the final ensemble classifier accuracy a is computed as follows:

$$a = \frac{\sum_{x_i \in T} I(y_i', y_i)}{n}, \tag{5}$$

$$\text{where } I(y_i', y_i) = \begin{cases} 1, & y_i' = y_i \\ 0, & y_i' \neq y_i \end{cases}$$

3.3.2 Incorporation of randomness

To accommodate for randomness that exists in the data, stratified 10-fold cross validation is adopted with one-fold being used for testing, also known as the test set T, and nine folds being used for training. The training is further partitioned with 80% being used as the training set X and 20% being used as the validation set V for optimisation and/or parameter adjustment. Depending on where the results are being published, either average classification accuracies over 10-folds are computed and reported, or average over 30 independent runs with each run having 10-fold cross validation are reported.

3.3.3 Pairwise diversity metrics

Different pairwise diversity measures are tested and measured in this thesis. These diversity measures were originally proposed in [24] and are Double Fault (DF) measure, Q statistics (Q), and Disagreement Measure (DM). These diversity measures are calculated by first computing the dissimilarity matrix given as:

Table 3: Relation between pairwise diversity measure calculations

	D_j correct (1)	**D_j wrong (0)**
$D_i$correct (1)	N_{11}	N_{10}
$D_i$wrong (0)	N_{01}	N_{00}

Using the dissimilarity matrix, the pairwise diversity measures are calculated as follows:

$$Q(D_i, D_j) = \frac{N^{11}N^{00} - N^{01}N^{10}}{N^{11}N^{00} + N^{01}N^{10}}, \tag{6}$$

$$DM(D_i, D_j) = \frac{N^{01} + N^{10}}{N^{11} + N^{10} + N^{01} + N^{00}}, \tag{7}$$

$$DF(D_i, D_j) = \frac{N^{00}}{N^{11} + N^{10} + N^{01} + N^{00}}, \tag{8}$$

3.3.4 Significance testing

To further validate the efficacy of the results, a series of non-parametric signed rank tests [87] with an alpha significance level of 0.05 were adopted.

3.3.5 Cluster-validation metrics

Various cluster validation metrics such as *Calinski Harabasz* analysis, *Davies Bouldin* analysis, gap analysis, elbow analysis, and dendrogram analysis are discussed in research [88]. In this thesis we opted for *Silhouette* analysis [89], which is the dissimilarity of a data points associations with its cluster, and is given as:

$$s(i) = \begin{cases} 1 - \frac{a(i)}{b(i)}, & if a(i) < b(i) \\ 0, & if (ai) = b(i), \\ \frac{b(i)}{a(i)} - 1, & if a(i) > b(i) \end{cases} \tag{9}$$

where $a(i)$ is the similarity of a data point i, and $b(i)$ is the dissimilarity of a data point i in a data cluster. For further details please refer to [89]. The *Silhouette* score ranges between {-1, 1} with a small value of s, which means that a datapoint i is well matched to its given cluster and a larger value means otherwise.

3.4 Experimental setup

All proposed ensemble classifier methods are implemented in MATLAB [90], default implementation of base classifiers in MATLAB such as SVM, ANN, DT, ND, KNN and LDA were used. For clustering the default implementation of K-means algorithm in MATLAB was used. Mostly default parameters of base classifiers were used besides the ones mentioned in Table 3. For ensemble classifier approaches that incorporate optimisation, default implementation of PSO in MATLAB "*particleswarm*" was used for optimisation. The parameters of PSO were as follows:

- Particles (swarm size) in PSO were represented as a bit string *pop* which simply is the number of classifiers in the pool bcp
- The binary threshold θ used is 0.6 which is the same as in [61] to determine whether a particle is 0 or 1
- The maximum evaluation time set was 2000 where for a single task it is 1000 (2000/2) because our method is optimizing two tasks in parallel
- A stall iteration limit of 200 was set to account for deadlocks.

Table 3: Various parameters used in the training of proposed ensemble methods

Algorithm / Classifier	**Parameter**	**Values**
Neural network	Hidden neuron	Random between 10-30
	Training function	Levenberg-Marquardt backpropagation / Bayesian regularisation backpropagation / Scaled conjugate gradient backpropagation / Resilient backpropagation
	Number of epochs	Random between 500-1000
	Hidden neuron	Random between 5-10
	Error goal	1e-5
Multi class support vector machine	Kernel function	Gaussian / Radial / Linear
	Iteration limit	Random between 1000 - 5000
Naïve Bayes	Distribution function	Kernel
K-Nearest neighbour	Number of neighbours	Random between 4-10
Decision tree	Minimum leaf size	No of class labels
Discriminant analysis	Kernel function	Polynomial
K-means	Number of iterations	2400
Particle swarm optimisation	Maximum iteration	100
	Stall iteration	10
	Swarm size	Number of classifiers in the pool

The parameter values mentioned in Table 3 are randomised so that classifier diversity is also incorporated in the ensemble, as classifiers on different data clusters will not only be trained on different regions but will also have different learning capabilities. The ranges of these parameters were searched exhaustively through trial and error, and the ones listed achieved the highest classification accuracy.

3.5 Summary

This chapter presented overall ensemble frameworks with several novel methods. different evaluation metrics and datasets that are used throughout this thesis are discussed. The methods are discussed in detail in later chapters. The tests conducted to validate the statistical significance of the results are also mentioned. The experimental setup is also detailed, and the hyper-parameters that were used throughout this thesis are also listed.

Chapter 4: **Upper Bounds Optimisation Method**

This chapter firstly proposes a novel ensemble method, namely Upper Bounds Optimization Method (UBOM), to mitigate the problem of having a fixed upper bound K when generating data clusters. Secondly, an algorithm of incorporating UBOM into ensemble framework from Section 3.1 is given. Lastly, details of the experiments and a comparative analysis of the results is provided.

4.1 Ensemble with upper bounds optimisation method

4.1.1 The proposed method

Most of the existing ensemble approaches discussed in the literature use a fixed upper bound of K for generating data clusters. Since datasets have different characteristics it is not an ideal strategy. Moreover, due to noise and randomness, the generated data clusters might also contain noisy samples. Therefore, a novel ensemble method called UBOM [91][1] is proposed to tackle these limitations. The proposed ensemble method, instead of using a single upper bound of K optimistically, searches for the best value of K through the incorporation of an optimisation algorithm. The proposed method represents selecting the value of K as a single objective optimisation problem. In a single iteration of search space there are $sp = k(k+1)/2$ data clusters. The problem search space is defined as follows:

$$minimize\ (f(sp)) \text{ subject } sp\ to\ K, \tag{10}$$

where K is the given upper bound of clustering, sp is the pool of generated data clusters, $f(sp)$ is the fitness function that takes all generated data clusters to train a multitude of single base classifiers, (any classifier can be used here but we opted for a multi class SVM with a gaussian kernel), and generates an ensemble. The ensemble's predicted class labels y' are generated using equations (1), (2), and (3) with validation dataset V having y true class labels and n number of samples. The predicted class labels are then utilised to compute the root mean square error *(*RMSE*)* of the ensemble which is the cost of the objective function given as follows:

[1] The work presented in this chapter has been published in the following paper: Z. Jan, B. Verma, and S. Fletcher, "Optimising Clustering to Promote Data Diversity When Generating an Ensemble Classifier," Proceedings of the Genetic and Evolutionary Computation Conference Companion, 2018, pp. 1402-1409

$$f(sp) = \frac{\sqrt[2]{\sum_{q=1}^{n}\left(y'_{q} - y_{q}\right)^{2}}}{n}, \qquad (11)$$

where y' is a column vector of predicted class labels, y is a column vector of actual class labels (ground truth), and total number of samples in the dataset are n.

The value of K for which the objective function achieved the minimum cost is selected.

4.1.2 Ensemble classifier generation using UBOM

The ensemble framework from Section 3.1 is utilised with UBOM to generate an ensemble with optimised value of K. In the proposed ensemble framework UBOM is identified and instead of incrementally increasing k to a fixed upper bound K, the proposed method searches for the optimum value of K and uses that to incrementally generate data clusters. The generated pool of data clusters is utilised to train a set of diverse base classifiers, and every trained classifier in the pool is utilised to classify the unseen test set T. Their class decisions are combined through majority voting and final ensemble accuracy is computed using equation (5). A stepwise algorithm of ensemble classifier generation using UBOM is given in Algorithm 2.

Algorithm 2: Ensemble classifier generation using UBOM

Input: Training dataset X, Validation dataset V, Testing data set T, Set of base classifiers ζ

Output: Upper bounds optimised-based ensemble classifier

```
Initialise $i$ and a population of $K$ particles
while termination criteria
  for $k = 1$ to $K$ do
     $sp^{i}$ ← partition training dataset into $k$ clusters by minimising the squared
     Euclidean distance of each sample from the cluster centroid
  endfor
while
  Map each particle to a cluster in the search space
  Calculate the fitness of the population using equation (11) by training a
  base classifier on all data clusters in the population using validation set $V$
Update the local best and global best of the population
```

Algorithm 2: Ensemble classifier generation using UBOM

```
    Update particle velocity and position
endwhile
initialise $i$
foreach data clusters $C$ in pool $sp$ do
    $bcp^i \leftarrow$ train a set of base classifiers $\beta = \{\zeta^1, \zeta^2, \dots, \zeta^\varsigma\}$ on the data cluster $C$ and add to the pool
endfor
Use the pool $bcp$ of trained classifiers to predict class labels of the unseen dataset, the test set and calculate ensemble classification accuracy
```

4.2 Experiments, results and analysis

4.2.1 Datasets

The following benchmark classification datasets, namely *Breast cancer, E.coli, Glass, Haberman, Ionosphere, Iris,* and *Vehicle* from the UCI Machine Learning repository [84] were utilised to evaluate the performance of the proposed ensemble method. Details of these datasets are already mentioned in Table 2, in Section 3.2.

4.2.2 Experimental setup

This section details the experiments that were conducted to measure the effect of incorporating an evolutionary algorithm to optimise the upper bounds K. The proposed ensemble framework with UBOM is implemented in MATLAB, and a 10-fold cross validation is adopted to incorporate for randomness as in other similar works. A cluster similarity threshold was also introduced in experiments to discard redundant data clusters from the pool. The similarity was computed using the Jaccard Index [30] between two data clusters, and any two data clusters having a similarity score of more than 90% were discarded. The similarity threshold is computed as follows:

$$J(\mathrm{C}_i, \mathrm{C}_j) = \frac{|\mathrm{C}_i \cap \mathrm{C}_j|}{|\mathrm{C}_i \cup \mathrm{C}_j|} \quad \forall\, i, j \in \mathrm{sp}, \tag{12}$$

where C_i and C_j are data clusters in the pool of clusters sp

4.2.3 Results

The average classification accuracy of the proposed ensemble framework with UBOM over 10-folds is reported in Table 4. Also, given in Table 4 is the total number of data clusters generated for each dataset and the total number of data clusters that were utilised.

Table 4: Classification performance of the proposed ensemble framework using UBOM on UCI benchmark datasets

Datasets	**UBOM**	**Total Clusters**	**Clusters Utilized**
Breast cancer	0.970	17	12
E.coli	0.968	3	2
Glass	0.994	2	1
Haberman	0.755	9	7
Ionosphere	0.923	4	3
Iris	0.976	2	1
Vehicle	0.902	11	8

4.2.4 Discussion

In the proposed ensemble framework with UBOM, the value of K is optimised to promote ensemble accuracy for each dataset, rather than using a fixed upper bound for every dataset. Across all datasets, on average 66% of clusters were utilised and the remaining 34% clusters were discarded based on similarity. The similarity threshold was empirically calculated to be 0.9 or 90%, on trial and error basis, as it resulted in the highest classification accuracy. It is interesting to note that K is approximately 3.5 for most of the datasets and this is summarised in Figure 5. We can see from Figure 5 that as the number of samples in a dataset increases, the value of K also increases, pointing to the fact that K, and the number of samples in a dataset, has a directly proportional relation. However, it is not a linear relation as the slope of the curve decreases further across x-axis eventually becoming 0. Thus, proving to the fact that for a very large dataset, the value of K would not exceed more than 10 approximately and a rough estimation would be:

$$K = log_2(n), \tag{13}$$

where n is the number of samples in a dataset

One thing to note is that since data clusters are discarded based on similarity, the actual number is not the same as given by equation (13), but it does provide a good approximation.

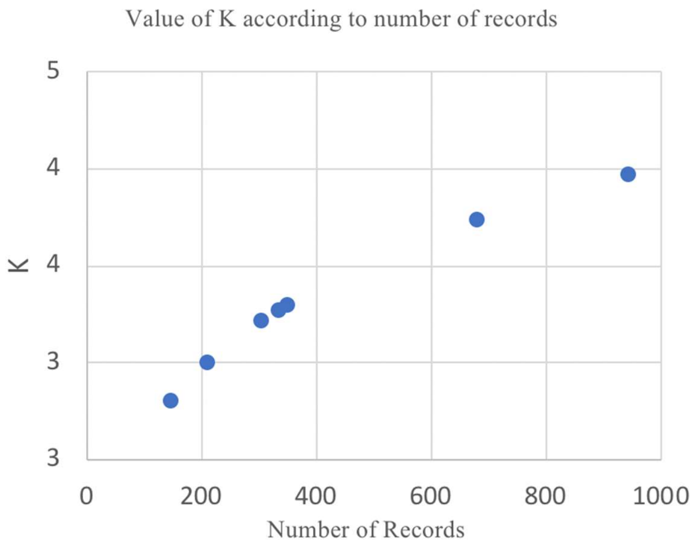


Figure 5: Value of K with respect to number of samples

4.2.5 Comparative analysis

This section details the comparative analysis of the proposed ensemble method with existing state-of-the-art ensemble classifier methods. Two legacy ensemble classifier methods, such as Bagging and Boosting [80], an incremental clustering-based ensemble classifier, namely Optimal Ensemble Classifier – Incremental Layered Clustering (OEC-ILC) [79], a Random Forest based ensemble classifier namely Multi Proximal Random Forest with Tikhonov (MPRaF-T) [34], and a majority voting-based ensemble classifier, namely Recall (REC) [92], are used to compare the classification performances . The classification accuracies were taken from the respective papers and the results are summarised in Figure 6 below.

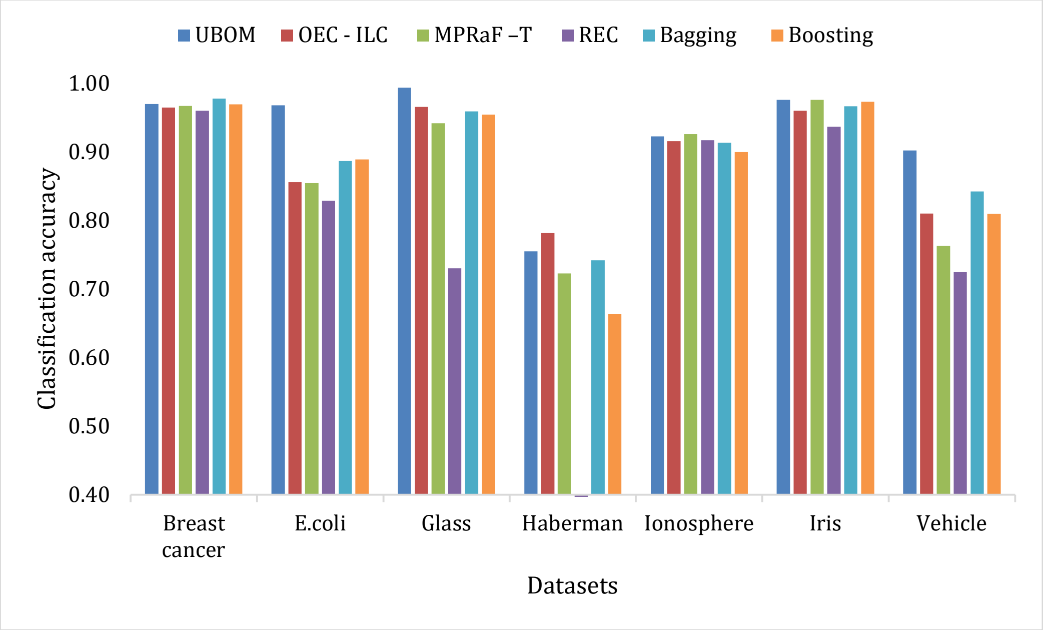


Figure 6: Performance of the proposed ensemble method UBOM in comparison with state-of-the-art ensemble methods

It can be noted that the proposed approach performed significantly better than other ensemble methods in four of seven datasets. The *p*-values are listed in Table 5, thus proving that the proposed ensemble method approximately achieved significant 1.03% performance gains over OEC-ILC, 1.05% over MPRaF-T, 1.09% over REC, 1.03% over Bagging, and 1.05% over Boosting.

Table 5: p-values of Wilcoxon signed rank tests of UBOM

Methods	***p*-values**
OEC-ILC	**0.045**
MPRaF-T	**0.025**
REC	**0.013**
Bagging	**0.014**
Boosting	**0.008**

4.3 Summary

In this chapter, an ensemble method was proposed that incorporates an evolutionary algorithm to optimise the upper bound K so that it can achieve the highest classification accuracy. The optimum value of K is then utilised to train a diverse set of base classifiers, which are then combined to generate the ensemble. The proposed method was incorporated with the ensemble framework from Section 3.1 to generate an ensemble, and experiments were conducted to analyse the performance. Through experiments it was evident that the number of clusters that should be generated for each dataset follow a log relationship with the number of samples, and that a fixed upper bound for different datasets is not an ideal strategy. A detailed comparative analysis with existing state-of-the-art ensemble classifier methods was also provided, and through experiments it was proven that the proposed ensemble method performed significantly better than other ensemble methods.

Chapter 5: **Optimised Clusters and Classifiers Method**

This chapter proposes a novel ensemble method, namely Optimised Clusters and Classifiers Method (OCCM), that incorporates an evolutionary algorithm in two phases, firstly to optimise the entire pool of generated data clusters rather than searching for a fixed upper bound K, and second, to optimise the pool of trained base classifiers. The pool of trained base classifiers is optimised to generate an ensemble that can not only achieve high classification performance, but also has lower component size. The algorithm of training an ensemble using the proposed OCCM is also given and detailed experiments are provided. An algorithm of incorporating OCCM into ensemble framework from Section 3.1 is also given. Furthermore, in this chapter a hybrid ensemble method, namely Feature Selection and Optimisation Method (FSOM), is discussed that further extends OCCM by exploiting both the input features and input samples to generate an optimised ensemble classifier. The proposed ensemble method reduces the dimensions of the input data by identifying significant input features and uses the reduced input features data to generate an optimised training space and then an optimized pool of base classifiers. An algorithm of incorporating FSOM into ensemble framework from Section 3.1 is also given. The proposed ensemble method is evaluated on several benchmark datasets, and results are compared with existing state-of-the-art ensemble methods.

5.1 Ensemble with optimised clusters and classifiers method

5.1.1 The proposed method

Instead of searching for the best upper bound K for generating data clusters, we proposed an ensemble method OCCM [93][2] that optimizes the pool of generated data clusters by representing the entire pool as a discrete optimisation problem, and later optimises the pool of base classifiers that are trained on the optimised data clusters. The benefits of the proposed ensemble method are threefold. Firstly, due to noise and randomness, not all samples in a dataset are suitable for the training of base classifiers and consequently data clusters containing noisy samples will train base

[2] The work presented in this section has been published in the following paper: Z. Jan and B. Verma, "Evolutionary Classifier and Cluster Selection Approach for Ensemble Classification," ACM Transactions on Knowledge Discovery in Data, vol. 14, no. 1, pp. 1-8, 2019.

classifiers that will negatively impact the classification accuracy of the ensemble., The proposed ensemble method alleviates this issue by discarding noisy data clusters. Secondly, not all classifiers can learn data patterns with equal generalisation capability, and therefore classifiers that can negatively impact the performance of the ensemble should be discarded. By optimising classifiers, the proposed ensemble method not only discards redundant base classifiers but also any base classifier that will negatively impact the classification of the ensemble. Lastly, by optimising the pool of data clusters instead of a fixed value of K, we alleviate the need to have a fixed value of data clusters. Irrespective of how many data clusters are generated, the proposed ensemble method will only use a subset that is best for the training of base classifiers.

Therefore, in this ensemble strategy we optimise the generated pool of data clusters by representing it as a discrete optimisation problem. Selecting a subset of data clusters from the pool is considered as a combinatorial problem and, through the incorporation of an optimisation algorithm, we search for a subset that can achieve the global minimum. This is done by representing each data cluster as a particle in PSO population having its position restricted to 1 or 0. This is done to 'binarize' the operation of cluster selection. A cluster is selected from the pool if its respective binary representation is 1. The search space for optimisation is all generated data clusters given as:

$$minimize\ (f(sp')) \text{ subject } sp' \subset sp, \tag{14}$$

where sp is the pool of generated data clusters, sp' is a random subset of those clusters, and $f(sp')$ is the fitness function that takes as input a set of data clusters and computes the RMSE using equation (11), as in the previous ensemble method mentioned in chapter 4. At the end of optimisation, we have a subset of data clusters ideally that can achieve the highset classification accuracy, and any redundant and noisy data clusters are discarded. All data clusters from the optimised pool sp' are now utilized to generate a base classifier pool bcp by training a set of diverse base classifiers on all data clusters in the pool sp'. The pool of base classifiers bcp is now optimised to search for a subset of base classifiers that can not only generate an ensemble with highest generalization ability, but also lower component size. The selection of base classifiers from the pool is also considered as a discrete optimisation problem, since it is a combinatorial problem, therefore, the problem formulation is as follows:

$$minimize\ (f(\xi))\ \text{subject to}\ \xi\ \in bcp, \tag{15}$$

where $\xi\ = \{B^1, B^2, \dots, B^{bcp}\}$ which essential is a possible ensemble solution containing a random subset of trained base classifiers, and bcp is the pool of trained classifiers. $f(\xi)$ is the objective function is given as:

$$f(\xi) = f_1 + f_2, \tag{16}$$

f_1 and f_2 are the two variables of the objective functions. f_1 is the $RMSE$ of the ensemble calculated using equation (11) after computing the final predicted class labels y' of the ensemble with the validation dataset V. f_2 is the second variable which is simply the size of the ensemble (number of classifiers) given as:

$$f_2 = |\xi|, \tag{17}$$

where ξ is a possible ensemble solution

Ideally the proposed method identifies the subset of classifiers from the pool to generate an optimised ensemble classifier which can not only achieve the highest classification accuracy, but also have lower component size as well. Population in PSO is represented as:

$$pop = \left[\varphi^1, \varphi^2, \dots., \varphi^{|bcp|}\right], \tag{18}$$

$$\text{where } \varphi^n = \begin{cases} 1, & if\ \varphi^n > \theta \\ 0, & otherwise \end{cases}.$$

Therefore, the population of particles at the end of optimisation that have respective 1s are used to identify the classifiers that will be utilised to generate the optimised ensemble classifier.

