Evolutionary Classifier and Cluster Selection Approach for Ensemble Classification

Zohaib Md. Jan and Brijesh Verma
Center for Intelligent Systems, Central Queensland University, email: {z.jan, b.verma}@cqu.edu.au

ABSTRACT

Ensemble classifiers improve the classification performance by combining several classifiers using a suitable fusion methodology. Many ensemble classifier generation methods have been developed that allowed the training of multiple classifiers on a single dataset. As such random subspace is a common methodology utilized by many state-of-the-art ensemble classifiers that generate random subsamples from the input data and train classifiers on different subsamples. Real-world datasets have randomness and noise in them, therefore not all randomly generated samples are suitable for training. In this paper, we propose a novel particle swarm optimization-based approach to optimize the random subspace to generate an ensemble classifier. We first generate a random subspace by incrementally clustering input data and then optimize all generated data clusters. On all optimized data clusters, a set of classifiers is trained and added to the pool. The pool of classifiers is then optimized and an optimized ensemble classifier is generated. The proposed approach is tested on 12 benchmark datasets from the UCI repository and results are compared with current state-of-the-art ensemble classifier approaches. A statistical significance test is also conducted and an analysis is presented.

CCS CONCEPTS

• Computing Methodologies • Machine Learning • Ensemble Methods

KEYWORDS

Multiple Classifier Systems; Clustering; Particle Swarm Optimization

1 Introduction

An ensemble classifier is a machine learning classification methodology aimed to improve the classification performance of single classifier models by combining several classifiers together. Ensemble classifiers have been widely adopted in several areas including the health sector, environmental sciences, weather forecasting and financial sector. Ensemble classifiers can outperform single classifier models because their classification decisions are not based on a single classifier rather a combination of multiple diverse classifiers. Moreover, a single classifier performing well on one dataset may not perform well on others, this is known as the "no free lunch theorem" [1]. An ensemble of classifiers consists of multiple weak learners (classifiers), therefore they are very robust and versatile. When combining classifiers to form an ensemble it has been shown in research that combinations of diverse and accurate classifiers can achieve better classification performance only if accurate classifiers are given preference [2-4]. Similarly preference given to diversity only may result in an ensemble classifier which is diverse in nature but performs inaccurately for unseen data and this is known as the "bias-variance co-variance" tradeoff and a balance must be maintained [3, 5]. The bias-variance decomposition theory states that a classifier's generalization error with respect to a problem can be decomposed into bias and variance [6]. Bias is errors due to inaccurate generalization by the classifier, whereas, variance is the sensitivity of generalization due to miniscule changes in the training set. Put simply, bias is a measure of underfitting, and variance is a measure of overfitting.

There are two main steps involved when generating ensemble classifiers: i) training of multiple base classifiers either homogeneous or heterogeneous on sub-samples of the data also known as the random subspace method and ii) suitably combining or fusing the class decisions of all trained base classifiers on the

unseen data using a fusion methodology such as majority voting to get the final decision of the ensemble. An important factor when generating ensembles is the size of the ensemble itself that is the number of base classifiers in the ensemble also plays an important role. It has been found that adding more than an optimal number of classifiers in an ensemble only adds to the computational complexity of the ensemble, if not affecting the accuracy negatively [7]. An ensemble cannot achieve higher classification accuracy if more accurate classifiers are added to it after an optimal number as the classification accuracy of the ensemble tends to plateau and this is known as the "law of diminishing returns" [8]. Therefore, only an optimum number of accurate and diverse base classifiers should be fused together to generate an ensemble that can achieve high classification accuracy.