5.1.2 Ensemble classifier generation using OCCM

The ensemble framework from Section 3.1 is utilised with OCCM to generate an optimised ensemble. In the proposed ensemble framework, OCCM is identified in Figure 2, and it generates a random subspace through incrementally clustering form k to an upper bound K. Instead of searching for an optimum value of K, the proposed method optimises the pool of data clusters, and then trained a set of diverse base classifiers on all optimized data clusters. The base classifier is also optimised, and a subset of base classifiers is chosen that can increase the ensemble accuracy and decrease the component size as well. A stepwise algorithm of ensemble classifier generation using OCCM is given in Algorithm 2.

Algorithm 2: Ensemble classifier generation using OCCM

```
Input: Training dataset $X$, Validation dataset $V$, Testing data set $T$, Set of base
        classifiers $\zeta$, Upper bounds of clustering $K$
Output: Optimized cluster, and classifier-based ensemble classifier

Initialize $k$ = 1, and $i = 1$
for $k = 1$ to $K$ do
    $sp^i$ ← partition training dataset into $k$ clusters by minimising the squared
    Euclidean distance of each sample from the cluster centroid
endfor
while termination criteria
    Map each particle to a cluster in the search space by generating a binary bit
    string pop representing each data cluster in the population
    Calculate the fitness of the population using equation (11) by training a
    base classifier on all data clusters in the population on validation dataset $V$
    Update the local best and global best of the population
    Update particle velocity and position
endwhile
update the pool of data clusters $sp'$
Initialise $i = 1$
foreach data clusters $C$ in pool $sp'$ do
    $bcp^i$ ← train a set of base classifiers $\beta = \{\zeta^1, \zeta^2, \dots, \zeta^{\varsigma}\}$ on the data
    cluster $C$ and add to the pool
endfor
Initialise a population of $|bcp|$ particles
while termination criteria
    Map each particle to a trained base classifier in the population by
    generating a binary bit string pop representing each classifier in the
    population
    Generate an ensemble solution $\xi$ of classifiers that have a higher value
    than the threshold $\theta$
    Calculate the fitness of the population using equation (7) with validation
```

Algorithm 2: Ensemble classifier generation using OCCM

```
    dataset $V$
    Update the local best and global best of the population
    Update particle velocity and position
endwhile
Use the new optimised pool $bcp'$ of classifiers and discard all other
  classifiers to predict class labels of the unseen dataset, the test set and
  calculate ensemble classification accuracy.
```

5.2 Experiments, results and analysis

5.2.1 Datasets

The following benchmark classification datasets, namely *Breast Cancer, Diabetic, E.coli, Haberman, Ionosphere, Iris, Liver, Segment, Sonar, Thyroid, Vehicle,* and *Wine,* from the UCI Machine Learning repository [84] were utilised to evaluate the performance of the proposed ensemble method. Details of these datasets are already mentioned in Table 2, in Section 3.2.

5.2.2 Experimental setup

This section details the experiments conducted to measure the effect of incorporating an evolutionary algorithm to generate an optimised ensemble classifier. The proposed ensemble framework with OCCM is implemented in MATLAB, and a 10-fold cross-validation is adopted to incorporate for randomness as in other similar works. The upper bounds of clusters K was set to $\sqrt[3]{n}$ where n is the number of samples in a dataset as in other similar works [29].

PSO is used as a black box optimisation toolbox to optimise the pool of data clusters and the pool of base classifiers. This is done by encoding each data cluster or base classifier as a particle in the PSO search space, with its position vector restricted to only 1s and 0s. The process of encoding base classifiers / data clusters as particles in the PSO search space is illustrated in Figure 7.

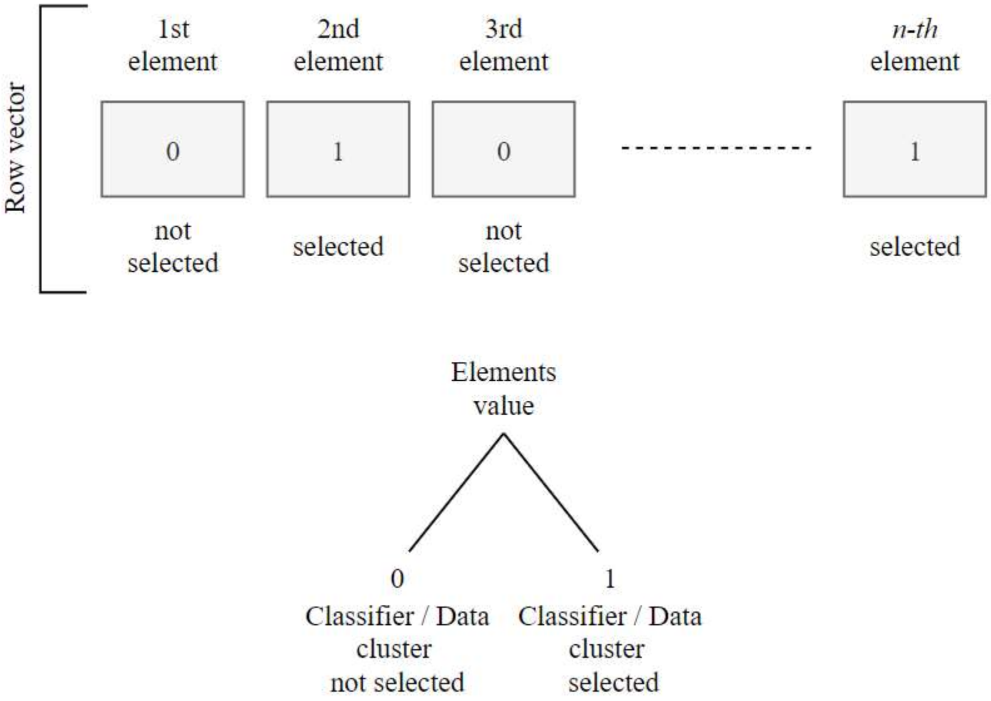


Figure 7: Encoding of data clusters or base classifiers as particles in PSO search space

5.2.3 Results

The average classification accuracy of the proposed ensemble framework with OCCM over 10-folds is reported in Table 6. Also given in Table 6 is the classification of the ensemble framework without the incorporation of optimisation. It can be noted that the average classification accuracy of the optimised ensemble classifier is 90.30%, whereas the average classification accuracy of the non-optimised ensemble is 85.39%. Thus, OCCM achieved an average of 5.75% performance gains over the original ensemble framework, adding to the efficacy of incorporating an optimisation algorithm.

Table 6: Classification performance of the proposed ensemble framework with and without OCCM

Datasets	**OCCM**	**Without OCCM**
Breast Cancer	**0.9677**	0.9040
Diabetic	**0.7789**	0.7670
E.coli	**0.9510**	0.9170
Haberman	**0.7670**	0.7450
Ionosphere	**0.9170**	0.9000
Iris	**0.9730**	0.7960
Liver	**0.7180**	0.7010
Segment	**0.9860**	0.9210
Sonar	**0.9190**	0.7840
Thyroid	**0.9640**	0.9620
Vehicle	**0.9030**	0.8980
Wine	**0.9920**	0.9520

Table 7: Effect of clustering on diversity of ensemble

Datasets	**Diversity without clustering**	**Diversity with clustering**
Adult	0.231	**0.287**
Australian	0.414	**0.475**
Balance	0.110	**0.348**
Banknote	0.353	**0.46**
Breast Cancer	0.021	**0.422**
E.coli	0.201	**0.353**
Haberman	0.272	**0.383**
Ionosphere	0.107	**0.383**
Iris	0.040	**0.398**
Liver	0.297	**0.411**
Page Block	0.099	**0.266**
Pima Diabetic	0.403	**0.463**
Segment	0.081	**0.408**
Sonar	0.261	**0.425**
Stat Image	0.199	**0.441**
Teaching	0.335	**0.450**
Thyroid	0.046	**0.163**
Vehicle	0.253	**0.465**
Vowel	**0.466**	0.400
WDBC	0.087	**0.321**
Wine	0.105	**0.454**

The effect of optimisation is also evaluated on the overall diversity of the ensemble. The optimisation process not only selects a subset of classifiers that can maximise the over classification accuracy of the ensemble, but also optimizes the input training space, $i.e.$ the random subspace of data clusters. To check the effect of clustering on diversity of the ensemble, a disagreement measure was computed using the ensemble with clustering and without. To compute the diversity, we calculated the average diversity of the pool of trained classifiers with and without clustering. The standard "*disagreement*" measure given in equation (6) is used and the results are given in Table 7.

We have also evaluated the effect of incorporating an evolutionary algorithm on reducing the number of clusters that are utilised by the proposed ensemble classifier. Since data clusters were generated using the raw data without any filtering incrementally, therefore, due to randomness and noise in the data, some data clusters are not suitable for training base classifiers. The results are given in Table 8.

Table 8: Effect of optimisation on discarding redundant and noisy data clusters from the pool

Datasets	**Average clusters generated**	**Average clusters utilised**
Breast Cancer	37	23.8
Diabetic	46	27.8
E.coli	22	11.7
Haberman	22	14.7
Ionosphere	29	17.8
Iris	16	10
Liver	29	19.5
Segment	79	33
Sonar	22	14.1
Thyroid	172	51.7
Vehicle	46	27.7
Wine	16	10.2

5.2.4 Discussion

It is evident from the results shown in Table 7, that the optimised ensemble achieved not only higher average diversity, but also an average performance improvement of 4.91%. This is because the proposed ensemble classifier filters redundant classifiers, which were not only adding to the computational complexity of the ensemble, but also affecting the overall classification accuracy of the ensemble to suffer from "*diminishing returns*". Additionally, it can be noted from Table 8 that, on average, a total of 44.66 clusters were generated, of which 21.83 clusters were utilised, which means an average of 15.83% cluster reduction was achieved.

5.2.5 Comparative analysis

The classification performance of the proposed ensemble method is also compared with other existing state-of-the-art ensemble methods that incorporated clustering OEC-ILC [79] and legacy ensemble methods such as Bagging [80], and Boosting [80]. The results are summarised in Figure 8.

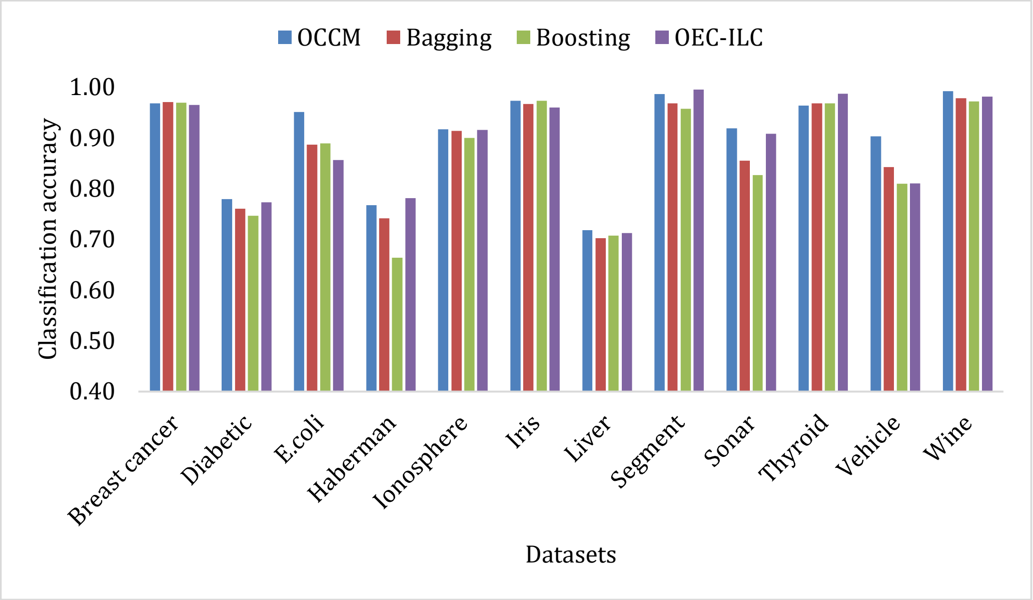


Figure 8: Performance of the proposed ensemble method OCCM in comparison with state-of-the-art ensemble classifiers

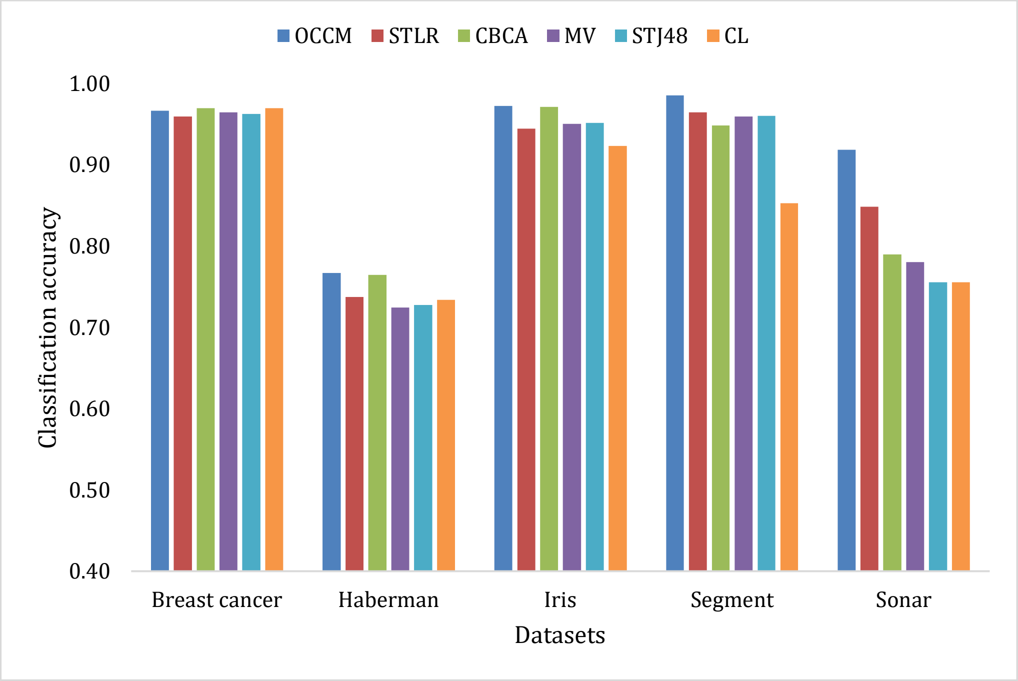


Figure 9: Performance of the proposed ensemble method OCCM in comparison with existing clustering-based state-of-the-art ensemble classifiers

It can be noted from Figure 8 that the proposed ensemble method achieved better classification performance than three existing state-of-the-art ensemble classifier approaches in seven of 12 datasets, achieving average performance gains of 2.3% over Bagging, 3.8% over Boosting, and 1.6% over OEC-ILC. The performance is also analysed in comparison with other clustering-based [94] ensemble methods, such as Majority Voting (MV), Stacking with Logistic Regression (STLR), Classification by Cluster Analysis (CBCA), standard classification with Clustering (CL), and Stacking with J48 as a combination function (STJ48). The results are summarised in Figure 9, and it can be noted that the proposed approach on average gained 3.47% performance improvement over STLR, 3.73% over CBCA, 5.24% over MV, 5.77% over STJ48 and 8.85% over CL. The proposed method performed significantly better in comparison to Bagging, Boosting, STLR, MV, STJ48, and CL rejecting the null hypothesis, and the results are inconclusive in comparison with OEC-ILC, and CBCA. The p-values are given in Table 9.

Table 9: p-values of Wilcoxon signed rank tests of OCCM

Methods	*p*-values
OEC-ILC	0.119
Bagging	**0.003**
Boosting	**0.004**
STLR	**0.021**
CBCA	0.112
MV	**0.021**
STJ48	**0.021**
CL	**0.039**

5.3 Ensemble with feature selection and optimisation method

5.3.1 The proposed method

Most of the ensemble approaches either exploit the input sample space to generate a random subspace or input feature space, but not both. Ensemble classifier methods that exploit both input features and input samples are considered hybrid ensemble methods. Therefore, we further extended the proposed ensemble method OCCM by proposing an ensemble classifier method FSOM [95] [3], that not only

[3] The work presented in this section has been published in the following paper: Z. Jan and B. Verma, "Ensemble Classifier Optimisation by Reducing Input Features and Base Classifiers," in Proceedings of the Congress on Evolutionary Computation, 2019, pp. 1580-1587.

exploits the samples space, but also exploits the input features. The intuition behind this is that not all input features should be incorporated in the training of base classifiers, since certain features are not particularly useful. For example, in house price prediction scenarios the colour of a house would be considered as a rogue feature which should be removed during the training process, otherwise any base classifiers trained will accommodate for the colour categorial values as well, impacting the final accuracy. This is particularly important in small datasets with large input features, and any insignificant feature(s) should be discarded for the ensemble to perform up to par.

Therefore to identify the significant features, a Neighbourhood Component Analysis (NCA) [96] is performed. This is done to calculate the relative weights w of the input features. The objective here is to find a vector w that contains a subset of input features that will maximise the following function:

$$argmax\left(\sum_{i=1}^{n} P\left(y'_i \middle| x_i\right)\right), \tag{19}$$

where $P(y'_i|x_i)$ is the posterior probability of correctly classifying x to generate y' and is defined as:

$$P\left(y'_i \middle| x_i\right) = \begin{cases} 1, & y'_i = y_i \\ 0, & y'_i \neq y_i \end{cases}, \tag{20}$$

y' are the predicted class labels obtained by training a base classifier ζ, in this case an NCA on the reduced input feature dataset and checking accuracy on validation dataset. The relative threshold l is calculated based on the feature weights given as:

$$l = \sum_{j=1}^{n} w_d I\left\{y'_j \neq y_j\right\}, \tag{21}$$

where $I\{x\}$ is an indicator function, and any feature having w less than l is discarded and the reduced dataset X' is then used to train the ensemble.

5.3.2 Ensemble classifier generation using FSOM

The ensemble classifier framework from Section 3.1 is utilised with FSOM to generate an optimised ensemble with reduced input features. In the proposed ensemble framework FSOM is identified in Figure 2. A stepwise algorithm of ensemble classifier generation using FSOM is given in Algorithm 3.

Algorithm 3: Ensemble classifier generation using FSOM

```
Input: Training dataset $X$, Validation dataset $V$, Testing data set $T$, Set of base
       classifiers $\zeta$, Upper bounds of clustering $K$
Output: Feature selection and optimisation-based ensemble classifiers

Initialize a random feature vector of weights of features
while termination criteria
    Train an NCA base classifier on the reduced feature data
    Compute the posterior probabilities
    Compute the loss threshold
    Update the feature weights
    Identify feature subset
endwhile
Initialise $k$ = 1, and $i = 1$
for $k = 1$ to $K$ do
    $sp^i$ ← partition training dataset into $k$ clusters by minimising the squared
        Euclidean distance of each sample from the cluster centroid
    Increment $i$
endfor
while termination criteria
    Map each particle to a cluster in the search space by generating a binary
        bit string pop representing each data cluster in the population
    Calculate the fitness of the population using equation (11) by training a
        base classifier on all data clusters in the population on validation dataset $V$
    Update the local best and global best of the population
    Update particle velocity and position
endwhile
update the pool of data clusters $sp'$
Initialise $i = 1$
foreach data clusters $C$ in pool $sp'$ do
    $bcp^i$ ← train a set of base classifiers $\beta = \{\zeta^1, \zeta^2, \ldots, \zeta^n\}$ on the data
        cluster $C$ and add to the pool
increment $i$
endfor
```

Algorithm 3: Ensemble classifier generation using FSOM

```
Initialise a population of $|bcp|$ particles
while termination criteria
    Map each particle to a trained base classifier in the population by
    generating a binary bit string pop representing each classifier in the
    population
    Generate an ensemble solution $\xi$ of classifiers that have a higher value than
    the threshold $\theta$
    Calculate the fitness of the population using equation (7) with validation
    dataset V
    Update the local best and global best of the population
    Update particle velocity and position
endwhile
Use the new optimised pool $bcp'$ of classifiers and discard all other
classifiers to predict class labels of the unseen dataset, the test set and
calculate ensemble classification accuracy.
```

5.4 Experiments, results and analysis

5.4.1 Datasets

The following *Breast Cancer, Diabetic, E.coli, Haberman, Ionosphere, Iris, Liver, Segment, Sonar, Thyroid, Vehicle,* and *Wine* benchmark classification datasets from the UCI Machine Learning repository [84] were utilised to evaluate the performance of the proposed ensemble method. Details of these datasets are already mentioned in Table 2, in Section 3.2.

5.4.2 Experimental setup

The proposed ensemble framework with FSOM is implemented in MATLAB, and a 10-fold cross-validation is adopted to incorporate for randomness as in other similar works. The setup is mostly like that in Section 5.2.2, besides the addition of NCA to identify significant features. Default implementation of NCA function "*fscnca*" in MATLAB was used. *Fscnca* learns feature weights through a diagonal adaptation of NCA with regularisation parameter. The features are selected based on their relative weights and any feature having w less than the threshold l is discarded.

5.4.3 Results

The average classification accuracy over 10-cross validated folds of FSOM with and without the incorporation of feature reduction is given in Table 10.

Table 10: Classification accuracy of the ensemble with and without FSOM

Datasets	FSOM	Without FSOM
Breast cancer	**0.9676**	0.9657
Diabetic	0.7686	**0.7786**
E.coli	**0.9492**	0.8544
Haberman	0.7504	**0.7516**
Ionosphere	**0.9215**	0.9201
Iris	**0.9778**	0.9533
Liver	0.7167	**0.7271**
Segment	**0.9918**	0.9597
Sonar	**0.8443**	0.8369
Thyroid	**0.9769**	0.9499
Vehicle	**0.9017**	0.8015
Wine	0.9925	**0.9944**

It can be noted from Table 10 that the proposed hybrid ensemble classifier achieved performance gains of an average of 2.21% compared to the ensemble generated without feature reduction; also, the proposed ensemble classifier with feature reduction outperformed the ensemble classifier without feature reduction in eight of 12 datasets.

Table 11: Effect of feature reduction and optimization on ensemble component size

Datasets	FSOM	Proposed method without optimization	Original size	Optimized size
Breast cancer	**0.9676**	0.9671	17	8
Diabetic	**0.7686**	0.7634	178	83
E.coli	**0.9492**	0.9234	39	16
Haberman	**0.7504**	0.7470	205	101
Ionosphere	**0.9215**	0.9070	84	40
Iris	0.9778	0.9778	14	7
Liver	**0.7167**	0.7043	208	99
Segment	**0.9918**	0.9896	6	3
Sonar	**0.8443**	0.8381	59	28
Thyroid	**0.9769**	0.9686	13	4
Vehicle	**0.9017**	0.8950	490	238
Wine	**0.9925**	0.9962	14	8

The effect of optimisation on reducing ensemble component size is also analysed and it can be noted from Table 11 that on average the optimised ensemble classifier achieved 89.65% classification accuracy, and the non-optimised ensemble achieved an average classification accuracy of 88.97%.

5.4.4 Discussion

The results in Table 10 support the assertion that not all features in a dataset are significant and should not to be used for the training of base classifiers. This is especially true for datasets that have lower numbers of samples and high numbers of features, as insignificant features can throw a base classifier off and such a base classifier should be discarded. This helps to mitigate the curse of dimensionality and hybrid methods are particularly effective in datasets of higher dimensions.

Results from Table 11 conclude that the optimised ensemble classifier gained 0.67% performance improvement over the non-optimised ensemble, which is very marginal; however, the average component size of ensembles without optimisation is 110, whereas the average component size of ensembles with optimisation is 52. Therefore, a 50% reduction is achieved whilst, if not increasing, at the least maintaining, the same classification accuracy. This certainly adds to the fact that more is not necessarily better and adding additional base classifiers than an optimal number will only add to the overall complexity of the ensemble.

5.4.5 Comparative analysis

The performance of the proposed ensemble classifier method is compared with existing legacy ensemble classifier methods such as Bagging [79], and Boosting [79], and a recent state-of-the-art clustering-based ensemble classifier method OEC-ILC proposed in [79]. The classification accuracies are taken directly from the respective paper. The results are summarised in Figure 10 and it can be noted that the proposed ensemble classifier out-performed existing ensemble methods in six of 12 datasets, achieving an average of 1.70% performance gain over Bagging, 3.13% performance gain over Boosting, and 0.94% performance gains over OEC-ILC. The performance is also analysed in comparison with other clustering-based [94] ensemble methods, such as MV, STLR, CBCA, CL, and STJ48, and the results are summarised in Figure 11. It can be noted that the proposed approach on average gained 1.66% performance improvement over STLR, 1.91% over CBCA, 3.40% over MV, 3.92% over STJ48 and 6.93% over CL.

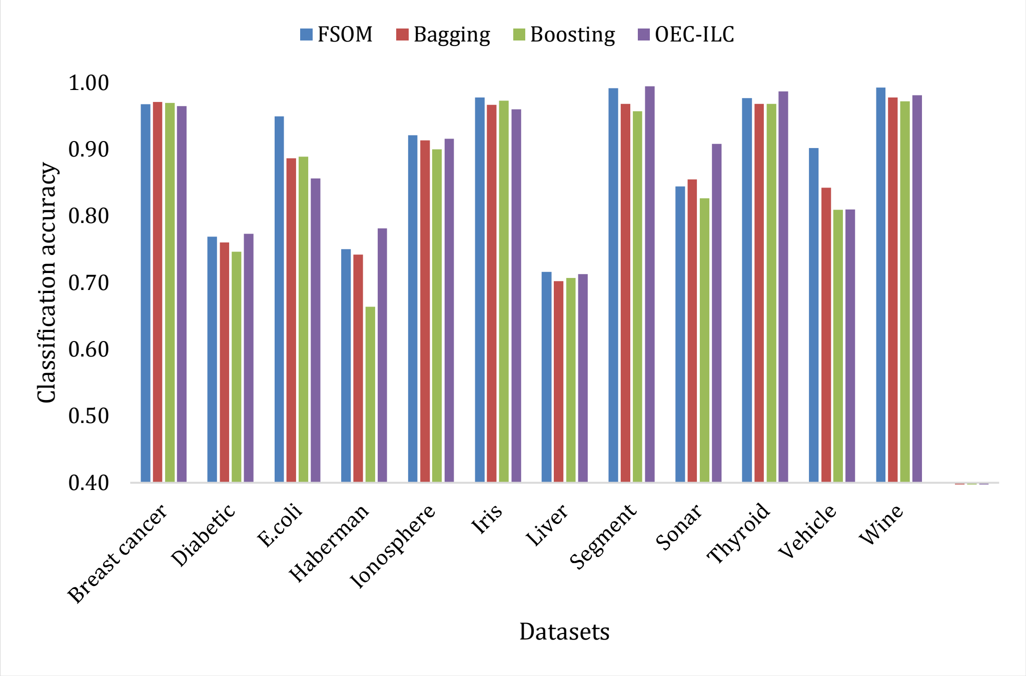


Figure 10: Performance of the proposed ensemble method FSOM in comparison with state-of-the-art ensemble classifiers

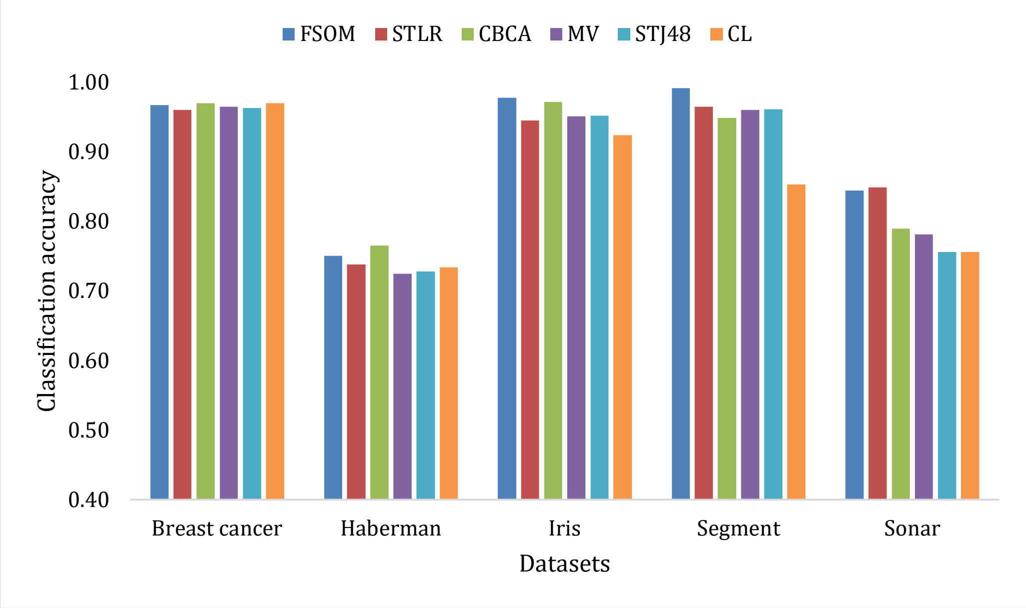


Figure 11: Performance of the proposed ensemble method FSOM in comparison with state-of-the-art clustering-based ensemble classifiers

The proposed method performed significantly better in comparison to Bagging, Boosting, STLR, MV, STJ48, and CL rejecting the null hypothesis, and the results are inconclusive in comparison with OEC-ILC, and CBCA. The p-values are given in Table 12.

Table 12: p-values of Wilcoxon signed rank tests of FSOM

Methods	***p*-values**
OEC-ILC	0.260
Bagging	**0.006**
Boosting	**0.001**
STLR	**0.039**
CBCA	0.172
MV	**0.021**
STJ48	**0.021**
CL	**0.039**

5.5 Summary

In this chapter an ensemble method is proposed that utilises an evolutionary algorithm to optimise the generated pool of data clusters, and the pool of trained base classifiers. The benefit of optimising the pool of data clusters is that the generated ensemble is independent of the value of *K* chosen, as any redundant or noisy data clusters are discarded through the incorporation of optimisation. Moreover, through optimising the pool of base classifiers, the proposed method discards not only redundant base classifiers, but also discards base classifiers that can negatively impact the performance of the ensemble. A detailed comparative analysis with existing state-of-the-art ensemble classifier methods, as well as existing clustering-based ensemble methods, was also provided and through experiments it was proven that the proposed ensemble method performed significantly better than most of the other ensemble methods. Furthermore, a hybrid ensemble method is also proposed that further extends the framework by identifying the significant features in a dataset and reduces the dimensions of data by utilising only the significant features. As before it utilises an evolutionary algorithm to optimise the generated pool of data clusters, and the pool of trained base classifiers as well. The method is particularly effective with small datasets with many features, as insignificant features will not be utilised for the training of base classifiers.

Chapter 6: **Class and Cluster Balancing Method**

This chapter proposes a novel ensemble method, namely Class and Cluster Balancing Method (CCBM), that utilises a cluster-balancing strategy to generate class balanced data clusters. The proposed ensemble method mitigates the problem of class imbalances, not just in data but also in data clusters (since clustering algorithms work independent of the data classes) by first generating class-pure datasets and then adding samples to each class-pure data cluster from other classes that closet to its centroid. The proposed ensemble method is evaluated on several benchmark datasets, and results are compared with existing state-of-the-art ensemble methods.

6.1 Ensemble with class and cluster balancing method

6.1.1 The proposed method

One limitation of generating a random subspace through clustering is class imbalances. Data class is basically the response variable y in a structured dataset with a feature vector of x (also known as the predictor variables). Since clustering algorithms work independently of the data classes that exist, the resultant data clusters may or may not have equal numbers of samples from different data classes. Any base classifier that is trained on a data cluster that has more samples from one class than others will be biased towards the majority class. Additionally, there is no guarantee that a data cluster will contain samples from all the data classes and any data cluster that does not contain a sample from a data cluster can be detrimental to the training of a base classifier. If a base classifier is trained on a data cluster that is missing samples from a data class, then that base classifier will negatively impact the final ensemble generalisation performance. Therefore, to tackle these limitations, we proposed an ensemble classifier method, namely Class and Cluster Balancing Method (CCBM)[4], that not only generates a random subspace through clustering, but

[4] The works presented in this chapter have been published in:
1) Z. Jan, and B. Verma," Multi-Cluster Class Balanced Ensemble", IEEE Transactions on Neural Networks and Learning Systems, 2020. (Accepted on: 6th march)
2) Z. Jan, and B. Verma," Balanced Learning with Ensemble of Convolutional Neural Networks for Image Classification", IEEE Symposium Series on Computational Intelligence, 2019, pp 2418 – 2424.

also balances all generated data clusters using the proposed cluster balancing methodology. The proposed method not only alleviates the problem of class imbalances within the data clusters, but also ensures that any class imbalances within the data does not affect the generated random subspace. We proposed three ensemble classifiers methods using this strategy, with different choices of base classifiers. We tested the proposed approach not only on UCI benchmark datasets, but also on benchmark image datasets. Since for classifying image datasets a more suitable option is the use of CNNs, , we generated a deep ensemble as well. We first discuss the general architecture of the proposed ensemble classifier method, and then detail the performance of same ensemble with different base classifiers on various datasets.

The proposed ensemble classifier method starts by partitioning data samples based on their respective classes. For example, if there are κ data classes in the dataset then there will be κ subsets of the training data X. As such the class-pure subsets are $X^1, X^2, \dots X^\kappa \subset X$ then the following rules hold $X^1 \cap X^n = \emptyset, \dots X^2 \cap X^\kappa = \emptyset$ and $X^1 \cup X^2 \cup \dots X^\kappa = X$, where κ is the number of data classes and X is the universal set which in this case is the original input training data. Each subset X^i contains samples belonging to a single class in κ. Each subset X^i having a total of n' number of samples (n is the total number of samples in the dataset and $n' \ll n$), is utilized to generate data clusters C which are also a subset of X^i defined as $C = \{(x_1, y_1), \dots, (x_m, y_m)\}$ where $m \ll n'$. The data that is utilised for clustering is stripped off its classes (y), meaning that only the feature vector is utilised in clustering, so that the clustering algorithm is not reflective of the input classes. Moreover, clusters are generated utilising the subsets of training data that belonging to a single class. Therefore, all generated data clusters contain data samples belonging to a single class. These data clusters are class-pure data clusters because they contain samples belonging to a single class. For each data class at least N data clusters are generated. Various values of N have been tested and results are presented in later sections.

Therefore, to generate N data clusters of each class, firstly class labels are identified and a row vector representing each class label is generated as follows:

3) Z Jan, and B. Verma," Ensemble Classifier Generation Using Class-Pure Cluster Balancing", International Conference on Neural Information Processing, 2019, pp 761 – 769.

$$r = \{N^1, N^2, \ldots, N^\kappa\}, \tag{22}$$

where κ is the number of unique class labels in the dataset and N^i is the number of data clusters required for class i. The process starts partitioning the input data into k data clusters in each iteration. k is initially set to 2 and is increased iteratively until the criteria is satisfied. In each iteration data clusters are checked for dominant class and N for the respective dominant class is decremented. The process is repeated until there are no non-zero Ns left in the row vector s.

For a base classifier to be effectively trained, it must be exposed to all class labels in the data set. This gives a base classifier the ability to classify unseen data successively without any bias towards a class. Therefore, each class-pure data cluster must be balanced before it is utilised to train a base classifier. In order to balance a strong data cluster, data samples from different classes are added to the cluster that are closest to the cluster centroid. Assuming $k = 2$, data clusters are generated from the data subsample X^i then the following holds $C^1 \cap C^2 = \emptyset$ and $C^1 \cup C^2 = X^i$ where C is a data cluster. Both C^1 and C^2 now must be balanced before they can be utilised in the training process. If data cluster C^1 has centroid c^1 and has m data samples belonging to one of the classes in κ, let us say κ', then data samples from classes besides κ' are added to the cluster. In order to do this, first normalised Euclidean distance from centroid $c^1$of each sample is computed as follows:

$$dist = norm(x_i - c) \ \ \forall \ i \in V \, and \, i \ \notin \kappa^{'} , \tag{23}$$

where $norm(x_i - c)$ is the Euclidean norm (also called the vector magnitude) given as $v = \sqrt{\sum_{i=1}^{n} |v_i|^2}$. Then m data samples that are closest to the centroid c^1 are added to cluster C^1. This process is repeated until at most m samples from each class besides κ' are added to the cluster C^1. The same is repeated for every cluster in the pool, until all generated class-pure data clusters are now balanced data clusters.

6.1.2 Ensemble classifier generation using CCBM

The ensemble classifier framework from Section 3.1 is utilized with CCBM to generate an ensemble trained on class-balanced data clusters. Essentially the main purpose of the balancing methodology is to have a pool of data clusters where each cluster has, ideally, an equal number of samples from all data classes from the dataset. All balanced data clusters are utilised to train base classifiers to generate the base classifier pool bcp. which are then combined to generate the ensemble as in the original ensemble framework. The algorithm for training an ensemble using CCBM is given in Algorithm 4.

Algorithm 4: Ensemble classifier generation using CCBM

Input: Training dataset X, Validation dataset V, Testing data set T, Set of base classifiers ζ, Number of clusters per class N

Output: Class and cluster balanced ensemble classifier

```
Generate $X^n$ subsets of the input data such that $X^1, X^2, \dots X^\kappa \subset X$
Initialise $i$, and row vector $r = \{N^1, N^2, \dots, N^\kappa\}$
foreach $X^n$ do
    $sp^i$ ← partition $X^n$into $N$ data clusters by minimizing the squared
    Euclidean distance of each sample from the cluster centroid
    Decrement respective $N^i$
endfor
foreach class-pure cluster $C$ in $sp$ do
    determine the dominant class $y_d$in cluster
    $y_{nd}$ ← determine nondominant classes in the cluster
    foreach $y_{nd}$ do:
        add feature vectors $x$ from remaining clusters in the pool closest to
        cluster centroid
    endfor
endfor
foreach data clusters $C$ in pool $sp$ do
    $bcp^i$ ← train a base classifier $\zeta$ on the data cluster $C$ and add to
    the pool
endfor
Use the pool $bcp$ of classifiers to predict class labels of the unseen dataset,
the test set and calculate ensemble classification accuracy.
```

6.2 Experiments, results and analysis

6.2.1 Datasets

The following benchmark classification datasets, namely *Appendicitis, Balance, Bank note, Breast cancer, Bupa, Diabetic, Fertility, Glass, Haberman, Hayes-roth, Heart, Hepatitis, Iris, Plan relax, Sonar, Segment, Spectfheart, Statimag, Teaching, Thyroid, Vehicle, WDBC, Wine,* and *Zoo,* from the UCI Machine Learning repository [84], were utilised to evaluate the performance of the proposed ensemble method. Details of these datasets are already mentioned in Table 2. Also, the two benchmark image datasets namely MNIST, and CIFAR-10 have been used in experimentations, which are mentioned in Section 4.2.1.

6.2.2 Experimental setup

The proposed ensemble framework with CCBM is implemented in MATLAB and classification accuracies are averaged over 30-independent runs, with each run having 10-fold cross validation. Default implementation of base classifiers was used. The architecture of CNN used for MNIST classification includes a combination of convolution layer, batch normalisation layer, max pooling layer, RELU layer, fully connected layer, and a SoftMax classification layer. An initial learning rate of 0.001, with maximum epochs set to 100 and a sigmoid solver. The architecture of CNN for CIFAR-10 is a slight variation of the one used for MNIST, with the addition of batch normalisation layer and average pooling layer. For MNIST dataset the CNN proposed in [97] in MathWorks deep learning toolbox user guide is used as a base classifier, and for CIFAR-10 dataset the CNN proposed in [98] is used as a base classifier. For further details of architecture readers can refer to the respective papers.

For clustering the image datasets, images were flattened into two dimensions and analysis was conducted on normalised pixel values. This is a necessary pre-processing step, as data clusters are generated using 2d Euclidean distance measurement. Furthermore, clusters are balanced using Euclidean norm in a 2d cartesian coordinate system. Before being fed to a CNN base classifier for training, the image was converted back into its original three-dimensional form. A multitude of CNNs are trained on all balanced data clusters and are added to the pool to generate the base classifier pool. The entire pool of classifiers is utilised to classify the unseen test images and class decisions of all classifiers are fused through majority voting to generate the class decisions of the ensemble. For fair comparisons, any pre-processing steps applied before feeding the data to the ensemble were the same for the base CNNs. The proposed ensemble was also tested on Kaggle.com varied MNIST classification competition [99], which has 42,000 training images and 28,000 test images; the remaining features and dimensions *etc.* are the same as the original MNIST data

6.2.3 Results

This section comprises the results and analysis of the proposed ensemble classifier framework using CCBM on benchmark classification datasets, and benchmark image datasets. The classification accuracies of the proposed ensemble classifier approach were computed on different values of N and the highest classification accuracy achieved with each dataset for a given value of N is listed in Table 13 below.

Table 13: Classification performance of the proposed ensemble framework with and without CCBM

Dataset	**CCBM**	**Std. dev.**	**# of clusters per class**
Appendicitis	0.8732	0.005	01
Balance	0.9162	0.006	13
Bank note	0.9993	0.000	02
Breast cancer	0.9706	0.002	10
Bupa	0.6928	0.010	18
Diabetic	0.7727	0.005	07
Fertility	0.8818	0.002	12
Glass	0.9569	0.015	01
Haberman	0.7426	0.007	01
Hayes-roth	0.6765	0.022	17
Heart	0.8412	0.009	18
Hepatitis	0.8574	0.020	10
Iris	0.9509	0.007	20
Plan relax	0.7313	0.000	02
Sonar	0.9504	0.003	04
Segment	0.7647	0.011	01
Spectfheart	0.8031	0.015	20
Statimag	0.8688	0.002	05
Teaching	0.5251	0.023	01
Thyroid	0.9630	0.001	18
Vehicle	0.7963	0.008	01
WDBC	0.9820	0.001	03
Wine	0.9781	0.004	02
Zoo	0.9623	0.000	02

For benchmark image datasets, the base accuracy of the stand-alone CNNs used in the example, and the classification accuracy of the proposed ensemble of deep learners, is given in Table 14. As for the varied MNIST dataset from Kaggle the proposed ensemble classifier was able to achieve classification accuracy of 99.21%. The results are posted publicly on Kaggle website.

Table 14: Classification accuracy of base CNNs and the proposed ensemble of CNNs on MNIST and CIFAR-10 datasets

	Test classification accuracy	
Classifiers	***MNIST***	***CIFAR***
Base CNN	99.24%	85.48%
CCBM with CNN	100.00%	87.11%

6.2.4 Discussion

It can be noted from Table 13 that the number of clusters for each data class is not the same for different datasets. This is because different datasets have different spatial characteristics: some datasets have dense local regions whereas others have sparsely distributed regions. N checks for sparseness in data patterns belonging to a similar class. For example, if we separate a dataset based on two classes like cats and dogs, then N will measure how different cats are from each other and whether they should be put in different data clusters or not. From the experiments it is evident that some datasets are sparse, and some are dense, therefore instead of analytically calculating the optimum value of N, validation data can be used to experimentally find the optimum value of N which can maximise the performance of the ensemble classifier. Figure 12 shows the effect of varying the number of clusters per class that is varying N on classification accuracy of the generated ensemble classifier for each dataset. It can be noted from Figure 12 that changing the number of clusters generated per class does influence the overall ensemble classifier accuracy.