Two state-of-the-art ensemble classifiers that generate a random subspace to train base classifiers are bagging and boosting [9, 10]. Bagging works by creating "bags" which are basically random subsamples created from the data containing a unique and repeating set of samples and trains multiple base classifiers on all generated bags. Classification decisions of all trained base classifiers are fused and an ensemble is formed. Boosting on the other hand works by successively retraining base classifiers on the data samples that a previously trained classifier performed poorly on. Many different variations of boosting have been proposed in research such as AdaBoost and it is detailed in [11-13]. Another strategy is to generate random sub-samples of the input features rather than input samples and an ensemble classifier that utilizes this technique is Random Forest (RaF) [14]. RaF creates subsets of input features and trains Decision Trees (DT) on feature subsets. Classification decisions of all DTs are combined, and an ensemble is formed. RaF is a very robust and versatile classifier and has been particularly applicable in classifying noisy real-world datasets. Many variations of RaF have been proposed in the literature , for example, Multisurface Proximal RaF (MPRaF) [15] which uses oblique DTs to generate a RaF. In a benchmark study of ensembles [16], it was concluded that MPRaF outperformed most of the existing state-of-the-art ensemble classifiers. Further random subspace-based ensembles are discussed in [17]. Some authors have utilized clustering [18] to generate data clusters from the input data. On all generated data clusters, a set of diverse base classifiers is trained and classification decisions of all classifiers in the pool is fused using majority voting. Since the generated clusters have a unique and repeating set of samples this incorporates diversity in the ensemble, as the base classifiers trained on a cluster will be diverse as well. Further examples of such works are given in [19-24].

Some authors have classified ensemble classification as an optimization problem and have incorporated various strategies to optimize the various components of an ensemble. For example, in [25] authors suggested a Multi-Objective (MO) sparse ensemble learning through the incorporation of an Evolutionary Algorithm (EA). MO problem is formulated using sparsity ratio, false positive rate, and false negative rate. In [26] authors classified ensemble classifier as a many objective optimization problem through the incorporation of clustering. The many objective optimization is formulated based on class accuracies, ensemble component size and diversity of the ensemble. In [27] authors categorized time series forecasting using ensemble learning as a MO optimization problem to diagnose power transformer failures. The fitness was computed based on the chosen time-series forecasting algorithm, the accuracy of the ensemble and the diversity of the algorithms. They used 23 time series forecasting algorithms which were represented by a binary chromosome vector. Similarly, in [28] authors employed a binary version of MO Particle Swarm Optimization (MOPSO) as a feature subset selection and ensemble classifier selection for diagnosing power transformer failures due to dissolved gas. In [29] authors employed evolutionary genetic programming strategy to evolve a Heterogeneous Flexible Neural Tree (HFNT). HFNT is used analogously used for Multi-Layer Perceptron Neural Network (MLP-NN), where the structure of the tree is first evolved starting with a random tree, then its hyperparameters represented as genotype are optimized through parameter optimization. The fitness of a tree or the phenotype was the approximated error, tree size/complexity and diversity index. Further there are ensemble classifier approaches where researches have utilized different optimization techniques. Genetic Algorithms (GAs) are discussed in [30-32], MOs are discussed in [33-35], EAs are discussed in [36, 37] and PSOs are discussed in [38-41].

Generating a pool of diverse and accurate classifiers is only half of the solution when it comes to ensemble learning. Different strategies must be employed to select either the entire pool of classifiers to generate an ensemble or the best subset of classifiers from the pool. These strategies can be generalized as (i) static

classifier selection and (ii) dynamic classifier selection strategies [42]. Static selection-based ensemble classifiers first generate a pool of classifiers and then selects the entire pool of classifiers on all test patterns to generate the ensemble. On the other hand, dynamic selection-based ensemble classifiers use different strategies such as optimization algorithms, rule-based strategies, *etc.* to select the best subset of classifiers from the pool that can maximize the generalization ability of the ensemble. Since dynamic ensemble selection works by adding classifiers in the ensemble from the pool based on the level of competence therefore majority voting is a suitable classifier fusion strategy and such ensemble strategies outperform static ensemble selection strategies in generalization [42, 43]. As such PSO particularly has been very applicable in generating dynamic ensemble classifiers firstly due to the nature of the algorithm as it requires little to no information about the optimization problem and it has been utilized as a model selection tool (selecting classifiers from the pool) to generate ensembles by different researchers [44, 45].
Although clustering and PSO have been used previously in ensemble classifiers, a number of things as listed below need further investigation: i) not all generated data clusters are suitable to train base classifiers because due to randomness and noise a cluster might contain noisy data samples and any base classifier trained on such data cluster will negatively impact the ensemble and such clusters should be discarded; ii) not all base classifiers from the pool should be added to the ensemble, firstly because more doesn't necessarily mean better, secondly adding redundant classifiers can have a negative impact on the diversity of the ensemble, and weekly trained classifiers should also be removed from the pool and should not be included in the ensemble. Therefore, in this paper we propose a novel ensemble classifier generation framework by incorporating an evolutionary algorithm such as PSO to i) optimize all generated data clusters to create a rich and diverse input space for training, and ii) optimize all trained base classifier by incorporating a PSO to generate an ensemble classifier. The main reasons of using PSO as an optimization tool are firstly, as discussed in the literature PSO is an effective model selection tool and can be utilized to dynamically select classifiers/clusters from the search space; secondly, PSO is a meta-heuristic algorithm making it an effective black-box optimization toolbox. The novel contributions of this paper are as follows:

1. A methodology of generating a rich and diverse input space by optimizing all generated data clusters.
2. A methodology of optimizing base classifier pool by incorporating a PSO to generate an ensemble that can achieve maximum classification accuracy and have lower component size as well.

The next section presents the background of the proposed method. Section 3 describes the proposed method. Section 4 shows experiments and comparative analysis. Finally, Section 5 presents a conclusion and future work.

2 Background

This section presents a background of the proposed ensemble classifier typically random subspace generation through clustering and Particle Swarm Optimization.

2.1 Random Subspace Through Clustering

Random subspace method also known as attribute bagging or feature bagging is a machine learning methodology that endeavors to lessen the connections between classifiers in an ensemble by training them on different subsamples of the data [46]. The intuition behind this is the "law of large numbers", which states that for a given population if enough random samples are generated the average distribution of the samples will eventually follow the distribution of the actual population. Ensemble classifier creates random subspaces to "perturb" the input data, later these "perturbed" subsets of the input data are utilized for training base classifiers. A common perturbation strategy is bagging, which is short for bootstrap aggregating. Bagging can be classified as attribute bagging [47] which is a type of subspace generation methodology where subsamples are created from the data patterns and feature bagging [48] which is a type where subsets of input features are created. Since base classifiers are trained on different subsets of either the input data or input

features, they are uncorrelated to each other and have different strengths and weaknesses therefore they contribute to the overall diversity of the ensemble.

Clustering is an un-supervised machine learning classification methodology that aims to partition data samples into k partitions with each sample belonging to a cluster with the nearest mean [49]. Clustering aims to group together a set of similar samples in a similar group (cluster). Clustering has been widely adopted in different areas of data sciences and machine learning. Clustering has been exhaustively used with ensemble classifiers because of their ability to generate dense mutually exclusive data clusters. This is further exploited in a number of works in ensemble classifiers [18–24], where the authors have incrementally generated data clusters from a given input and trained multitude of classifiers on all generated data clusters. The benefit of this is in two folds: firstly, clustering utilizes the spatial information to partition the input data into sparse data clusters. This is done in cartesian coordinate system using either L1 distance or L2 distance measurement. This allows an ensemble to partition a complex decision boundary into dense local regions where on each data cluster a base trained to learn the local ideally linearly separable decision boundary instead of learning a rather complex decision boundary in a higher dimension. Learning a complex decision boundary by training base classifiers on dense local regions can be thought of as the reverse of using a kernel function. Secondly benefit of clustering is that it enables an ensemble classifier to generate a large pool of trained classifiers that are not only trained on unique data patterns but are also structurally different. This contributes to the data diversity of the ensemble as well as the classifier diversity. An illustration of random subspace generation through is given in figure 1. In figure 1 on each data cluster a single Neural Network (NN) base classifier is trained which in turn generates a pool of diverse NNs that can be suitably combined to generate an ensemble.