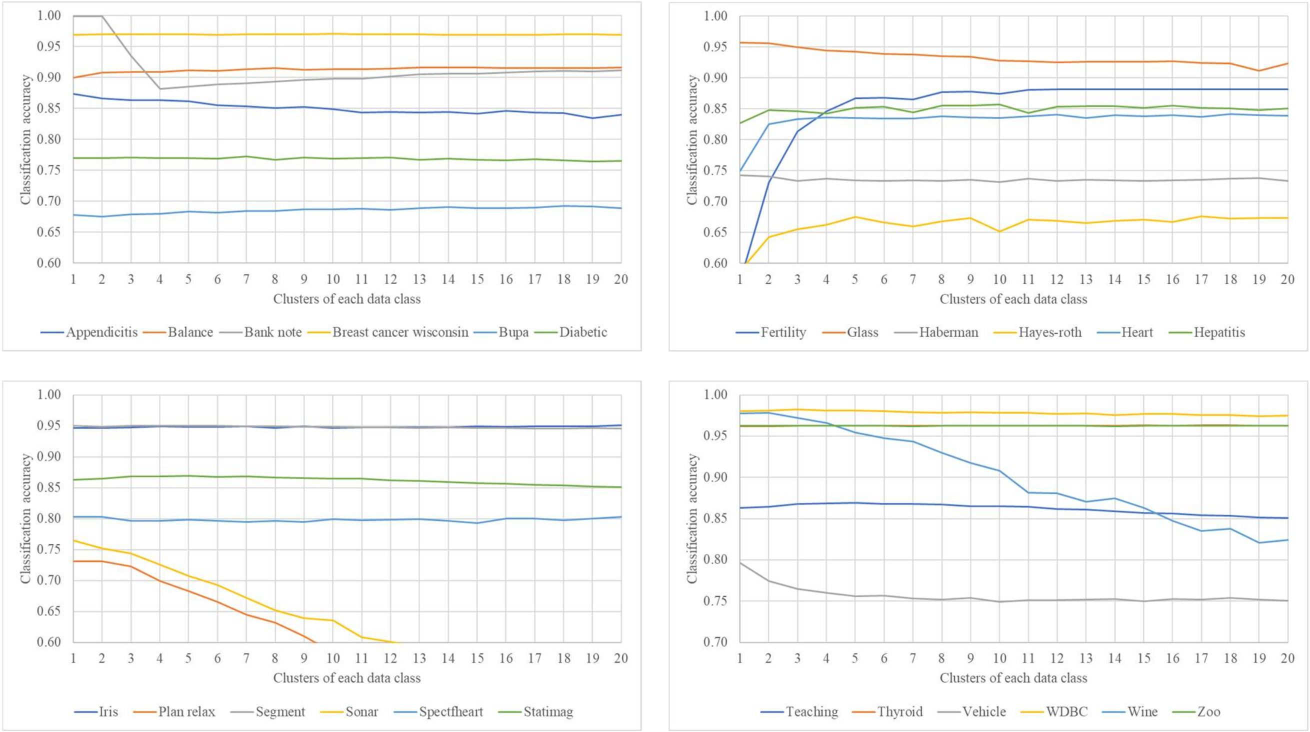


Figure 12: Effect of different values of *N* on classification accuracy of the proposed ensemble classifier

It is evident that in *Plan-relax* and *Sonar* datasets, the classification accuracy significantly dropped when more than three data clusters per class were generated. However, with H*eart* dataset the generalisation performance improved significantly when more than five data clusters per class were generated. On the other hand, with *Banknote*, *Breast Cancer*, and *Zoo* datasets, no significant change in the classification performance of the ensemble occurred with different values of *N*. This is because the decision boundary of the dataset is not particularly complex, and a rather simple classifier is able to achieve higher accuracy on the datasets, as evident by many existing works in research. It can also be noted that in the proposed, CCBM did not work with *Spectfheart* dataset. A further analysis was conducted to investigate the cause and is shown in Figure 13. It can be noted from Figure 13 that the *Spectfheart* dataset is condensed into two distinct regions, therefore the proposed method generated clusters using the data samples which were essentially outliers in the dataset, which in turn trained base classifiers which negatively affect the performance of the ensemble.

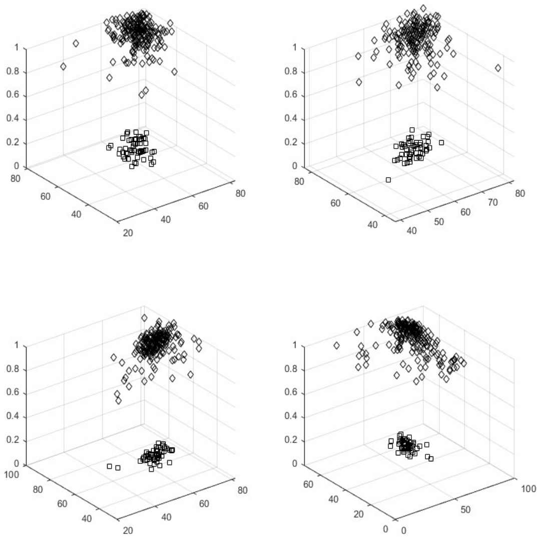
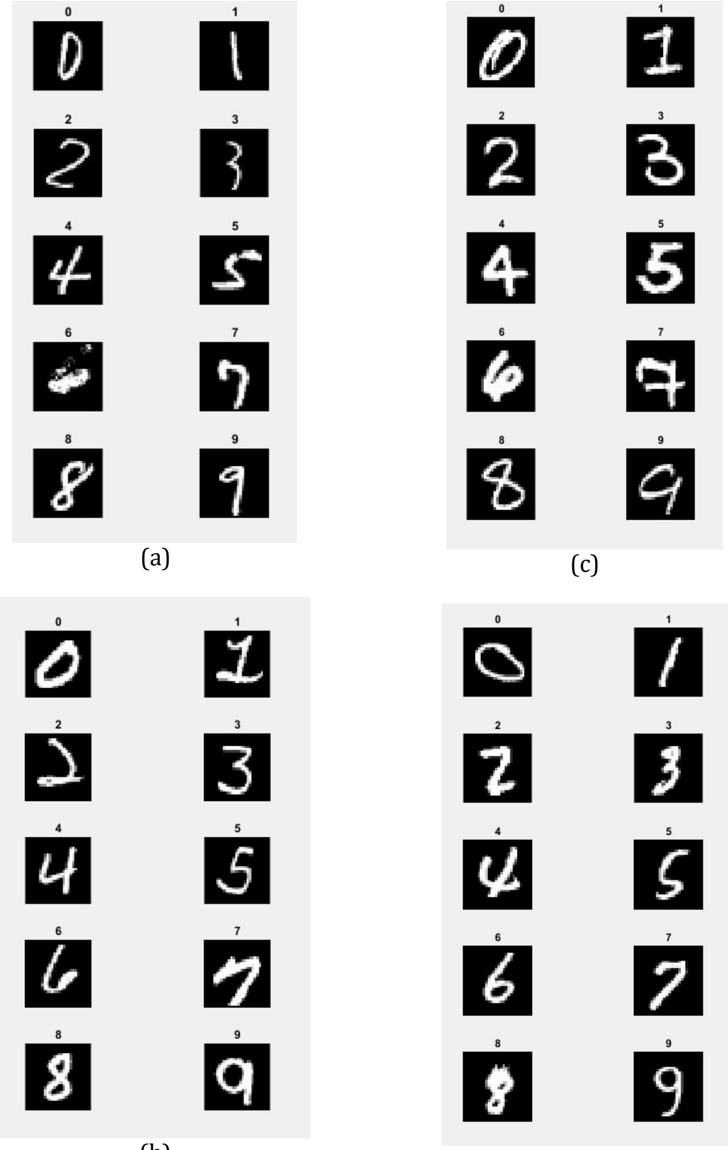


Figure 13: Clustering analysis of Spectfheart dataset

It can be concluded that the proposed method of class-based clustering and balancing will work or improve the ensemble classification accuracy for datasets that are sparse in nature, *i.e.* datasets in which samples of each class can be grouped into multiple similar groups. Therefore, instead of analytically testing when to use the proposed method, validation dataset can be used and the value of N can be tweaked to maximise the performance of the ensemble.

Figures 14, and 15 show the effect of cluster analysis conducted on benchmark image datasets, namely the MNIST and CIFAR-10 datasets. It can be noted from Figure 14 that there is some level of similarity in the images that are grouped together in similar clusters based on Euclidean distances. For example, in Figure 14(d) most of the letters are slightly italicised, whereas in Figure 14(c) it can be noted that most of the letters are following a Neuton font style; similarly, Figure 13(b) is a slight variation of Arvo and Figure 13(a) is more of a free style.



(a)

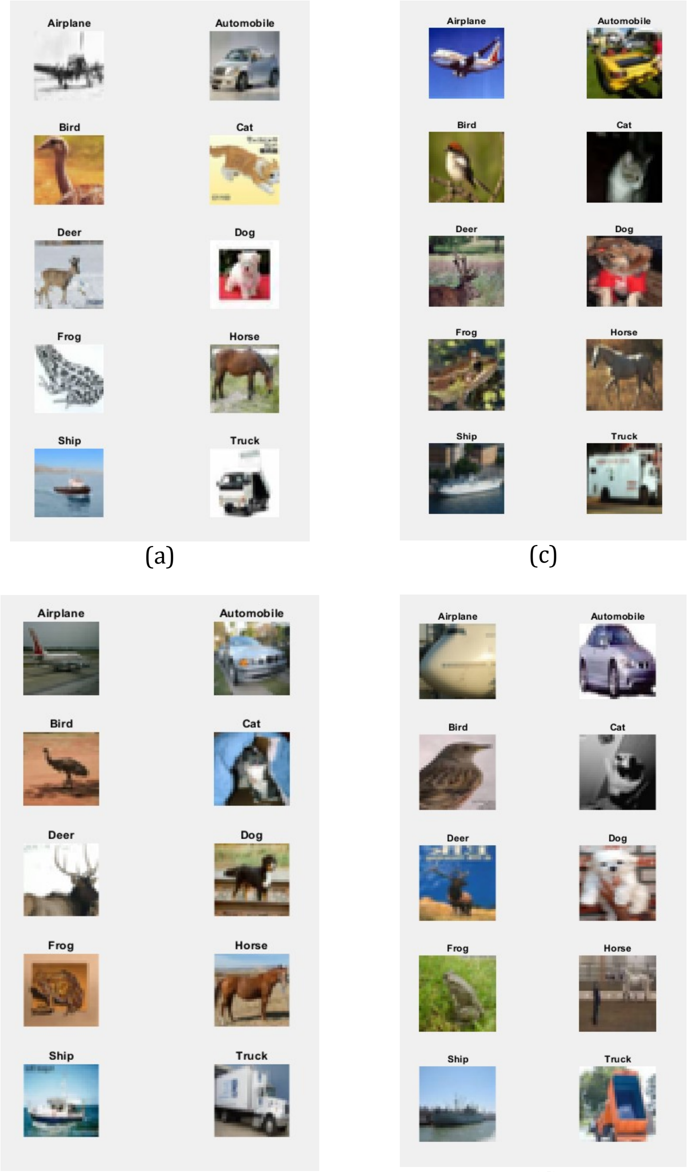
(c)

(b)

(d)

Figure 14: Different images in 4 different balanced data clusters of MNIST dataset that are closest to each other based on Euclidean distance

Furthermore, Figure 15 shows the cluster analysis conducted on CIFAR – 10 dataset. As stated, each balanced cluster will contain images from all different classes in the dataset so that a CNN trained on a balance cluster can classify all patterns. It can be noted from Figure 15 that the images have a certain level of similarity, not based on shape of the image as in MNIST dataset, but on colour saturation. This is particularly prominent in Figures 15(a) and 15(b). Images in 15(a) follow a gradient towards blue and white, whereas, in 15(b) the colour gradients are towards blue (cause of sky) and yellow (cause of desert).



(a) (c)

(b) (d)

Figure 15: Different images in 4 different balanced data clusters of CIFAR-10 dataset that are closest to each other based on Euclidean distance

6.2.5 Comparative analysis

Firstly, it is assessed whether generating class balanced data clusters is having a positive effect on overall ensemble classifier accuracy, and whether a selection of base classifier has any effect or not. To compute this experiment is conducted using the proposed ensemble framework with CCBM using different base classifiers, and ensemble framework without CCBM, the classification accuracies over 30 independent runs with 10-fold cross validation are given in Table 14.

Table 15: Effect of class-based cluster balancing on overall ensemble classifier accuracy

Dataset	CCBM			
	SVM		Naïve Bayes	
	With cluster balancing	**Without cluster balancing**	**With cluster balancing**	**Without cluster balancing**
Appendicitis	**0.8732**	**0.8732**	0.8538	0.8256
Balance	**0.9162**	0.9153	0.8990	0.8451
Bank note	**0.9993**	0.8120	0.9503	0.9376
Breast cancer	**0.9706**	0.9603	0.9624	0.9591
Bupa	0.6900	**0.6928**	0.6690	0.6652
Diabetic	**0.7727**	0.7696	0.7566	0.7566
Fertility	0.8818	0.8595	**0.8824**	0.7089
Glass	**0.9569**	0.9089	0.8717	0.5627
Haberman	**0.7426**	0.7365	0.7398	0.6776
Hayes-roth	**0.6765**	0.5919	0.5998	0.5848
Heart	**0.8412**	0.8356	0.8041	0.7841
Hepatitis	0.8522	0.8574	0.8667	**0.8782**
Iris	0.9509	0.6791	**0.9667**	0.9667
Plan relax	0.7268	**0.7313**	0.7086	0.5848
Sonar	**0.9504**	0.6912	0.9158	0.6240
Segment	0.7644	**0.7647**	0.7489	0.7330
Spectfheart	0.5490	0.8031	0.7019	**0.8114**
Statimag	0.8620	**0.8688**	0.8316	0.8019
Teaching	**0.5251**	0.5043	0.5045	0.4845
Thyroid	**0.9630**	0.6843	0.9386	0.8197
Vehicle	**0.7963**	0.7881	0.6450	0.6172
WDBC	**0.9820**	0.9705	0.9593	0.9413
Wine	0.9750	**0.9781**	0.9578	0.9678
Zoo	**0.9624**	0.7076	0.8269	0.4187

It can be noted from Table 14 that in most cases the proposed ensemble classifier with CCBM achieved higher classification accuracy than the ensemble classifier without CCBM, irrespective of the choice of base classifier. Moreover, it can also be noted that the ensemble with SVM and ensemble with Naïve Bayes both with CCBM improved in their generalization ability, although, CCBM with SVM performed better indicating that SVM is a more robust base classifier than NB. This is further investigated with different choice of base classifiers, and the results are given in Table 15 to identify any base classifier preference.

Table 16: Proposed ensemble classifiers performance with different base classifiers

	CCBM			
Dataset	**SVM**	**Naïve Bayes**	**LDA**	**KNN**
Appendicitis	**0.873**	0.854	0.862	0.852
Balance	**0.916**	0.899	0.887	0.864
Bank note	**0.999**	0.953	0.950	0.993
Breast cancer	**0.971**	0.962	**0.971**	0.946
Bupa	**0.690**	0.669	0.680	0.620
Diabetic	**0.773**	0.756	0.762	0.757
Fertility	0.882	**0.882**	0.813	0.857
Glass	**0.957**	0.871	0.847	0.737
Haberman	0.743	0.739	**0.766**	0.734
Hayes-roth	**0.677**	0.599	0.651	0.628
Heart	0.841	0.804	**0.844**	0.805
Hepatitis	0.852	**0.866**	0.839	0.83
Iris	0.951	**0.966**	0.965	0.964
Plan relax	**0.727**	0.708	0.645	0.718
Sonar	**0.950**	0.915	0.860	0.939
Segment	0.764	0.748	**0.768**	0.752
Spectfheart	0.549	**0.701**	0.652	0.568
Statimag	**0.862**	0.831	0.773	0.804
Teaching	**0.525**	0.504	0.510	0.446
Thyroid	**0.963**	0.938	0.829	0.396
Vehicle	**0.796**	0.645	0.489	0.482
WDBC	**0.982**	0.959	0.953	0.933
Wine	**0.975**	0.957	0.932	0.839
Zoo	**0.962**	0.827	0.520	0.814

It can be noted from Table 15 that in 17 out of 24 datasets, the proposed ensemble classifier using CCBM with SVM as a base classifier out-performed others. On average CCBM with SVM as a base classifier achieved average performance gains of 2.6% over CCBM with Naïve Bayes, 5.9% over CCBM with LDA, and 7.9% over CCBM with KNN. This adds to the fact that the best performing base classifier is SVM in comparison to others, and the ensemble generated with SVM as a base classifier achieved the best classification performance.

The classification performance of ensemble with CCBM is also compared with existing legacy ensemble classifier approaches, namely Boosting and Random Forest. For fair comparisons, default implementation of Random Forest was used in MATLAB, using the classifier "*treeBagger*" with a maximum of 50 trees. The average classification accuracy over 30 independent runs was compiled; for Boosting "*lpBoost*" was used since we have datasets with multiple classes. The highest classification accuracies are given in bold in Table 16. It can be noted that CCBM out-performed Random Forest and Boosting in 14 out of 24 datasets.

Table 17: CCBM in comparison with existing state-of-the-art ensemble approaches

	CCBM					
Dataset	**SVM**	**Naïve Bayes**	**LDA**	**KNN**	**Boosting**	**Random Forest**
Appendicitis	**0.873**	0.854	0.862	0.852	0.855	0.871
Balance	**0.916**	0.899	0.887	0.864	0.830	0.843
Bank note	**0.999**	0.950	0.950	0.993	0.997	0.993
Breast cancer	0.971	0.962	**0.971**	0.946	0.962	0.969
Bupa	0.690	0.669	0.680	0.620	0.688	**0.722**
Diabetic	**0.773**	0.757	0.762	0.757	0.714	0.760
Fertility	0.882	**0.882**	0.813	0.857	0.815	0.872
Glass	0.957	0.872	0.847	0.737	0.355	**0.981**
Haberman	0.743	0.740	**0.766**	0.734	0.666	0.679
Hayes-roth	0.677	0.600	0.651	0.628	**0.841**	0.808
Heart	0.841	0.804	**0.844**	0.805	0.781	0.828
Hepatitis	0.852	**0.867**	0.839	0.830	0.838	0.866
Iris	0.951	**0.967**	0.965	0.964	0.333	0.948
Plan relax	**0.727**	0.709	0.645	0.718	0.607	0.692
Sonar	0.950	0.916	0.860	0.939	0.979	**0.980**
Segment	0.764	0.749	0.768	0.752	**0.840**	0.835
Spectfheart	0.549	0.702	0.652	0.568	0.799	**0.812**
Statimag	0.862	0.832	0.773	0.804	0.836	**0.918**
Teaching	0.525	0.505	0.510	0.446	0.649	**0.671**
Thyroid	0.963	0.939	0.829	0.396	0.980	**0.997**
Vehicle	**0.796**	0.645	0.489	0.482	0.741	0.751
WDBC	**0.982**	0.959	0.953	0.933	0.952	0.964
Wine	0.975	0.958	0.932	0.839	0.571	**0.978**
Zoo	**0.962**	0.827	0.520	0.814	0.406	0.961

A further analysis of CCBM is conducted in comparison with state-of-the-art ensemble classifier methods such as MPRaF-T [34], clustering-based ensemble classifiers [94], a hybrid ensemble classifier, namely Progressive Semi-supervised Ensemble Learning (PSEMISEL) proposed in [62], and an ensemble classifier that uses a Weighted Majority Voting (WMV) proposed in [92]. The classification accuracies are taken directly from their respective papers, and the results are given in Tables 17 and 18 with highest classification accuracies given in bold.

Table 18: CBCLEL in comparison with existing state-of-the-art ensemble classifiers

	CCBM				**Published state-of-the-art ensemble classifier methods**		
Dataset	**SVM**	**Naïve Bayes**	**LDA**	**KNN**	**MPRaF-T [34]**	**PSEMISEL [62]**	**WMV [92]**
Balance	**0.916**	0.899	0.887	0.864	0.890		
Bank note	**0.999**	0.950	0.950	0.993	0.999	0.887	
Breast cancer	**0.971**	0.962	**0.971**	0.946	0.967		0.958
Diabetic	**0.773**	0.757	0.762	0.757	0.759	0.631	0.757
Glass	**0.957**	0.872	0.847	0.737	0.942		0.710
Haberman	0.743	0.740	**0.766**	0.734	0.724		
Heart	0.841	0.804	**0.844**	0.805	0.837		
Hepatitis	0.852	**0.867**	0.839	0.830	0.836		0.810
Iris	0.951	0.967	0.965	0.964	**0.976**	0.896	0.933
Plan relax	**0.727**	0.709	0.645	0.718	0.705		
Sonar	0.950	0.916	0.860	0.939	0.944	0.923	**0.961**
Segment	0.764	0.749	0.768	0.752	**0.821**		0.759
Teaching	0.525	0.505	0.510	0.446	**0.551**		
Vehicle	**0.796**	0.645	0.489	0.482	0.763	0.618	0.724
Wine	0.975	0.958	0.932	0.839	**0.976**	0.937	
Zoo	**0.962**	0.827	0.520	0.814			0.912

It can be noted from Table 18 that CCBM outperformed three state-of-the-art ensemble classifiers in 11 of 16 datasets. Further comparisons with existing state-of-the-art clustering-based ensemble classifiers are given in Table 19. For fair comparisons, the classification accuracies over one run with 5-fold cross validation were averaged as the same were used in the respective paper.

Table 19: CBCLEL comparison with existing cluster-based ensemble classifiers

	CCBM				**Existing clustering-based state-of-the-art methods**				
Dataset	**SVM**	**Naïve Bayes**	**LDA**	**KNN**	**STLR [94]**	**CBCA [94]**	**MV [94]**	**STJ48 [94]**	**CL [94]**
Breast cancer	**0.971**	0.969	0.967	0.971	0.960	0.970	0.965	0.963	0.970
Glass	**0.967**	0.967	0.907	0.893	0.624	0.691	0.691	0.638	0.648
Haberman	0.752	0.765	0.752	**0.775**	0.738	0.765	0.725	0.728	0.734
Heart	**0.859**	0.696	0.815	**0.859**	0.842	0.845	0.766	0.842	0.823
Hepatitis	**0.900**	0.838	**0.900**	0.875	0.833	0.860	0.820	0.820	0.787
Iris	0.973	0.987	0.967	**0.987**	0.945	0.972	0.951	0.952	0.924
Segment	0.959	0.931	0.913	0.917	**0.965**	0.949	0.96	0.961	0.853
Sonar	0.822	0.831	0.841	0.798	**0.849**	0.790	0.781	0.756	0.756

It can be noted from Table 19 that CCBM out-performed existing clustering-based state-of-the-art ensemble classifiers in five out of eight datasets. SVM-based CCBM achieved the highest classification performance. To further validate the efficacy of results, non-parametric tests are adopted, and the p-values are listed in Table 20. It can be noted that that all p-values are statistically significant, further adding to the fact that the proposed ensemble classifier CCBM achieved significantly better performance than the other proposed methods, and the null hypothesis can be rejected which is that the performance improvements are a matter of chance.