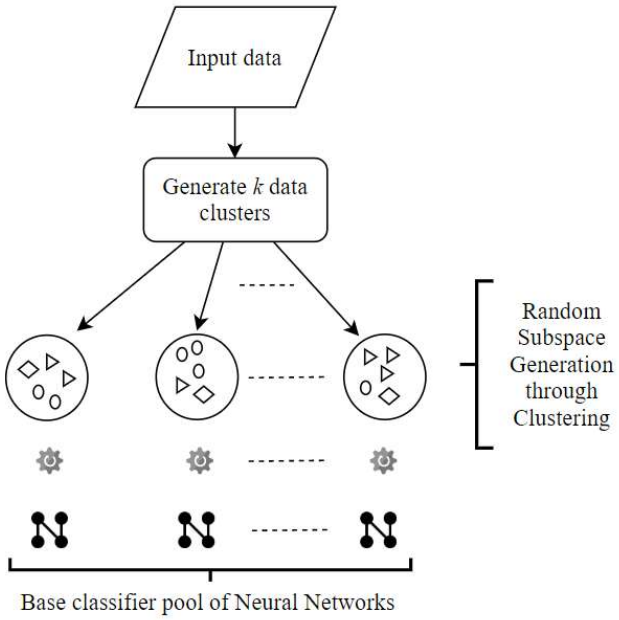

Figure 1: Random Subspace Generation through Clustering

2.2 Particle Swarm Optimization

Particle Swarm Optimization (PSO) [43] is an optimization algorithm that aims to find a candidate solution by iteratively trying to improve a certain objective. PSO is a metaheuristic algorithm that requires little to no information about the problem to optimize allowing it to be widely applicable. PSO is a population-based algorithm in which a swarm of particles depicts the behavior of a flock of birds with a common goal. Each particle has a personal best and a global best solution. Iteratively PSO optimizes the search space to find a solution where no change in the personal best can achieve a better global best. Each particle in PSO modifies

its position based on the personal best, the global best, the velocity and the current position. Further details and mathematical model of PSO can be found in [44]. In ensemble classifiers, PSO can be utilized to optimize the pool of base classifiers or data clusters in which each classifier/cluster correlates to a particle in the population. The global best is the overall ensemble classifier classification accuracy and the personal best is the possible inclusion of a classifier in an ensemble. Following the merits of PSO in ensembles, in this research we utilize PSO to optimize the pool of classifiers and pool of clusters to generate an optimized ensemble

3 Proposed Method

This section describes the proposed ensemble classifier generation framework. First, an overview of the proposed framework is given. Then different components are discussed and illustrated including the proposed PSO based optimization approach. The framework of the proposed ensemble classifier is shown in figure 2.