Table 20: p-values of non-parametric signed rank tests

Methods	***p*-value**
MPRaF-T	**0.009**
PSEMISEL	**0.013**
STLR	**0.040**
CBCA	**0.013**
MV	**0.010**
STJ48	**0.006**
CL	**0.001**
WMV	**0.007**

The classification performance of the proposed approach has also been compared with some existing CNNs and the results are given in Tables 21 and 22, where can be noted that the proposed approach performed significantly better than the other CNNs.

Table 21: Comparative analysis of different CNNs on MNIST dataset

Approach	Methods	Accuracy
Niu et al. [100]	Hybrid CNN-SVM	99.81%
Ranzato et al. [101]	CNN	99.61%
Simard et al. [102]	CNN	99.60%
Ensemble of CNNs	**CCBM_CNN**	**100.00%**

Table 22: Comparative analysis of different CNNs on CIFAR-10 dataset

Approach	Methods	Accuracy
Sinha et al. [103]	Optimized CNN	78.60%
Krizhevsky et al. [104]	Alex Net	77.75%
Yamasaki et al. [104]	PSO CNN	80.15%
Ensemble of CNNs	**CCBM_CNN**	**87.11%**

6.3 Summary

In this chapter a novel clustering-based ensemble classifier method was proposed that was incorporated in the original ensemble framework. The intuition behind CCBM was to improve the input data space by generating balanced clusters and training multiple base classifiers on small balanced dataset. The proposed method not only generated a rich input space for training of base classifiers, but also solved the problem of class imbalances from the data to lurk into the data clusters. To achieve this, input data was clustered based on their classes to generate class-pure data clusters. Each class-pure data cluster contained data patterns (images) only from a certain class. All class-pure data clusters were then balanced by adding samples from other classes that were closest to the centroid of the data cluster. On each balanced data cluster, a base classifier was trained to generate a pool of base classifiers. All classifiers in the pool were utilised to generate the ensemble. The proposed approach was not only evaluated on classification benchmark datasets but was also tested with benchmark image datasets. The experimental results analysis showed that the proposed ensemble method can achieve higher accuracy than existing state-of-the-art methods on benchmark datasets in which samples of each class can be grouped into multiple groups. The statistical significance test was also conducted, and it showed that the results are statistically significant.

Chapter 7: **Optimal Class-Pure Cluster Generation Method**

This chapter further extends CCBM by proposing a novel ensemble method, namely Optimal Class-Pure Cluster Generation Method (OCGM), that computes the optimal number of data clusters k of each data class by conducting a cluster analysis. Later, only the optimal k is utilised to generate a pool of class-pure data clusters which are balanced, and an ensemble is generated by training a set of diverse base classifiers on balanced data clusters. The proposed ensemble method is evaluated on several benchmark datasets and results are compared with existing state-of-the-art ensemble methods.

7.1 Ensemble with optimal class-pure cluster generation method

7.1.1 The proposed method

A key hyper-parameter to clustering algorithms is the number of clusters the given input samples should be partitioned into. As stated earlier, this key hyper-parameter is commonly referred to as k, which is given as an argument to the clustering algorithm before the process starts. Another limitation of utilising clustering to generate a random subspace, is that we do not know a-priori the optimal value of k. Therefore, to tackle this limitation, an ensemble classifier method OCGM was proposed[5] that finds the optimal value of k that can satisfy a given criteria, and the optimal value is used to generate a random subspace of class-pure data clusters. As before, the dataset is first partitioned into its various constituent classes as $X^1, X^2, \dots, X^\kappa \subset X$ and the following holds true $X^1 \cup X^2, \dots \cup X^\kappa = X$ and $X^1 \cap X^\kappa = \emptyset$ meaning the partitions X^κ are disjoint sets. Then for each subset X^κ data clusters are generated, but first the optimal number of clusters k that should be generated is computed. A *Silhouette* analysis using equation (9) of each data subset X^n is conducted to identify the optimum number of data clusters that must be generated to partition the data subset efficiently. All generate class-pure data clusters are balanced as previously from Section 6.1, by adding samples from non-dominant classes that are closest to cluster centroid.

[5] The work presented in this chapter has been published in: Z. Jan and B. Verma, "Optimal Clusters Generation for Maximizing Ensemble Classifier Performance", International Joint Conference on Neural Networks, 2020, pp. 1-7, (Accepted on 20th March 2020).

7.1.2 Ensemble classifier generation using OCGM

The ensemble classifier framework from Section 3.1 is utilised with OCGM to generate an ensemble trained on the optimal number of class-balanced data clusters.

Algorithm 5: Ensemble classifier generation using OCGM

```
Input: Training dataset $X$, Validation dataset $V$, Testing data set $T$, Set of base
        classifiers $\beta$
Output: Optimal class-pure cluster ensemble classifier

Generate $X^{\kappa}$ subsets of the input data such that $X^1, X^2, \dots X^{\kappa} \subseteq X$ that are
  class pure, meaning data samples from only a single class in each subst. If
  there are $\kappa$ data classes in the dataset $\kappa$ subsets are generated
Initialise $i = 1$
foreach $X^{\kappa}$ do
   $k \leftarrow$ find optimal value of clusters by conducting Silhouette analysis of $X^{\kappa}$
   using eq (9)
   $sp^i \leftarrow$ partition $X^{\kappa}$ into $k$ data clusters by minimizing the squared
   Euclidean distance of each sample from the cluster centroid
endfor
foreach class-pure cluster $C$ in $sp$ do
    determine the dominant class $y_d$ in cluster
   $y_{nd} \leftarrow$ determine nondominant classes in the cluster
   foreach $y_{nd}$ do:
      add feature vectors $x$ from remaining clusters in the pool closest to
      cluster centroid
   endfor
endfor
Initialize $i = 1$
foreach data clusters $C$ in pool $sp$ do
      $bcp^i \leftarrow$ train a set of base classifiers $\beta = \{\zeta^1, \zeta^2, \dots, \zeta^{\varsigma}\}$ on the data
      cluster $C$ and add to the pool
endfor
Use the pool $bcp$ of classifiers to predict class labels of the unseen dataset,
  the test set and calculate ensemble classification accuracy.
```

All balanced data clusters are utilised to train a set of diverse base classifiers to generate the base classifier pool bcp. These are then combined to generate the ensemble as in the original ensemble framework. The algorithm for training an ensemble using OCGM is given in Algorithm 5.

7.2 Experiments, results and analysis

7.2.1 Datasets

The following benchmark classification datasets, namely *Breast Cancer, Diabetic, E.coli, Glass, Haberman, Heart-s, Hepatitis, Ionosphere, Iris, Liver, Segment,* and *Sonar,* from the UCI Machine Learning repository [84], were utilised to evaluate the performance of the proposed ensemble method. Details of these datasets are already mentioned in Table 2.

7.2.2 Experimental setup

To accommodate for randomness, a 10-fold cross validation is conducted and classification accuracy over 10 independent runs is calculated for analysis. The default implementation of base classifiers SVM, ANN, DISCR, DT, NB, and KNN is used without any parameter optimisation. For cluster validation, default implementation of "*evalclusters*" in MATLAB is used with "*Silhouette*" as a criterion parameter. The range of clusters for each dataset is measured from two to 20, and the optimal number of clusters identified is used as a parameter to K-means. For comparative analysis Adaboost, and Random Forest were also used in experiments. The default implementation of ensembles in MATLAB were used with the following parameters for Raf: *fitcensemble, Method = bag, Learners = tree;* and for Adaboost: fitcensemble, *Method = AdaboostM2 (for datasets with more than two classes), Method = AdaboostM1 (for datasets with two classes).*

7.2.3 Results

The average classification accuracy of the proposed ensemble classifier with OCGM over 10-independent runs is given in Table 23.

Table 23: Classification accuracies of the proposed ensemble framework using OCGM

Dataset	**OCGM**	**Std. Dev.**	**Avg. clusters per class**
Breast Cancer	0.9700	0.011	3
Diabetic	0.7722	0.035	2
E.coli	0.8513	0.034	6
Glass	0.9673	0.021	9
Haberman	0.7652	0.035	5
Heart-s	0.8374	0.008	2
Hepatitis	0.8625	0.028	6
Ionosphere	0.9262	0.009	5
Iris	0.9667	0.033	10
Liver	0.7246	0.058	2
Segment	0.9525	0.002	3
Sonar	0.7945	0.022	3
Spectfheart	0.8023	0.011	8
Thyroid	0.9113	0.029	13
Vehicle	0.8026	0.037	2
Wine	0.9887	0.015	4

7.2.4 Discussion

The number of data clusters generated for each dataset in Table 23 is computed by conducting a *Silhouette* analysis, and instead of trying for multiple combinations as in ensemble proposed in chapter 6, only the optimal number was utilised to generate the data clusters. It can be noted from Table 23 that for each dataset a different number of clusters is generated. This is predominantly due to equation (9), as only the optimal number of clusters is generated for each dataset. This adds to the fact that each dataset has different characteristics and using the same upper bounds for different datasets is not an ideal strategy.

7.2.5 Comparative analysis

The classification performance of the proposed ensemble classifier with OCGM is compared with existing state-of-the-art ensemble classifiers, namely Random Forest and Adaboost. The results are summarised in Figure 16, and it can be noted that the proposed ensemble outperformed Adaboost, and RaF in 11 out of 16 datasets and achieved an average of 1.52% performance gains over Adaboost, and 2.71% over RaF.

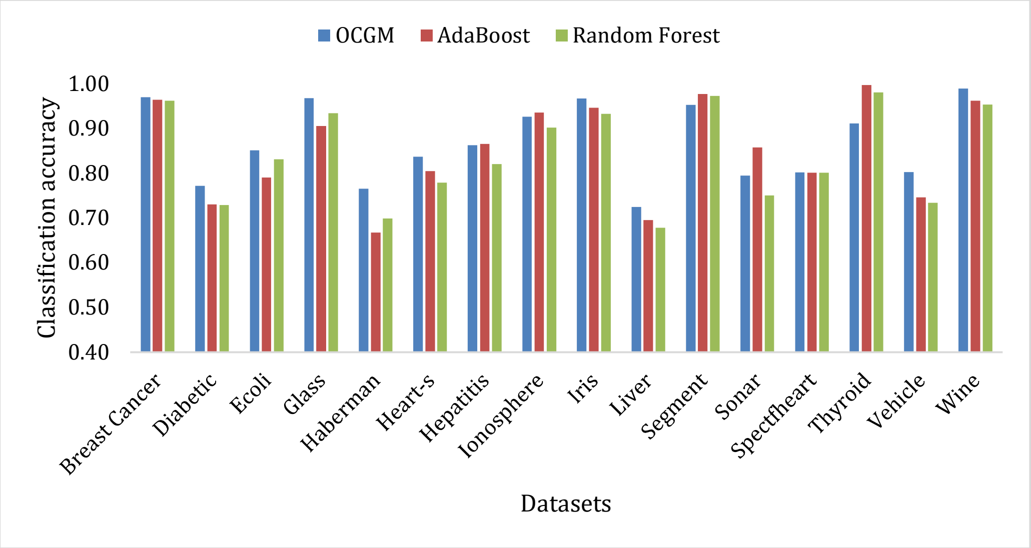


Figure 16: Performance of the proposed ensemble method OCGM in comparison with state-of-the-art ensemble classifiers

The proposed ensemble is also compared with various clustering-based ensemble classifiers [94], such as STLR, CBCA, CL, and STJ48. The classification accuracies are taken directly from the respective papers and the results are summarised in Figure 17. It can be noted that, on average, the proposed ensemble approach achieved 7.98% performance gains over STJ48, 5.36% over STLR, 7.64% over CL, and 3.68% over CBCA.

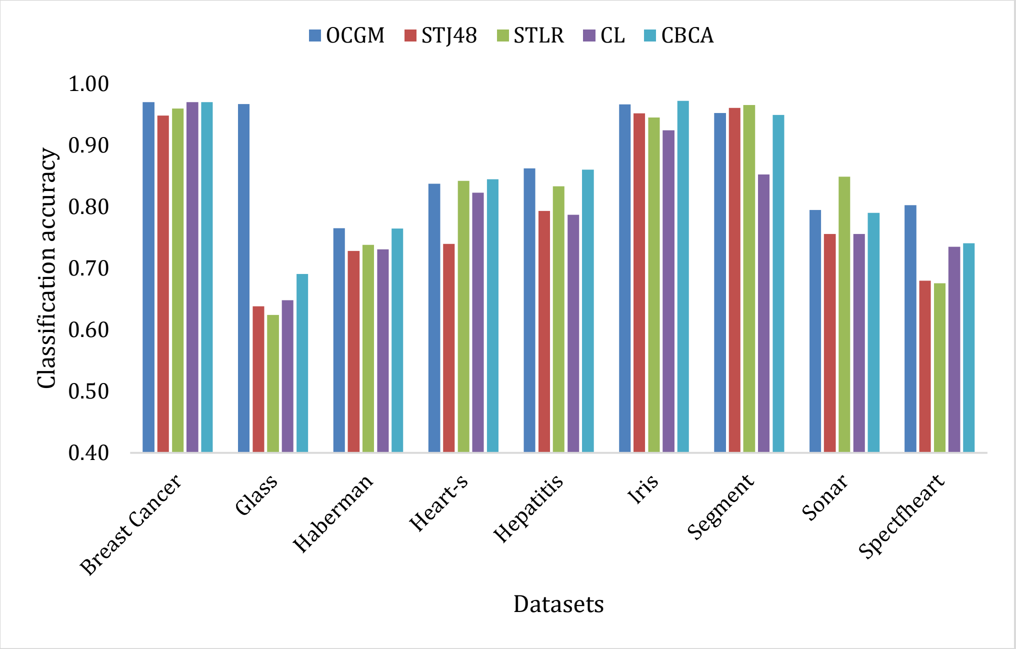


Figure 17: Performance of the proposed ensemble method OCGM in comparison with state-of-the-art clustering-based ensemble classifiers

The proposed method performed significantly better in comparison to Bagging, Random Forest, STJ48, and CL rejecting the null hypothesis, and the results are inconclusive in comparison with AdaBoost, STLR and CBCA. The p-values are given in Table 24.

Table 24: p-values of Wilcoxon signed rank tests of OCGM

Methods	*p*-value
AdaBoost	0.080
Random Forest	**0.005**
STLR	0.086
CBCA	0.163
STJ48	**0.005**
CL	**0.005**

7.3 Summary

In this chapter a novel ensemble classifier method was proposed that utilizes a cluster validation strategy to find the optimal value of k that each data class must be partitioned into. The optimal number of data clusters is determined by conducting a *Silhouette* analysis. All generated clusters are then balanced by adding samples from other classes closest to the cluster centroids. From experiments, it was evident that each dataset has different characteristics, which in turn required a different number of optimal data clusters to be generated to achieve the highest classification accuracy.

Chapter 8: **Classifier Selection by Multiple Elimination Method**

In this chapter a classifier selection methodology is proposed namely, Classifier Selection by Multiple Elimination Method (CSMEM), that generates an ensemble by selecting base classifiers from the pool based on accuracy and diversity comparisons. A classifier is given multiple opportunities to participate in a round of selection, and it will become a part of the ensemble only if it can either contribute to overall accuracy of the ensemble or maintain accuracy and increase diversity. The proposed ensemble method is evaluated on several benchmark datasets and results are compared with existing state-of-the-art ensemble methods.

8.1 Ensemble with classifier selection by multiple elimination method

8.1.1 The proposed method

When generating an ensemble classifier, selecting the best set of classifiers from the base classifier pool is categorized as a combinatorial problem, and an efficient classifier selection methodology must be utilised to select the best subset of classifiers from the pool. For a large problem space, a combinatorial problem becomes an NP-hard problem. The computational complexity of combinatorial problems is largely dependent on the input size and therefore, it is not completely known if an algorithm exists that can solve all instances in polynomial time. Therefore, optimisation algorithms are employed in the hope of finding the best "combinations" in polynomial time. Although many optimisation algorithms have been utilised to select the best set of classifiers, optimising classifiers is limited by the performance of the optimiser as well. Also, optimizers are constrained by the upper bounds of their iteration. Therefore, in this chapter an ensemble method is proposed CSMEM[6], where the focus is to select the best set of classifiers to generate an ensemble classifier that can not only achieve higher classification accuracy, but diversity as well. The proposed method also reduces the complexity of the base

[6] The work presented in this chapter is currently under 3rd round of review and given consideration for full publication in ACM Transactions on Intelligent Systems and Technology: Z. Jan and B. Verma, "Multiple Elimination of base Classifiers in Ensemble Learning using Accuracy and Diversity Comparisons'", ACM Transactions on Intelligent Systems and Technology, 2020.

classifier selection process to $O(3 \times b)$ where $b = n - m$ and m is the number of classifiers that achieved maximum performance in first round and n is the total number of classifiers in the pool.

CSMEM allows each classifier in the pool to have multiple chances to participate in a round of selection. In each round a classifier is given a chance to be taken from the base classifier pool and become a part of the ensemble. A classifier is selected if it can contribute to ensemble accuracy or maintain accuracy whilst increasing ensemble diversity, otherwise it is discarded. If a classifier fails in the first attempt, it still has chances to participate in a later round of selection. A classifier is discarded if it is not selected and has no remaining chances of selection. Firstly, a base ensemble solution is generated consisting of a subset of base classifiers from the pool bcp that achieved the highest classification accuracy on validation dataset given as:

$$\xi^j = \{\zeta^1, \zeta^2, \ldots, \zeta^m\}, \tag{24}$$

where ζ is a trained base classifier from the pool bcp consisting of m base classifiers and $m < n$, and n is the total number of base classifiers in the pool. The pool now has a total of has $z = n - m$ remaining base classifiers, which will go through the process of multiple eliminations. A row vector is generated which denotes the new base classifier pool and is represented as:

$$b = [R_1, R_2, \ldots, R_z], \tag{25}$$

where R_z is the total number of chances each classifier has in the pool. A classifier is randomly selected from the pool, let's say ζ^z, and is added to the ensemble ξ^j; let the new ensemble solution be denoted as $\xi^k = \{\zeta^1, \zeta^2, \ldots, \zeta^m, \zeta^z\}$. Ensemble accuracy is calculated using equation (5). If ensemble solution ξ^k has higher classification accuracy than ensemble ξ^j , then ensemble ξ^k is the new base ensemble solution, and the classifier's respective index in b is changed to 0, however, if ξ^k has the same accuracy as ξ^j, then ensemble diversity is calculated. DF diversity measure from equation (8) is computed for both the ensembles ξ^j and ξ^k. If ξ^k has higher diversity than ξ^j , then again ensemble ξ^k is the new base solution and the classifier's respective index in b is changed to 0. If however, ξ^k neither has higher classification accuracy nor diversity than ξ^j, then classifier ζ^z's respective index (chance) is decreased by 1, and ensemble ξ^j remains as the base ensemble solution. Another random classifier is chosen from the pool b and the process is repeated until all classifiers in b have 0 chances remaining in b. At the end each classifier has had multiple opportunities to be a part of the ensemble and they are discarded only if they fail to contribute to the ensemble in multiple attempts.

8.1.2 Ensemble classifier generation using CSMEM

The ensemble framework from Section 3.1 is utilised with CSMEM to generate an ensemble that selects base classifiers from the pool based on accuracy and diversity comparisons. The base classifier pool is generated by incrementally clustering input data up to K with k data clusters generated in each iteration. A set of diverse base classifiers is trained on all generated data clusters and a base classifier pool is generated. Classifiers from the pool are selected using CSMEM and an ensemble is formed. The algorithm of the proposed CSMEM is given in Algorithm 6.

Algorithm 6: Ensemble classifier generation using CSMEM

Input: Training dataset X, Validation dataset V, Testing data set T, Set of base classifiers ζ, Number of chances R, upper bounds K

Output: Classifier selection by multiple elimination ensemble classifier

```
Initialise $i$
for $k = 1$ to $K$ do
    $sp^i$ ← partition training dataset into $k$ clusters by minimising the squared
    Euclidean distance of each sample from the cluster centroid
endfor
initialize $i$
foreach data clusters $C$ in pool $sp$ do
    $bcp^i$ ← train a set of base classifiers $\beta = \{\zeta^1, \zeta^2, \ldots, \zeta^{\varsigma}\}$ on the data cluster
    $C$ and add to the pool
increment $i$
endfor
Generate a base ensemble solution $\xi$ consisting of the top performance base
classifiers from the pool $bcp$ using the validation dataset $V$
Compute ensemble $\xi^j$ accuracy using equation (4)
Generate row vector $b$ with $R$ chances representing the remaining base
classifiers in the pool
```

Algorithm 6: Ensemble classifier generation using CSMEM

```
while $b$ has any $R$ that is non-zero
    Select a base classifier $\zeta^i$ randomly from $bcp$
    Generate new ensemble solution $\xi^l$ by adding classifier $\zeta^i$ to $\xi^l$
    Compute ensemble $\xi^l$ and $\xi^j$ accuracy using equation (4)
    Compute ensemble $\xi^l$ and $\xi^j$ diversity using equation (7)
    if $\xi^l$'s accuracy > $\xi^j$'s accuracy
        update new base ensemble solution to $\xi^l$
        set $R$ of selected classifier's in $b$ to 0
    elseif $\xi^l$'s accuracy = $\xi^j$'s accuracy and $\xi^l$'s diversity > $\xi^j$'s diversity
        update new base ensemble solution to $\xi^l$
        set $R$ of selected classifier's in $b$ to 0
    else
        decrease selected classifier's (chance) $R$ by 1 in b
    endif
endwhile
Use the updated base ensemble solution predict class labels of the unseen
  dataset, the test set and calculate ensemble classification accuracy
```

8.2 Experiments, results and analysis

8.2.1 Datasets

The following benchmark classification datasets, namely *Breast cancer, Diabetic, E. coli, Glass, Haberman, Ionosphere, Iris, Liver, Segment, Sonar, Thyroid, Vehicle,* and *Wine,* from the UCI Machine Learning repository [84], were utilised to evaluate the performance of the proposed ensemble method. Details of these datasets are already mentioned in Table 2, in Section 3.2.

8.2.2 Experimental setup

The proposed ensemble framework with CSMEM is implemented in MATLAB, a 10-fold cross validation is adopted to incorporate for randomness as in other similar works. The upper bounds of clusters K was set to $\sqrt[3]{n}$ where n is the number of samples in a dataset as in other similar works [29]. The value of R

(chances for each classifier) was set to three. This was calculated based on trial and error and three achieved the highest classification accuracy.

8.2.3 Results

Average classification accuracies over 10-folds of CSMEM over the 13 benchmark datasets are given in Table 25. Also, given in Table 25 are the number of base classifiers generated that become part of the base classifier pool, and the number of classifiers that were selected after CSMEM.

Table 25: Classification performance of the proposed ensemble framework using CSMEM

Dataset	Proposed approach	Standard deviation	Classifiers in pool	Classifiers selected
Breast cancer	0.9685	0.023	990	40
Diabetic	0.7549	0.035	1908	15
E. coli	0.8777	0.033	697	11
Glass	0.9999	0.080	500	20
Haberman	0.7551	0.060	740	8
Ionosphere	0.9178	0.030	572	5
Iris	0.9800	0.160	303	15
Liver	0.7242	0.023	4262	13
Segment	0.9683	0.050	1920	7
Sonar	0.9102	0.090	170	9
Thyroid	0.9959	0.0004	5510	20
Vehicle	0.8226	0.050	1360	15
Wine	0.9830	0.026	435	50

8.2.4 Discussion

It can be noted from Table 25, that from a pool of 1908 classifiers for *diabetic* dataset, only 15 classifiers were enough to achieve the highest classification accuracy. Similarly, for *thyroid* dataset out of 5510 trained base classifiers only 20 classifiers were selected that achieved the highest classification accuracy. A higher number of data samples will typically have more classifiers in the pool because more data clusters are generated, and on each cluster a set of base classifiers is trained. Results from Table 25 certainly add to the fact that more does not necessarily mean better and adding classifiers in an ensemble after an optimum value does not add any value to the ensemble besides increasing computational complexity. Therefore, an appropriate classifier selection methodology should always be employed when selecting trained base classifiers from the pool.

8.2.5 Comparative analysis

The proposed ensemble method CSMEM is compared with legacy ensemble classifiers methods such as Bagging, and Boosting, and an existing state-of-the-art ensemble classifier method namely OEC-ILC that generates ensemble classifiers through the incorporation of a rule-based accuracy and diversity comparison. The results are summarised in Figure 18 and it can be noted that CSMEM achieved 1.16% performance improvement over Bagging, 2.83% over Boosting, and 0.34% performance improvement over OEC-ILC.

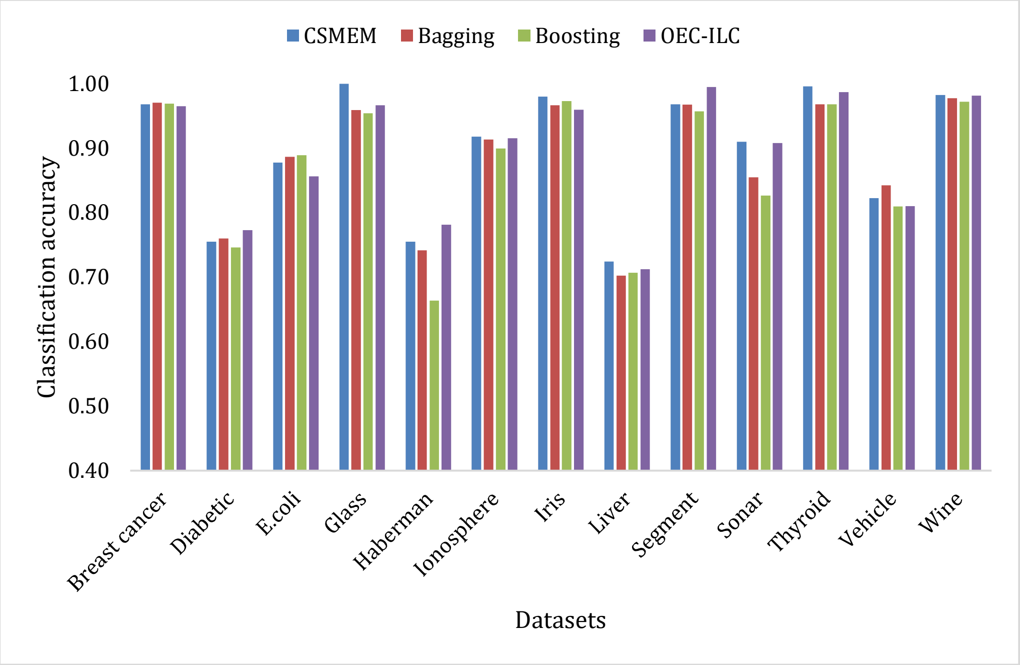


Figure 18: Performance of the proposed ensemble method CSMEM in comparison with state-of-the-art ensemble classifiers

In comparison with existing clustering-based algorithms, CSMEM achieved on average 12.01% performance improvement over STJ48, 9.8% over STLR, 14.34% over CL, and 8.66% over CBCA. The results are summarised in Figure 18. The proposed method performed significantly better in comparison to Bagging, Boosting, STJ48, STLR, and CL rejecting the null hypothesis, and the results are inconclusive in comparison with OEC-ILC, and CBCA. The p-values are given in Table 26.

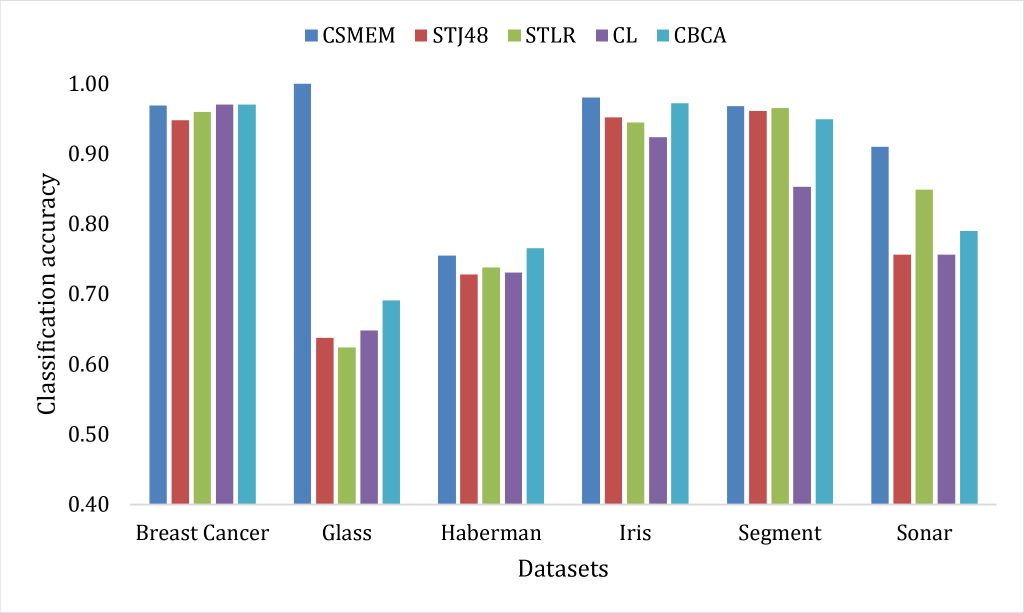


Figure 19: Performance of the proposed ensemble method CSMEM in comparison with state-of-the-art clustering-based ensemble classifiers

Table 26: p-values of Wilcoxon signed rank tests of CSMEM

Methods	p-value
OEC-ILC	0.155
Bagging	0.050
Boosting	**0.004**
STJ48	**0.010**
STLR	**0.010**
CL	**0.020**
CBCA	0.080

8.3 Summary

In this chapter a novel classifier selection method is proposed. The proposed method gives each classifier in the pool multiple opportunities to be taken from the pool and become a part of the ensemble. The proposed ensemble classifier first generates a diverse input space by clustering input data incrementally to generate sparse data clusters. A set of diverse base classifiers is trained on all generated data clusters to generate a base classifier pool. Classifiers from the pool are selected using the proposed classifier selection methodology to generate an ensemble classifier.

Decisions from all selected classifiers are combined through majority voting to compute final ensemble decision. The proposed classifier selection methodology selects a classifier from the pool if it can contribute to the overall ensemble classifier accuracy, or at the least maintain the classification accuracy whilst increasing ensemble classifier diversity. Every classifier is given multiple chances to participate in a round of selection, and it is discarded only if that classifier has no chances left. From a thorough analysis and experimentation, it was evident that from a relatively large pool only a few classifiers were chosen which can achieve high classification accuracy. This adds to the idea that a large component size ensemble is not necessarily a better ensemble, and a relatively few classifiers with diverse learning capabilities is enough to get optimum classification accuracy.

Chapter 9: **Misclassification Diversity Method**

In this chapter a further classifier selection methodology is proposed that utilises the proposed diversity measure, namely Misclassification Diversity (MD). The proposed method's performance is compared with different pairwise diversity measures to test its efficacy and other ensemble classifier approaches.

9.1 Ensemble with misclassification diversity method

9.1.1 The proposed method

Diversity is considered an important factor when it comes to generating an ensemble classifier, as diverse classifiers, when combined suitably, generate accurate ensembles. If we combine classifiers that have correlated errors, then the ensemble classifier generated with n classifiers, and an ensemble classifier generated with only two classifiers, will have no difference. In order to achieve the benefits of having more than one classifier in an ensemble, we must have diverse classifiers. Therefore, in this chapter a pairwise diversity measure is proposed, and using the proposed diversity measure a classifier selection methodology is implemented, namely MDM [105][7], and an ensemble is generated.

In order to understand the proposed diversity measure, let's assume two ensemble solutions: i^{th} ensemble solution denoted as $\xi^i = \{\zeta^1, \zeta^2, \zeta^3\}$ and j^{th} ensemble solution denoted as $\xi^j = \{\zeta^1, \zeta^2, \zeta^9, \zeta^{12}\}$ where ζ^n is a classifier from the pool. The purpose of introducing a diversity measure here is to find how diverse two classifiers are, and whether adding a new classifier in the ensemble causes any differences in the prediction capabilities of the ensemble; if it does then that classifier can be added to the ensemble, otherwise adding it to the ensemble is not beneficial. To compute the diversity all classifiers from both ensembles are utilised to classify the input feature vector of the validation dataset. Results are stored in two-dimensional data matrices as:

[7] The work presented in this chapter has been published in the following paper: Z. Jan and B. Verma, "A Novel Diversity Measure and Classifier Selection Approach for Generating Ensemble Classifiers," IEEE Access, vol. 7, pp. 156360-156373, 2019.

$$\xi^i = \begin{bmatrix} {y'}^{c1}_{1} & {y'}^{c2}_{1} & {y'}^{c3}_{1} \\ {y'}^{c1}_{2} & {y'}^{c2}_{2} & {y'}^{c3}_{2} \\ \vdots & \vdots & \vdots \\ {y'}^{c1}_{n} & {y'}^{c2}_{n} & {y'}^{c3}_{n} \end{bmatrix}, \tag{26}$$

$$\xi^j = \begin{bmatrix} y'^{c1}_1 & y'^{c2}_1 & y'^{c9}_1 & y'^{c12}_1 \\ y'^{c1}_2 & y'^{c2}_2 & y'^{c9}_2 & y'^{c12}_2 \\ \vdots & \vdots & \vdots & \vdots \\ y'^{c1}_n & y'^{c2}_n & y'^{c9}_n & y'^{c12}_n \end{bmatrix}, \tag{27}$$

where y'^{c1}_1 is the predicted class labels of classifier 1 on the 1st sample. A column wise mathematical mode of the data matrices is taken to get the predictions of each ensemble (majority voting) as given below:

$$\xi^i = \begin{bmatrix} {y'}_1 \\ {y'}_2 \\ \vdots \\ {y'}_n \end{bmatrix}, \qquad \xi^j = \begin{bmatrix} {y'}_1 \\ {y'}_2 \\ \vdots \\ {y'}_n \end{bmatrix}, \tag{28}$$

After obtaining the final predictions of each ensemble, a column wise matrix of misclassified samples is generated. This matrix contains 1 for any misclassified labels and 0 otherwise. For example, if dataset *X* has only six samples (for the sake of simplicity), and ensemble i and ensemble j misclassified the following labels $y^o_i =< y^o_2, y^o_3, y^o_6 >$ and $y^o_j =< y^o_1, y^o_3, y^o_6 >$ then their misclassification matrices is computed as follows:

$$y^o_i = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \qquad y^o_j = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \tag{29}$$

Then the Misclassification Diversity (MD) of the two ensemble solutions is calculated using the following equation:

$$MD^n = \frac{\sum_{\forall i,j \mid i \neq j} I'\{y^o_i \neq \ y^o_j\}}{\left|y^o_i \ \cup \ y^o_j\right|}, \tag{30}$$

where y^o_i is the misclassified label of ith ensemble ξ^i on a validation dataset V, the denominator $\left|y^o_i \ \cup \ y^o_j\right|$ gives the count of total number of errors caused by both the ensembles, and I' is an indicator function of misclassified labels between two ensembles given as:

$$I'(y_i^o, y_j^o) = \begin{cases} 0, & y_i^o = y_j^o \\ 1, & y_i^o \neq y_j^o \end{cases}, \qquad (31)$$

$$\forall i, j \mid i \neq j$$

The output of the indicator function I' for ensemble's misclassification matrices in (27) is computed as follows:

$$I'(y_i^o, y_j^o) = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \qquad (32)$$

The diversity MD using equation (28) is computed as follows: The numerator $\sum_{\forall i,j \in Z \mid i \neq j} I'\{y_i^o \neq y_j^o\}$ becomes 2 and denominator $|y_i^o \cup y_j^o|$ becomes 4. Therefore, the diversity MD is $2/4$ or 0.5, since there are only 2 different misclassified labels out of a total of 4. The diversity is 1 if the two ensembles in comparison made totally different errors and 0 otherwise. The proposed diversity helps in identifying those classifiers which bring new learning capabilities to the ensemble and helps in removing the redundant classifiers.

9.1.2 Ensemble classifier generation using MDM

The ensemble framework from Section 3.1 is utilised with MDM to generate an ensemble that selects base classifiers from the pool based on accuracy and diversity comparisons. The base classifier pool is generated by incrementally clustering input data up to K with k data clusters generated in each iteration. A set of diverse base classifiers is trained on all generated data clusters and a base classifier pool is generated. Classifiers from the pool are selected using MDM and an ensemble is formed. The algorithm of using the proposed diversity measure to generate an ensemble classifier is given in Algorithm 7.

Algorithm 7: Ensemble classifier generation using MDM

Input: Training dataset X, Validation dataset V, Testing data set T, Set of base classifiers ζ, upper bounds K

Output: Misclassification diversity-based ensemble classifier

```
Initialise $i = 1$
for $k = 1$ to $K$ do
    $sp^i$ ← partition training dataset into $k$ clusters by minimising the squared Euclidean distance of each sample from the cluster centroid
endfor
initialise $i = 1$
foreach data clusters $C$ in pool $sp$ do
    $bcp^i$ ← train a set of base classifiers $\beta = \{\zeta^1, \zeta^2, \dots, \zeta^\varsigma\}$ on the data cluster $C$ and add to the pool
endfor
Generate a base ensemble solution $\xi^j$ consisting of two random base classifiers from the pool $bcp$
Compute ensemble $\xi^j$ accuracy using equation (4)
foreach base classifier $\zeta$ in pool $bcp$
    add classifier $\zeta$ to ensemble $\xi^j$ and generate new ensemble solution $\xi^l$
    Compute ensemble $\xi^l$ and $\xi^j$ accuracy using equation (4)
    Compute ensemble $\xi^l$ and $\xi^j$ diversity using equation (28)
    if $\xi^l$'s accuracy > $\xi^j$'s accuracy and $\xi^l$'s diversity > $\xi^j$'s diversity
        update new base ensemble solution to $\xi^l$
    else
        discard base classifier $\zeta$
    endif
endfor
Use the updated base ensemble solution to predict class labels of the unseen dataset, the test set and calculate ensemble classification accuracy
```

9.2 Experiments, results and analysis

9.2.1 Datasets

The following benchmark classification datasets namely *Breast cancer, Diabetic, E. coli, Glass, Haberman, Ionosphere, Iris, Liver, Segment, Sonar, Thyroid, Vehicle,* and *Wine,* from the UCI Machine Learning repository [84], were utilised to evaluate the performance of the proposed ensemble method. Details of these datasets are already mentioned in Table 2, in Section 3.2.

9.2.2 Experimental setup

The proposed ensemble framework with MDM is implemented in MATLAB, a 10-fold cross validation is adopted to incorporate for randomness, as in other similar works. The upper bounds of clusters K was set to $\sqrt[3]{n}$ where n is the number of samples in a dataset as in other similar works [29]. To calculate diversity of two ensemble solutions the dissimilarity matrix from Section 3.3.3 is computed after classifying the feature vector of the training set using the predicted labels and the ground truth.

9.2.3 Results

In this Section the efficacy of the proposed ensemble framework with MDM is analysed. The classifier selection methodology is named as Incremental Layer Classifier Selection *with* Misclassification Diversity (ILCSMD). ILCSMD is not only tested in comparison with existing state-of-the-art ensemble classifier approaches, but also compared with existing pairwise diversity measures given in [24], namely Disagreement Measure (DM), Q test (QT), Double Fault (DF), Inverse Correlation Coefficient (IC), and Interrater-K (IK).

Average classification accuracies over 27 datasets achieved by ILCSMD, ILCSIC, ILCSIK, ILCSQT, ILCSDF, and ILCSDM are given in Table 27. It can be noted that ILCSMD performed 21.12%, 21.27%, 21.30%, 3.17%, and 1.39% better than ILCSIC, ILCSIK, ILCSQT, ILCSDF, and ILCSDM respectively. DF and DM performed relatively better than other diversity measures with DM achieving the second highest average classification accuracy and DF the third, however, a significant performance boost was achieved in comparison to IC, IK and QT.

Table 27: Comparison of ILCS with different diversity measures

ILCS with different diversity measures	Average accuracy
ILCSMD	**0.893**
ILCSIC	0.737
ILCSIK	0.736
ILCSQT	0.736
ILCSDF	0.865
ILCSDM	0.881

9.2.4 Discussion

Figure 20 shows the effect of incrementally selecting classifiers in layers in ILCSMD. For the sake of simplicity eight datasets were chosen based on number of records, number of features, and number of classes. We can see from the Figure that in ILCSMD there is a positive linear relation between accuracy and proposed diversity. Also, the number of layers increment only if both accuracy and diversity increase.

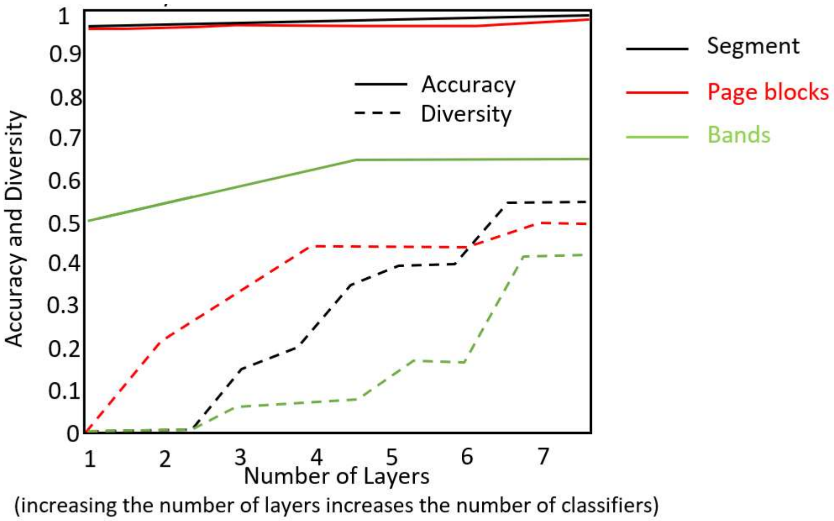


Figure 20: Accuracy and diversity comparison over different layers with ILCSMD

9.2.5 Comparative analysis

The classification performance of ILCSMD is compared with two pruning based ensemble classifiers, namely PSEMISEL [62], EBAGTS [106], and IDAFSEN [107]. The classification accuracies are taken directly from their respective papers, and the results are summarised in Figures 20, 21, and 22 below. On average the proposed method achieved performance gains of 2.3% over IDAFSEN, and 6.84% over EBAGTS, and 15.75% over PSEMISEL.