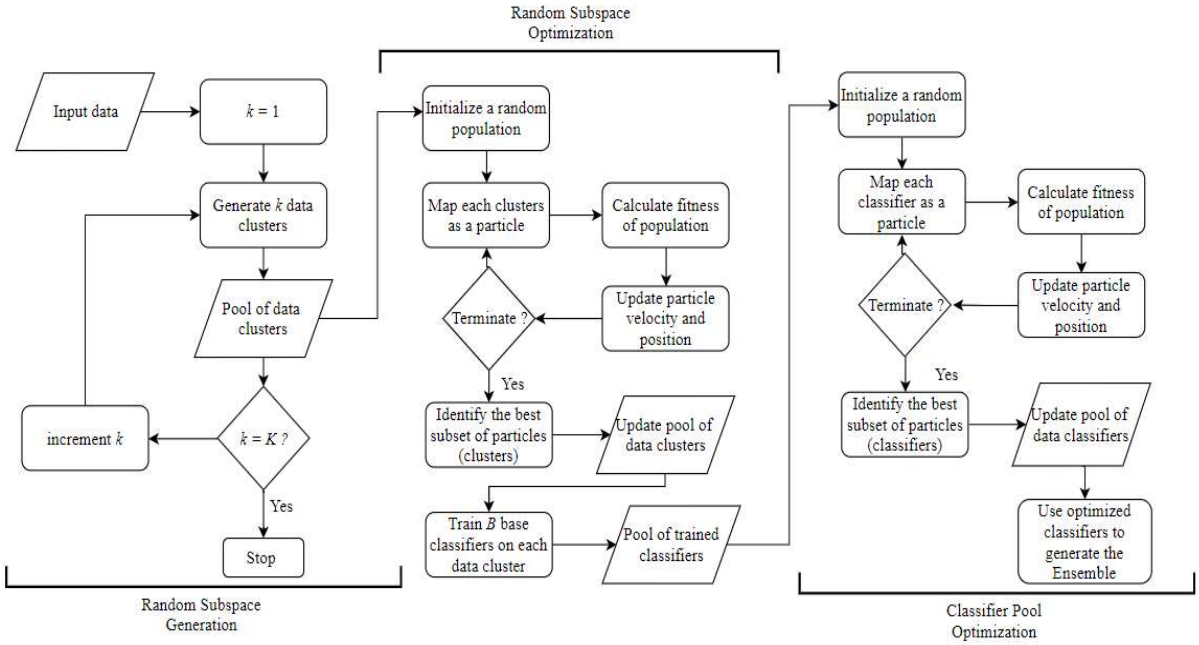

Figure 2: Proposed evolutionary ensemble classifier generation framework

3.1 Ensemble Classifier Generation Framework

The proposed ensemble classifier starts off with a training data and a set of diverse base classifiers *B* (Artificial Neural Networks (ANN), Discriminant Analysis (DISCR), Decision Trees (DT), Support Vector Machine (SVM), Naïve Bayes (NB), K-Nearest Neighbor (KNN), etc.). A diverse subspace is generated through clustering data incrementally by creating a pool of data clusters. A subset of all generated clusters is selected by optimizing the pool with the objective of maximizing the classification accuracy. Base classifiers are trained on all optimized data clusters thus generating the base classifier pool (bcp). The pool is then optimized to select the best subset of base classifiers and an optimized ensemble is generated.

3.2 Diverse Sub-space Generation

Clustering is utilized to generate data clusters from the input data. Generated data clusters form the diverse sub-space for the training of base classifiers.

3.2.1 Generate Data Clusters

Given a dataset $X = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ where $x \in \mathbb{R}^d$, d represents the dimensions of the input feature vector, $y \in \{1,2, \dots, z\}$, z is the number of classes in the dataset, n is the number of samples, we first cluster the dataset incrementally from 1 to n, generating k clusters in each iterations. The clusters are obtained by minimizing the sum of the squared Euclidean distance of each feature vector from the cluster centroid given as:

$$argmin \sum_{i=1}^{n} \sum_{\forall j \in k} \left(euc(x_i, c_j) \right) \qquad (1)$$

where x is a feature vector and c is a cluster centroid and $euc(x, c)$ denotes the squared euclidean distance and is given as:

$$euc(x, c) = \sqrt{\sum_{i=1}^{n} (x_i - c_i)^2} \qquad (2)$$

Euclidean distance is used because the individual features in the data are not interrelated in most of the datasets used in experiments, however, any other distance measurement such as Manhattan distance or Taxicab distance can be utilized.

3.2.2 Optimize Data Clusters

The generated data clusters are optimized by incorporating a PSO. The population of particles in the PSO is the number of clusters in the random subspace which is represented by a binary string 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 optimization is all generated data clusters given as:

$$minimize\ (f_1(C)) \text{ subject } C \in s \qquad (3)$$

Where s is the generated clusters pool, $f_n(C)$ is the fitness function that takes a random subset of C data clusters from s, and computes the average cost by training a base classifier (any base classifier can be used here) on all data clusters in the subset. The cost is computed by first calculating the individual root mean square error ($RMSE$) of each cluster given as:

$$RMSE = \frac{\sqrt[2]{\sum_{q=1}^{n} \left(y'_q - y_q \right)^2}}{|n|} \qquad (4)$$

where V is a validation dataset having y true class labels; y' is the predicted class labels of a classifier, n is the number of samples. Let m be the number of clusters in the population then the cost of the fitness function $f_n(C)$ is given as:

$$f_n(C) = {\sum_{i=1}^{m} RMSE^i} \Big/ {|m|} \qquad (5)$$

The objective function with lowest average $RMSE$ is selected and the remaining data clusters are discarded.

3.3 Base Classifier Pool Generation

For the purpose of this study we have used a set of 6 structurally different and diverse base classifiers (ANN, DT, SVM, KNN, NB, and DISCR) in the training process. The selection of classifiers is independent of the proposed framework and any combination(s) of classifier(s) can be utilized. However, the main intuition of using 6 different classifiers here is that since each classifier has different learning capabilities this contributes to the overall classifier diversity of the ensemble which in turn will generate a diverse pool of trained classifiers for optimization. Each cluster in the pool is utilized to train a set B of base classifiers which are then placed in the base classifier pool bcp. Therefore, if there are s' clusters then a total of $s' \times B$ classifiers are allocated in the pool bcp that will be utilized in the optimization process.