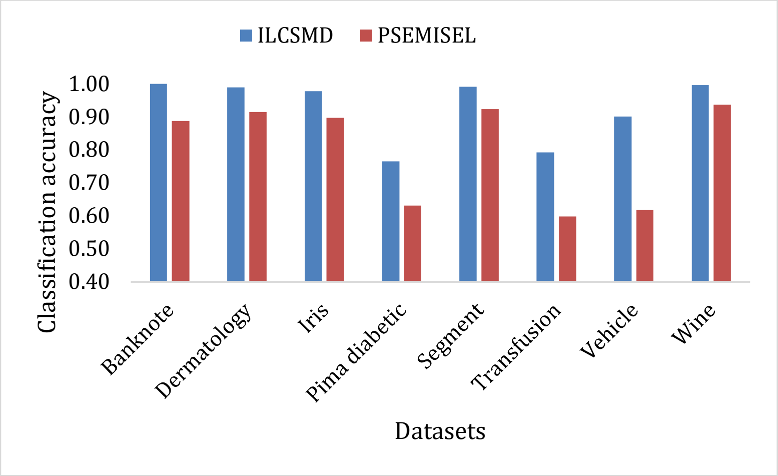


Figure 21: Comparison of ILCSMD with PSEMISEL

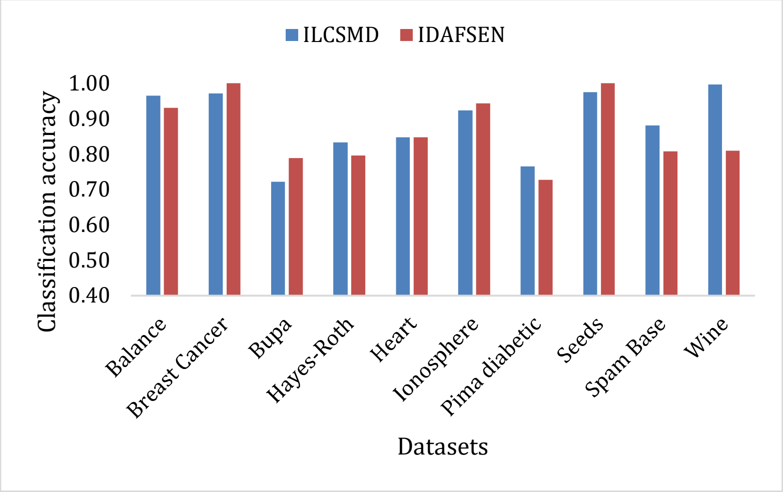


Figure 22: Comparison of ILCSMD with IDAFSEN

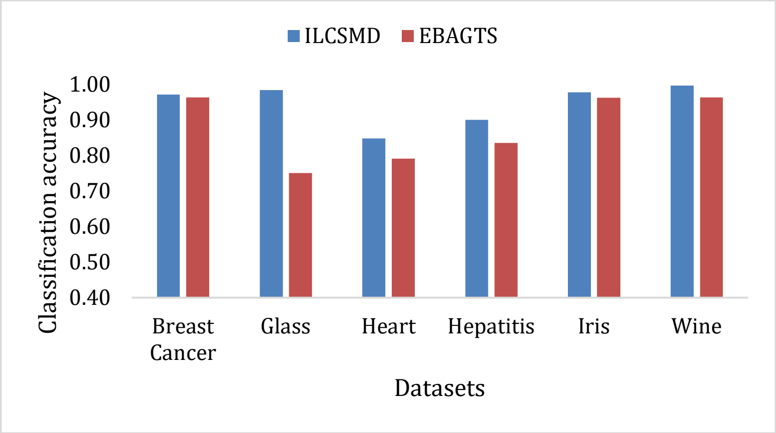


Figure 23: Comparison of ILCSMD with EBAGTS

To further validate the efficacy of results, the p-values are given in Table 28 below. ILCSMD performed significantly better than PSEMISEL, and EBAGTS rejecting the null hypothesis, and the results are inconclusive in comparison with PSEMISEL.

Table 28: p-values of Wilcoxon signed rank tests of ILCSMD

Methods	**p-value**
PSEMISEL	**0.011**
EBAGTS	**0.013**
IDAFSEN	0.130

9.3 Summary

In this chapter, a pairwise diversity measure and a classifier selection method that generates an ensemble using the proposed diversity measure, is introduced. The results and analysis presented in this chapter have shown that: selecting classifiers from the base classifier pool on the basis of diversity and accuracy have a positive effect on the overall ensemble classification accuracy; adding more classifiers to the ensemble classifier does not necessarily increase the performance of the ensemble, and a suiTable classifier selection process must be adopted; diversity measure on the basis of misclassified labels works better than other pairwise diversity measures, and; a robust classifier selection process has benefits in two folds; firstly, only those classifiers which can positively effect the ensemble classifier are selected and secondly, instead of having a very large ensemble component size, a small number of classifiers can achieve the same if not higher classification accuracy.

Chapter 10: **Conclusions**

This thesis proposes a novel ensemble classifier framework that utilises clustering to generate a random subspace for the training of base classifiers, to generate an ensemble classifier. Several novel ensemble classifier methods were proposed which used the proposed ensemble classifier framework, and incorporated optimisation algorithms, clustering balancing mechanisms, cluster optimisation mechanisms, various diversity measures, and classifier optimisation mechanisms. This chapter summarises the findings and provides future directions.

10.1 Contributions of this research

In this thesis several ensemble classifier methods were proposed. Initially a general ensemble learning framework is proposed in Chapter 3, that was used as a base line for developing further ensemble methods. The main building block of the proposed ensemble learning framework was the utilisation of clustering to generate a diverse input space for the training of base classifiers, also known as a random subspace.

Most of the existing ensemble classifier approaches utilise bagging to generate a random subspace, however, in this thesis, an argument was put in favour of clustering. It argues that clustering is better than bagging as it allows for the control of various hyper-parameters that can generate a diverse input space. This is done to control the bias and variance of base classifiers by training base classifiers on various data clusters. Additionally, since clustering is done in a two-dimensional cartesian coordinate system, it also incorporates and exploits any spatial characterise that may exist in a dataset. This enables the base classifiers that are trained on such data clusters to have local expertise rather than global, and this way a complex decision boundary can be broken down into its smaller less-complex constituents which are represented by simple local base classifiers. However, as effective as it may be, clustering still has some limitations and one key limitation is the number of data clusters a given input space should be separated into. Most of the existing approaches either determine the value of K through trial and error or use some derived formulae-based approach. The problem, firstly, is that using a static value of K for different datasets is not ideal and although a certain value may work well for one dataset, it may not work well for others. Secondly, calculating the value of K using a formulae-based approach using the raw data is not effective as an unbalanced data can have a negative effect on the derived value. Thirdly, not all trained base classifiers on generated data clusters should be included in the final ensemble, as class imbalances in the data lurk into generated data clusters and any classifier trained on such data clusters will negatively affect the performance of the ensemble. Therefore, several ensemble classifier methods were proposed in this thesis that can mitigate these limitations.

In Chapter 3, a general ensemble classifier learning framework is proposed that utilises clustering to generate a pool of data clusters which is used as an input space. Clustering is utilised to generate a random subspace, that in turn not only allows for the training of multiple base classifiers, but also incorporates diversity in two folds. Firstly, diversity is incorporated by using a set of diverse base classifiers that are structurally different from each other; secondly, since each set of classifiers is trained on a different data cluster, they bring with them different learning capabilities. The results are evident of the fact that the diversity incorporation through clustering is effective, and the ensemble generated through clustering achieved an average diversity increase of 18%.

In Chapter 4, a methodology of optimizing the upper bound K of clustering is introduced. This is done by training a single base classifier on generated data clusters and computing the accuracy over validation dataset. The optimisation process dynamically searches for a value of $K,$ until no further error minimization RMSE (root mean square error), or no relative change can occur. It was evident through experiments that on average four data clusters were generated per dataset. Although K appears to be directly proportional to the number of samples in the dataset, and large datasets did end up having a higher value of K, an average K of four resulting in 20 data clusters was enough to generate an ensemble that can achieve good classification performance. Furthermore, it was also proved that K has a logarithmic relation with the number of samples in a dataset.

Chapter 5 discussed a proposed methodology of incorporating an optimisation algorithm that optimises the pool of generated clusters and the pool of trained base classifiers to generate an optimised ensemble classifier. The main novelty here is that no matter what value of K is chosen, the proposed ensemble will only select a subset of the generated data clusters from the pool that can maximise the generalisation performance of the ensemble, discarding any redundant or noisy data clusters from the pool. Moreover, all base classifiers that are trained on the optimised pool of data clusters were also optimised further to select a subset of base classifiers that can not only achieve better, if not the same classification performance, than using the entire pool of base classifiers. Through experiments it was observed that the optimised ensemble achieved on average a component size reduction of 50%. This further adds to the fact that in ensemble classifiers, more does not necessarily mean better and adding more classifiers after an optimal number only adds to computational complexity rather than increasing the generalisation performance. It was also evaluated whether the optimisation process has any preference of base classifier or not. The results are given in the Table below:

Table 29: Effect of optimization on the choice of base classifiers and the size of ensemble classifier

			Number of classifiers selected after optimization					
Datasets	**Original ensemble size**	**Optimized ensemble size**	**ANN**	**SVM**	**KNN**	**DT**	**LDA**	**NB**
Adult	18	6	2	1	1	1	1	0
Australian	6	3	1	1	1	0	0	0
Balance	6	3	1	1	1	0	0	0
Bank Note	186	93	16	16	16	15	15	15
Breast Cancer	17	8	2	2	1	1	1	1
E.coli	39	16	4	4	4	4	0	0
Haberman	205	101	18	17	17	17	16	16
Ionosphere	84	40	7	7	7	7	6	6
Iris	14	7	2	1	1	1	1	1
Liver	208	99	17	17	17	16	16	16
Page Blocks	21	9	2	2	2	1	1	1
Diabetic	178	83	14	14	14	14	14	13
Segment	6	3	1	1	1	0	0	0
Sonar	59	28	5	5	5	5	4	4
Stat Image	266	127	24	24	24	23	16	16
Teaching	10	4	1	1	1	1	0	0
Thyroid	13	4	1	1	1	1	0	0
Vehicle	490	238	42	42	42	42	35	35
Vowel	95	46	8	8	8	8	7	7
WDBC	74	37	7	7	7	6	5	5
Wine	14	8	2	2	1	1	1	1

The Table lists the original ensemble size, ensemble size after optimisation, and the number and type of base classifiers in the final optimised ensemble. It can be noted that the most preferred classifier is ANN, after that in priority second best is SVM, third is KNN, fourth is DT and the least preferred are LDA and NB classifiers. These results are summarised in Figure 25 on the next page. It can also be noted from the Table that almost half of ensemble components (classifiers) are selected after optimisation, resulting in either the same accuracy or increased accuracy. The classifier preference is illustrated in Figure 24 below.

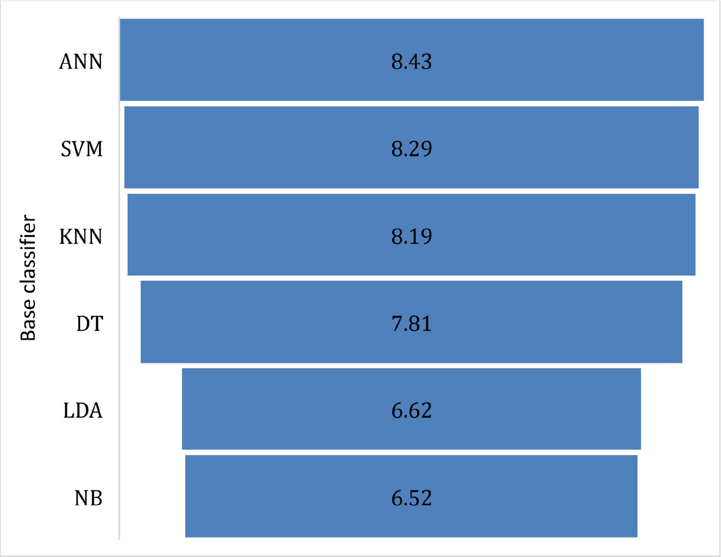


Figure 24: Number and type of base classifiers used in ensemble generation

The proposed optimised ensemble method was further extended by proposing a hybrid optimised ensemble learning framework. The results and analysis show that:

(i) the reduction of input feature space in ensemble has a positive impact and improves classification accuracy;

(ii) the reduction of base classifiers through optimisation has benefits in two folds, firstly, optimising the base classifier pool leads to the generation of an ensemble classifier that has lower number of ensemble components (base classifiers) and, secondly, the generated optimised ensemble has same if not higher accuracy than the non-optimised ensemble classifier, and lastly, hybrid ensemble classifier methods are beneficial in a sense that they exploit both the input feature space and the input sample space, allowing them to be more effective in classifying noisy datasets that have small number of samples but higher number of features.

In Chapter 6, a novel cluster balancing strategy is proposed that generates an ensemble classifier by training base classifiers on balanced data clusters. The intuition behind this method was that since clustering algorithms work independent of the class labels in the dataset, , the generated data clusters may or may not be class balanced. Moreover, class imbalances in the data will cause the generated data clusters to be imbalanced as well. Any base classifier trained on such data clusters will be detrimental for the ensemble's generalisation ability. For example, if a data cluster is missing data samples from a class, and a base classifier is trained on that data cluster, then such a base classifier will negatively impact the accuracy of the ensemble when its class decision will be combined with others. Therefore, in the proposed cluster balancing strategy, data samples were first separated based on their classes and class-pure data clusters were generated which contained samples from only one data class. Each class-pure data cluster is then balanced by adding samples from other classes which are closest to the cluster centroid, so that every sample in the cluster has some intrinsic similarity to each other. Since all class-pure data clusters are now balanced, class decisions of base classifier's trained on them are safely combined through majority voting. The proposed was not only tested against benchmark datasets from UCI repository, but we also tested on benchmark image datasets. The results were conclusive that the proposed cluster balancing not only works with SVM, but with any base classifier chosen such as CNNs. As illustrated in Figure 12 each dataset has different characteristics, some datasets are sparse, and some are dense. For example, the *Spectfheart* dataset is dense with two distinct regions which are easily separable from each other. Therefore, the proposed method failed to work there, however, with other datasets it was clearly noticeable from Figure 12 that each dataset has a different optimal number of class separations and using the optimal number can generate an ensemble that can achieve the maximum classification accuracy.

In Chapter 7, the proposed method was further extended by generating the optimal number of data clusters for each data class, rather than searching for a value of N. This was done by conducting a *Silhouette* analysis of each subset of the data *i.e.* subset that has samples from a single class only. It was concluded through experimentation that each data subset has a different optimal *Silhouette* score and therefore, different numbers of data clusters were generated. The results are summarised in Figure 25 and it can be noted that the number of clusters generated vary from two to 13 for various datasets, and a single value would have ended up in training an ensemble classifier that would not have been performing optimally.

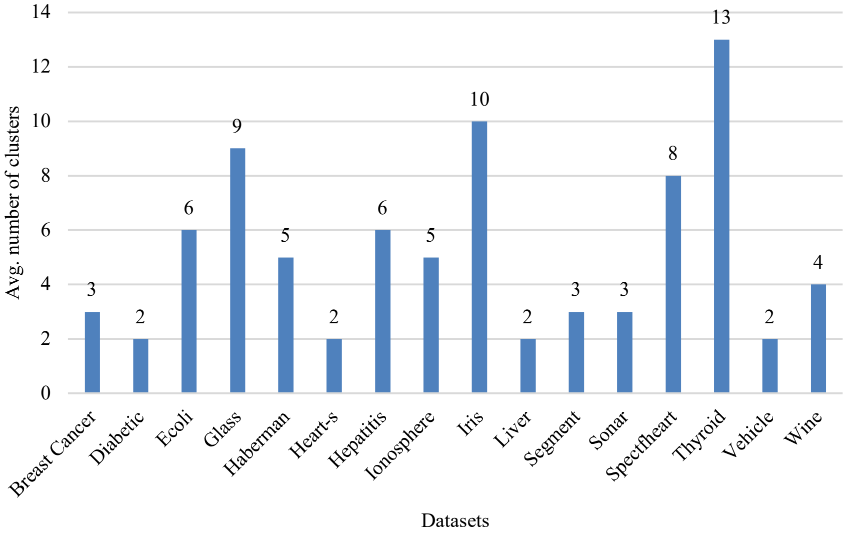


Figure 25: Avg. number of clusters generated for different datasets

In Chapter 8, a classifier selection methodology was proposed that selected base classifiers from the pool using accuracy and diversity comparisons. The intuition was to illuminate the need for an optimisation algorithm by proposing an incremental classifier selection methodology that allows each classifier in the pool to participate in multiple rounds of selection. If a classifier does not perform well in a single round it is not discarded and it still has remaining, chances to participate. It will only be discarded if the base classifier has no more chances left. Through experiments it was observed that only a few base classifiers were enough to achieve the optimum ensemble generalisation ability and adding more would only contribute to the ensemble complexity.

Lastly, in Chapter 9, a pair wise diversity measure was proposed that utilised an incremental classifier selection methodology with the proposed diversity measure. The proposed diversity measure utilised the errors each classifier was making to promote diversity. The proposed classifier selection methodology with the proposed diversity measure not only performed better than existing state-of-the-art ensemble classifier methods but also performed better in comparison to other diversity measures.

Below we summarise the answers to the research questions mentioned in Section 1.3:

- How can evolutionary algorithms be utilised to optimise an ensemble classifier?

  As identified in proposed ensemble methods in Chapters 4 and 5, there are multiple areas where an evolutionary algorithm can be incorporated to optimise upper bounds of clustering, a subset of data clusters and number of base classifiers as discussed below.

  (i) The upper bounds of clustering K to search for the most optimal upper bounds that can achieve the highest classification accuracy;

  (ii) The pool of data clusters that are generated by selecting only a subset of data clusters that can train an intermediate ensemble solution that can achieve the highest classification accuracy on validation data. By optimising the pool of generated data clusters, the need to know the value of K priori is eliminated as the optimisation process selects a subset of data clusters that maximise the ensemble classification accuracy. However, a general safety rule when applying this strategy, is to generate a pool large enough to search for an optimal subset, and in that case a safe upper bound is $K = \sqrt[3]{n}$ [108], where n is the number of samples in a dataset.

  (iii) Lastly, the pool of trained base classifiers can also be optimised to generate an ensemble that can not only achieve the same if not better classification accuracy, but also has a lower component (number of classifiers) size.

  In this thesis various evolutionary algorithms have been tested to optimise the three areas identified, but as was suggested in research, it was proven through experimentation that Particle Swarm Optimisation, typically Binary Particle Swarm Optimisation (BPSO), achieved the highest classification performance because both selecting classifiers or selecting data clusters are considered to be discrete optimisation problems, and in both cases BPSO performed better than other optimisation algorithms.

- What is the optimal number of clusters to generate from training data in order to train diverse classifiers?

  Through extensive experiments it was determined that a fixed value of K for different datasets is not an ideal strategy, and one value of K may work well for one dataset but may not work well for others. Since datasets have different characteristics and attributes, it is not feasible to use a fixed value. If an optimisation algorithm is incorporated to select a subset of data clusters that can maximise the classification accuracy of the ensemble, relatively large values of K are preferable so that a large enough pool is generated for the optimisation process to search in.

  But a point to ponder is how large a value of K can be, simply if $K = n$ would mean generate as many clusters as the number of samples in the dataset then that will end up in generating data clusters with only one sample in it. In such a case $\frac{n(n+1)}{2}$ data clusters will be generated or simply n^2 data clusters. For datasets with many samples for example, $Adult$ dataset having 10,000 samples, this will be computationally taxing, and the complexity can be simplified to $O(n^2)$, or simply exponential time complexity. Such problems are considered NP-hard problems which are difficult to solve in polynomial time, and optimising such a search space will require a lot of resources for example time, hardware, *etc.* Therefore, as suggested, a safe value is setting $K = \sqrt[3]{n}$ which not only generates a large search space for optimisation but also keeps the complexity that can be solved in polynomial time. Another approach, if optimisation algorithm is not incorporated, is to conduct a *Silhouette* analysis to compute the optimal value and select the value of K which has the highest *Silhouette* dissimilarity score. Discussing cluster analysis techniques will be beyond the scope of this research and therefore readers can refer to the respective paper. In this thesis it was found empirically that using *Silhouette* dissimilarity measure scores worked better in terms of ensemble training.

- Which set of classifiers or base learners act as the most optimum set for the ensemble pool?

  Through experiments it was evident that highest performing base classifier was ANN. But there is a limitation as ANNs cannot be trained without parameter optimisation, and base classifiers must be "tweaked" according to the problem they are used to solve. After ANNs the next performing base classifier was SVM, which even in its default form, *i.e.* without any hyper-parameter optimisation, can achieve good classification performance. Also, SVM base classifiers utilise kernel functions which help them in identifying complex decision boundaries effectively. This makes it easier to incorporate SVM base classifiers in ensemble generation, as default implementation can achieve performance that is almost on par to a base classifier whose parameters are optimised. However, when an evolutionary algorithm is incorporated to select the subset of base classifiers from a pool of trained base classifiers, then preference is given to ANN, SVM and KNN more than LDA and DT. One thing to note though is that these classifiers had parameter optimisation done, that is why ANN appeared as the top choice for base classifier. If, however, no parameter optimisation is done, then ANN will be given no preference. This is primarily due to the nature of Neural Networks as they must be tailored to the problem they are solving and there is no general architecture that works well in every approach. In conclusion the more diverse a set of base classifiers is, the better it is for the optimisation algorithm as the search space is more diverse. This adds to the fact that a diverse population converges better than a correlated one.

- How well can the proposed technique perform in comparison with the existing state-of-the-art techniques used for ensemble classifiers?

  The classification performance of the various proposed ensemble classifier method was compared with existing state-of-the-art ensemble classifier methods. For fair comparisons, experiments were conducted in the same environment with Random forest, Boosting, and Bagging. Furthermore, the performance of the proposed methods was also compared with existing published state-of-the-art ensemble classifier methods and various clustering-based ensemble approaches. Through experiments it was evident that the proposed ensemble not only achieved significantly better performance than other state-of-the-art ensemble classifier methods but clustering-based methods as well.

10.2 Future directions

This thesis proposed a novel clustering-based ensemble classifier learning framework. Furthermore, it investigated whether incorporating an evolutionary algorithm to optimise various components of ensemble has a positive effect on ensemble accuracy or not. Various novel ensemble methods that further extended the proposed ensemble framework were also proposed and evaluated.

The methods in this thesis can be further extended, as the training of an optimised clustering-based ensemble classifier has several issues which require investigation. The first area of extension is the effect of incorporating different optimisation algorithms on optimising the pool of base classifiers, and whether different optimisers can achieve different performance. Although it was empirically concluded that meta heuristic algorithms, especially Particle Swarm Optimisation, was particularly effective in being used as a black box optimisation tool to select the base classifiers, there are other algorithms such as Fire Flies, Ant Colony, Fish Swarm, *etc.* that can be utilised to select the best subset of classifiers to generate an optimized ensemble classifier. The same can be done for optimising the clusters pool for random subspace generation.

Furthermore, the novel cluster balancing method that was proposed in this thesis can be extended by incorporating various cluster validation methodologies to find the optimal number of data cluster that must be generated to partition various data classes in the data. In this thesis the effect of using Silhouette analysis was tested, but there are many more cluster validation strategies that can be incorporated. The proposed method was effective not only with benchmark datasets, but also in image classification and shows potential for future research.

Another area which should be further investigated is the practical implications of using the proposed methods in real-world scenarios and on real-world datasets. As real-world datasets have noisy attributes and patterns and it would be very interesting to apply the proposed framework in classifying real-world data.

BIBLIOGRAPHY

[1] M. J. A. Marquis de Condorcet, *Essai sur l'application de l'analyse a la Probabilite des Decisions: Rendues a la Pluralite de voix*. De l'Imprimerie royale, 1785.

[2] M. B. Araujo and M. New, "Ensemble Forecasting of Species Distributions," *Trends Ecol Evol,* vol. 22, no. 1, pp. 42-7, 2007.

[3] Z.H. Zhou, *Ensemble methods: Foundations and Algorithms*. CRC Press, 2012.

[4] T. G. Dietterich, "Ensemble Methods in Machine Learning," *Multiple Classifier Systems,* vol. 1857, pp. 1-15, 2000.

[5] T. G. Dietterich, "Machine-Learning Research," *AI Magazine,* vol. 18, no. 4, p. 97, 1997.

[6] R. Kohavi and D. H. Wolpert, "Bias Plus Variance Decomposition for Zero-one Loss Functions," *International Conference on Machine Learning*, 1996, vol. 96, pp. 275-83.

[7] Y. Ren, L. Zhang, and P. N. Suganthan, "Ensemble Classification and Regression-Recent Developments, Applications and Future Directions," *IEEE Computational Intelligence Magazine,* vol. 11, no. 1, pp. 41-53, 2016.

[8] R. E. Schapire, "The Strength of Weak Learnability," *Machine learning,* vol. 5, no. 2, pp. 197-227, 1990.

[9] J. Kittler, M. Hatef, R. P. Duin, and J. Matas, "On Combining Classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 20, no. 3, pp. 226-239, 1998.

[10] "The KDD Cup." ACM. www.kdd.org (accessed 17/09/2017, 2017).

[11] "Netflix Prize." Netflix. www.netflixprize.com (accessed 17/09/2017, 2017).

[12] Y. Yang and J. Jiang, "Adaptive Bi-Weighting Toward Automatic Initialisation and Model Selection for HMM-Based Hybrid Meta-Clustering Ensembles," *IEEE Transactions on Cybernetic Systems,* vol. 49, no. 5, pp. 1657-1668, 2019.

[13] Y. Q. Zhang, G. Cao, B. S. Wang, and X. S. Li, "A Novel Ensemble Method for K-Nearest Neighbor," *Pattern Recognition,* vol. 85, pp. 13-25, 2019.

[14] M.N. Haque, and P. Moscato, "From Ensemble Learning to Meta-Analytics: A Review on Trends in Business Applications," *In Business and Consumer Analytics: New Ideas,* Springer, 2019, pp. 703-731.

[15] L. Breiman, "Bagging Predictors," *Machine Learning,* vol. 24, no. 2, pp. 123-140, 1996.

[16] Y. Freund and R. E. Schapire, "Experiments with a new Boosting Algorithm," *International Conference on Machine Learning*, 1996, vol. 96, pp. 148-156.

[17] L. Breiman, "Random Forests," *Machine Learning,* vol. 45, no. 1, pp. 5-32, 2001.

[18] M. Wozniak, M. Grana, and E. Corchado, "A Survey of Multiple Classifier Systems as Hybrid Systems," *Information Fusion,* vol. 16, pp. 3-17, 2014.

[19] H. R. Bonab and F. Can, "A Theoretical Framework on the Ideal Number of Classifiers for Online Ensembles in Data Streams," *International on Conference on Information and Knowledge Management*, 2016, pp. 2053-2056.

[20] T. M. Oshiro, P. S. Perez, and J. A. Baranauskas, "How Many Trees in a Random Forest?," *International Conference on Machine Learning and Data Mining*, 2012, pp. 154-168.

[21] P. Latinne, O. Debeir, and C. Decaestecker, "Limiting the Number of Trees in Random Forests," *Multiple Classifier Systems,* pp. 178-187, 2001.

[22] J. A. Lustosa Filho, A. M. Canuto, and J. C. Xavier, "An Analysis of Diversity Measures or the Dynamic Design of Ensemble of Classifiers," *International Joint Conference on Neural Networks, 2015*, pp. 1-8.

[23] R. Lysiak, M. Kurzynski, and T. Woloszynski, "Optimal Selection of Ensemble Classifiers Using Measures of Competence and Diversity of Base Classifiers," *Neurocomputing,* vol. 126, pp. 29-35, 2014.

[24] L. I. Kuncheva and C. J. Whitaker, "Measures of Diversity in Classifier Ensembles and Their Relationship With The Ensemble Accuracy," *Machine Learning,* vol. 51, no.2, pp. 181-207, 2003.

[25] I. Barandiaran, "The Random Subspace Method for Constructing Decision Forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 20, no. 8, 1998.

[26] A. Rahman and B. Verma, "Ensemble Classifier Generation Using Non-Uniform Layered Clustering and Genetic Algorithm," *Knowledge-Based Systems,* vol. 43, pp. 30-42, 2013.

[27] A. Rahman and B. Verma, "Novel Layered Clustering-Based Approach for Generating Ensemble of Classifiers," *IEEE Transactions on Neural Networks and Learning Systems,* vol. 22, no. 5, pp. 781-92, 2011.

[28] M. Asafuddoula, B. Verma, and M. Zhang, "An Incremental Ensemble Classifier Learning By Means of a Rule-Based Accuracy and Diversity Comparison," *International Joint Conference on Neural Networks*, 2017, pp. 1924-1931.

[29] S. Fletcher and B. Verma, "Removing Bias from Diverse Data Clusters for Ensemble Classification," *International Conference on Neural Information Processing*, 2017, pp. 140-149.

[30] L. Hamers, "Similarity Measures in Scientometric Research: The Jaccard Index versus Salton's Cosine Formula," *Information Processing and Management,* vol. 25, no. 3, pp. 315-18, 1989.

[31] D. Huang, C. D. Wang, and J. H. Lai, "Locally Weighted Ensemble Clustering," *IEEE Transactions on Cybernetic Systems,* vol. 48, no. 5, pp. 1460-1473, 2018.

[32] M. Bagheri *et al.*, "Keep It Accurate and Diverse: Enhancing Action Recognition Performance by Ensemble Learning," *Conference on Computer Vision and Pattern Recognition Workshops*, 2015, pp. 22-29.

[33] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A Dataset of 101 Human Actions Classes from Videos in The Wild," 2012.

[34] L. Zhang and P. N. Suganthan, "Oblique Decision Tree Ensemble via Multisurface Proximal Support Vector Machine," *IEEE Transactions on Cybernetic Systems,* vol. 45, no. 10, pp. 2165-76, 2015.

[35] K. S. Bhattacharjee, H. K. Singh, M. Ryan, and T. Ray, "Bridging the Gap: Many-Objective Optimisation and Informed Decision-Making," *IEEE Transactions on Evolutionary Computation,* vol. 21, no. 5, pp. 813-820, 2017.

[36] J. Branke, K. Deb, H. Dierolf, and M. Osswald, "Finding Knees in Multi-Objective Optimisation," *International Conference on Parallel Problem Solving from Nature*, 2004, pp. 722-731.

[37] Z. Q. Qi, B. Wang, Y. J. Tian, and P. Zhang, "When Ensemble Learning Meets Deep Learning: a New Deep Support Vector Machine for Classification," *Knowledge-Based Systems,* vol. 107, pp. 54-60, 2016.

[38] K.-H. Chang and D. S. Parker, "Complementary Prioritised Ensemble Selection," *International Joint Conference on Neural Networks,* 2016, pp. 863-872.

[39] D. Huang, C. D. Wang, J. H. Lai, Y. Liang, S. Bian, and Y. Chen, "Ensemble-Driven Support Vector Clustering: From Ensemble Learning to Automatic Parameter Estimation," *International Conference on Pattern Recognition,* pp. 444-449, 2016.

[40] E. Kilic and E. Alpaydin, "Learning the Areas of Expertise of Classifiers in an Ensemble," *World Conference on Information Technology,* vol. 3, pp. 74-82, 2011.

[41] C.E. Rasmussen, R.M. Neal, G. Hinton, D. Van Camp, M. Revow, Z. Ghahramani, R. Kustra and R. Tibshirani, "Delve data for Evaluating Learning in Valid Experiments," *1995–1996. URL http://www. cs. toronto. edu/～delve.*

[42] S. S. Mao, L. C. Jiao, L. Xiong, S. P. Gou, B. Chen, and S. K. Yeung, "Weighted Classifier Ensemble Based on Quadratic Form," *Pattern Recognition,* vol. 48, no. 5, pp. 1688-1706, 2015.

[43] J.-S. Lee and E. Pottier, *Polarimetric Radar Imaging: From Basics to Applications*. CRC Proess, 2017.

[44] S. Gu and Y. C. Jin, "Multi-train: A Semi-Supervised Heterogeneous Ensemble Classifier," *Neurocomputing,* vol. 249, pp. 202-211, 2017.

[45] K. Kim, H. Lin, J. Y. Choi, and K. Choi, "A Design Framework for Hierarchical Ensemble of Multiple Feature Extractors and Multiple Classifiers," *Pattern Recognition,* vol. 52, pp. 1-16, 2016.

[46] X. C. Yin, K. Z. Huang, H. W. Hao, K. Iqbal, and Z. B. Wang, "A Novel Classifier Ensemble Method with Sparsity and Diversity," *Neurocomputing,* vol. 134, pp. 214-221, 2014.

[47] S. Webb, J. Caverlee, and C. Pu, "Introducing the Webb Spam Corpus: Using Email Spam to Identify Web Spam Automatically," *International Conference on Aerospace Europe*, 2006, pp. 1-9.

[48] J. Abellán and J. G. Castellano, "A Comparative Study on Base Classifiers in Ensemble Methods for Credit Scoring," *Expert Systems with Applications,* vol. 73, pp. 1-10, 2017.

[49] B. Krawczyk and M. Woźniak, "Online Query by Committee for Active Learning From Drifting Data Streams," *International Joint Conference on Neural Networks*, 2017, pp. 2120-2127.

[50] E. Santucci, L. Didaci, G. Fumera, and F. Roli, "A Parameter Randomisation Approach for Constructing Classifier Ensembles," *Pattern Recognition,* vol. 69, pp. 1-13, 2017.

[51] K. W. De Bock and D. Van den Poel, "An Empirical Evaluation of Rotation-Based Ensemble Classifiers for Customer Churn Prediction," *Expert Systems with Applications,* vol. 38, no. 10, pp. 12293-12301, 2011.

[52] H. Kadkhodaei and A. M. E. Moghadam, "An Entropy Based Approach to Find The Best Combination of The Base Classifiers in Ensemble Classifiers Based on Stack Generalization," *International Conference on Control, Instrumentation, and Automation*, 2016, pp. 425-429.

[53] F. Han, D. Yang, Q.H. Ling, and D.S. Huang, "A Novel Diversity-Guided Ensemble of Neural Network Based on Attractive and Repulsive Particle Swarm Optimisation," *International Joint Conference on Neural Networks*, 2015, pp. 1-7.

[54] H. J. Escalante, M. Montes, and E. Sucar, "Ensemble Particle Swarm Model Selection," *International Joint Conference on Neural Networks,* 2010, pp. 1-8.

[55] L.Y Yang, J.Y. Zhang, and W.J. Wang, "Cluster Ensemble Based on Particle Swarm Optimization," *Global Congress on Intelligent Systems, 2009,* vol. 3, pp. 519-523.

[56] U. Bhowan, M. Johnston, M. J. Zhang, and X. Yao, "Evolving Diverse Ensembles Using Genetic Programming for Classification With Unbalanced Data," *IEEE Transactions on Evolutionary Computation,* vol. 17, no. 3, pp. 368-386, 2013.

[57] S. Gu and Y. Jin, "Generating Diverse and Accurate Classifier Ensembles Using Multi-Objective Optimisation," *IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making,* 2014, pp. 9-15.

[58] V. H. A. Ribeiro and G. Reynoso-Meza, "A Multi-Objective Optimization Design Framework for Ensemble Generation," *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 2018, pp. 1882-1885.

[59] Chandrasekaran Sowmya, Friese Martina, Stork JÃűrg, Rebolledo Margarita, and B.B. Thomas. "GECCO Challenge 2017 Monitoring of drinking-water quality." (URL: http://www.spotseven.de/gecco/gecco-challenge/gecco-challenge-2017/)

[60] J. Q. Zhao, L. Jiao, S. Xia, V.B. Fernandes, I. Yevseyeva, Y. Zhou, and M.T. Emmerich, "Multiobjective Sparse Ensemble Learning by Means of Evolutionary Algorithms," *Decision Support Systems,* vol. 111, pp. 86-100, 2018.

[61] B. Zhang, A. Qin, and T. Sellis, "Evolutionary Feature Subspaces Generation for Ensemble Classification," *Proceedings of the Genetic and Evolutionary Computation Conference*, 2018, pp. 577-584.

[62] Z. Yu, Y. Lu, J. Zhang, J. You, H.S. Wong, Y. Wang, and G. Han, "Progressive Semisupervised Learning of Multiple Classifiers," *IEEE Transactions on Cybernetic Systems,* vol. 48, no. 2, pp. 689-702, 2018.

[63] G. G. Yen and Z. N. He, "Performance Metric Ensemble for Multiobjective Evolutionary Algorithms," *IEEE Transactions on Evolutionary Computation,* vol. 18, no. 1, pp. 131-144, 2014.

[64] M.P. Hansen, and A. Jaszkiewicz, "Evaluating The Quality Of Approximations To The Non-Dominated Set," *IMM Technical Report IMM-REP-1998-7, Department of Mathematical Modelling, Technical University of Denmark*, 1994.

[65] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable Multi-Objective Optimisation Test Problems," *Proceedings of the Congress on Evolutionary Computation*, 2002, vol. 1, pp. 825-830.

[66] S. Huband, P. Hingston, L. Barone, and L. While, "A Review of Multiobjective Test Problems and A Scalable Test Problem Toolkit," *IEEE Transactions on Evolutionary Computation,* vol. 10, no. 5, pp. 477-506, 2006.

[67] A. Rosales-Perez, S. Garcia, J. A. Gonzalez, C. A. C. Coello, and F. Herrera, "An Evolutionary Multi-Objective Model and Instance Selection for Support Vector Machines with Pareto-based Ensembles," *IEEE Transactions on Evolutionary Computation,* vol. 21, no. 6, pp. 863-877, 2017.

[68] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera, "KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework," *Journal of Multiple-Valued Logic and Soft Computing,* vol. 17, no. 2-3, pp. 255-287, 2011.

[69] H. J. Escalante, M. Montes, and E. Sucar, "Ensemble Particle Swarm Model Selection," *International Joint Conference on Neural Networks*, 2010, pp. 1-8.

[70] H. J. Escalante, M. Montes, and L. E. Sucar, "Particle Swarm Model Selection," *Journal of Machine Learning Research,* vol. 10, pp. 405-440, 2009.

[71] H. J. Escalante, M. Montes, and L. Villaseñor, "Particle Swarm Model Selection for Authorship Verification," *Iberoamerican Congress on Pattern Recognition*, 2009, pp. 563-570.

[72] H. J. Escalante, M. M. y Gómez, and L. E. Sucar, "PSMS for Neural Networks on The IJCNN 2007 Agnostic ws Prior Knowledge Challenge," *International Joint Conference on Neural Networks*, 2007, pp. 678-683.

[73] J. Kennedy, "Particle Swarm Optimisation," *Encyclopedia of machine learning*: Springer, 2011, pp. 760-766.

[74] K. Deb, *Multi-objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, 2001.

[75] L. Davis, *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York, 1991.

[76] C. Beleites, U. Neugebauer, T. Bocklitz, C. Krafft, and J. Popp, "Sample Size Planning for Classification Models," *Analytica Chimica Acta,* vol. 760, pp. 25-33, 2013.

[77] Z. Yu, D. Wang, Z. Zhao, C.P. Chen, J. You, H.S. Wong, and J. Zhang, "Hybrid Incremental Ensemble Learning for Noisy Real-World Data Classification," *IEEE Transactions on Cybernetic Systems,* vol. 49, no. 2, pp. 403-416, 2019.

[78] Y. Yang and J. Jiang, "Hybrid Sampling-Based Clustering Ensemble With Global and Local Constitutions," *IEEE Transactions on Neural Networks and Learning Systems,* vol. 27, no. 5, pp. 952-65, 2016.

[79] M. Asafuddoula, B. Verma, and M. Zhang, "An Incremental Ensemble Classifier Learning By Means of a Rule-Based Accuracy And Diversity Comparison," *International Joint Conference on Neural Networks*, 2017, pp. 1924-1931.

[80] B. Verma and A. Rahman, "Cluster-Oriented Ensemble Classifier: Impact of Multicluster Characterisation on Ensemble Classifier Learning," *IEEE Transactions on Knowledge and Data Engineering,* vol. 24, no. 4, pp. 605-618, 2012.

[81] L.Y. Yang, J.Y. Zhang, and W. J. Wang, "Cluster Ensemble Based on Particle Swarm Optimisation," *Global Congress on Intelligent Systems*, 2009, vol. 3, pp. 519-523.

[82] H. Kadkhodaei and A. M. E. Moghadam, "An Entropy Based Approach to Find The Best Combination of The Base Classifiers in Ensemble Classifiers Based on Stack

Generalisation," *International Conference on Control, Instrumentation, and Automation*, 2016, pp. 425-429.

[83] F. Han, D. Yang, Q.-H. Ling, and D.-S. Huang, "A Novel Diversity-Guided Ensemble of Neural Network Based on Attractive and Repulsive Particle Swarm Optimisation," *International Joint Conference on Neural Networks*, 2015, pp. 1-7.

[84] K. Bache and M. Lichman. "UCI machine learning repository." (URL: http://archive.ics.uci.edu/ml/).

[85] L. Deng, "The MNIST Database Of Handwritten Digit Images for Machine Learning Research," *IEEE Signal Processing Magazine,* vol. 29, no. 6, pp. 141-142, 2012.

[86] A. Krizhevsky and G. Hinton, "Learning Multiple Layers of Features From Tiny Images," *Citeseer*, 2009.

[87] F. Wilcoxon, "Individual Comparisons by Ranking Methods," *Biometrics bulletin,* vol. 1, no. 6, pp. 80-83, 1945.

[88] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu, "Understanding of Internal Clustering Validation Measures," *IEEE International Conference on Data Mining*, 2010, pp. 911-916.

[89] P. J. Rousseeuw, "Silhouettes: A Graphical Aid to The Interpretation and Validation of Cluster Analysis," *Journal of Computational and Applied Mathematics,* vol. 20, pp. 53-65, 1987.

[90] MATLAB, *Statistics and Machine Learning Toolbox*. Natick, Massachusetts: The MathWorks Inc., 2013.

[91] Z. Jan, B. Verma, and S. Fletcher, "Optimising Clustering to Promote Data Diversity When Generating an Ensemble Classifier," *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 2018, pp. 1402-1409.

[92] L. I. Kuncheva and J. J. Rodriguez, "A Weighted Voting Framework for Classifiers Ensembles," *Knowledge and Information Systems,* vol. 38, no. 2, pp. 259-275, 2014.

[93] Z. Jan and B. Verma, "Evolutionary Classifier and Cluster Selection Approach for Ensemble Classification," *ACM Transactions on Knowledge Discovery in Data,* vol. 1, no. 1, pp. 1-8, 2019.

[94] A. Jurek, Y. X. Bi, S. L. Wu, and C. D. Nugent, "Clustering-Based Ensembles as an Alternative to Stacking," *IEEE Transactions on Knowledge and Data Engineering,* vol. 26, no. 9, pp. 2120-2137, 2014.

[95] Z. Jan and B. Verma, "Ensemble Classifier Optimisation by Reducing Input Features and Base Classifiers," *Proceedings of the Congress on Evolutionary Computation*, 2019, pp. 1580-1587.

[96] W. Yang, K. Wang, and W. Zuo, "Neighborhood Component Feature Selection for High-Dimensional Data," *Journal of Computers* vol. 7, no. 1, pp. 161-168, 2012.

[97] MathWorks, *Deep Learning Toolbox R2019a,* Mathworks: MathWorks, 2017.

[98] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770-778.

[99] A. Akgul, V. Gidla, and B. Willets, "Varied MNIST Kaggle Project.", 2018.

[100] X.X. Niu and C. Y. Suen, "A Novel Hybrid CNN–SVM Classifier for Recognising Handwritten Digits," *Pattern Recognition,* vol. 45, no. 4, pp. 1318-1325, 2012.

[101] C. Poultney, S. Chopra, and Y. L. Cun, "Efficient Learning of Sparse Representations With an Energy-Based Model," *Advances in Neural Information Processing Systems*, 2007, pp. 1137-1144.

[102] P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis," *International Conference on Document Analysis and Recognition*, 2003, vol. 3, no. 2003.

[103] T. Sinha, B. Verma, and A. Haidar, "Optimisation of Convolutional Neural Network Parameters for Image Classification," *Symposium Series on Computational Intelligence* 2017, pp. 1-7.

[104] T. Yamasaki, T. Honma, and K. Aizawa, "Efficient optimisation of convolutional neural networks using particle swarm optimisation," *International Conference on Multimedia Big Data,* 2017, pp. 70-73.

[105] Z. Jan and B. Verma, "A Novel Diversity Measure and Classifier Selection Approach for Generating Ensemble Classifiers," *IEEE Access,* vol. 7, pp. 156360-156373, 2019.

[106] H. Jamalinia, S. Khalouei, V. Rezaie, S. Nejatian, K. Bagheri-Fard, and H. Parvin, "Diverse Classifier Ensemble Creation Based on Heuristic Dataset Modification," *Journal of Applied Statistics,* vol. 45, no. 7, pp. 1209-1226, 2018.

[107] X. Zhu, Z. Ni, M. Cheng, F. Jin, J. Li, and G. Weckman, "Selective Ensemble Based on Extreme Learning Machine and Improved Discrete Artificial Fish Swarm Algorithm for Haze Forecast," *Applied Intelligence,* vol. 48, no. 7, pp. 1757-1775, 2018.

[108] S. Fletcher, B. Verma, Z. M. Jan, and M. Zhang, "The Optimized Selection of Base-Classifiers for Ensemble Classification using a Multi-Objective Genetic Algorithm," *International Joint Conference on Neural Networks,* 2018, pp. 1-8, 2018.

[109] M.N. Haque, N.J. de Vries, and P. Moscato,"A Multi-objective Meta-Analytic Method for Customer Churn Prediction," *Business and Consumer Analytics: New Ideas*, 2019, pp. 781-813.