3.4 Base Classifier Pool Optimization

An evolutionary algorithm PSO is incorporated to optimize the pool bcp as well. The problem formulation for optimization is as follows:

$$minimize\ (f(\xi))\ \text{subject to}\ \xi\ \in bcp \qquad (6)$$

Where ξ is a possible ensemble solution containing a random subset of trained base classifiers from the pool bcp given as $\xi = \{B^1, B^2, \dots, B^{bcp}\}$ and the objective function is given as:

$$f(\xi) = f_1 + f_2 \qquad (7)$$

f_1 and f_2 are the two variables of the objective functions. f_1 is the $RMSE$ of the ensemble calculated using equation (4) after computing the predicted class labels y' of the ensemble. Predicted class labels of the ensemble are obtained by combining the predicted class labels of all classifiers in the ensemble using the following equation:

$$y'_\xi = mode(\xi) \qquad (8)$$

where $mode$ is a mathematical operator that depicts majority voting. Majorioty voting is used here because it is established in research that when dynamic classifier selection methodlogy is employed then majority voting outperforms other decision fusion methodologies. Refer to [42, 43] for further details. Additionally, most of the other works of clustering-based ensemble classifier utilized majority voting as well [19–26].

f_2 is the second variable which is simply the size of the ensemble (number of classifiers) given as:

$$f_2 = |\xi| \qquad (9)$$

Ideally the proposed approach identifies the subset of classifiers from the pool to generate an optimized ensemble classifier which can not only achieve the highest classification accuracy but also have lower component size as well. A classifier is mapped to a particle in PSO search space and to binarize the process of classifier selection particle positions are restriction to 1s and 0s, this is achieved by using a threshold θ which converts a position into discrete 1s and 0s. This is done in order to convert the continuous PSO search space into a binary search space because as updating the velocity vector of a particle using the original

equation proposed in vanilla PSO is not suitable for a binary search space optimization. Population in PSO is represented is represented as follows as:

$$pop = [\varphi^1, \varphi^2, \ldots., \varphi^{|bcp|}] \qquad (10)$$

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

Therefore, the population of particles at the end of optimization that have respective 1s are used to identify the classifiers that will be utilized to generate the optimized ensemble classifier. The algorithm of the proposed ensemble classifier framework is given in algorithm 1.

3.5 Ensemble Classifier Generation

The optimization process dynamically searches for the best ensemble solution that can not only achieve the highest classification accuracy but also have the smallest component size as well. The new optimized pool of classifiers bcp' is used to generate the ensemble. All classifiers are utilized to predict the feature vector x of the unseen dataset . The predicted class labels y' of all classifiers in the optimized ensemble are fused using equation (8) (majority voting) and ensemble classification accuracy is calculated.

3.6 Complexity Analysis

We have conducted a theoretical analysis of the proposed ensemble classifier with respect to its computational cost. The complexity of the proposed approach is as follows:

- $T = T_{clustering} + T_{O\_clusters} + T_{O\_bcp}$
- where $T_{clustering}$, $T_{O\_clusters}$, and T_{O_bcp} are the computational costs of generating data clusters, optimizing the pool of data clusters and optimizing the base classifier pool.
- The computation complexity of K-means clustering is $O(ikn)$ where i is the number of iterations, k is the number of clusters and n is the number of data samples. As we iteratively cluster data into 1 to K clusters. the complexity of clustering is as follows:
  - $T_{clusters} = O(i(1 + 2 + \cdots + K)n)$ or simply $T_{clusters} = O(i\,K!\,n)$
- As max number of generated data clusters is $s = K(K+1)/2$ and the worst case complexity of PSO is $O(n^2)$, the complexity of optimizing the pool of data clusters can be as given below:
  - $T_{O_{clusters}} = O(s^2)$
- On each optimized data cluster, assuming the worst-case cost of training a base classifier is $O(n^2)$ therefore, if a set of B base classifiers is trained on all generated data clusters then the cost of generating the bcp is $O(s \times n^2)$.
- Finally, the cost of optimizing the base classifier pool is given as:
  - $T_{O_{bcp}} = O((s \times n^2)^2)$ $\quad or \quad$ $T_{O_{bcp}} = O(s^2 \times n^4)$
- Therefore, the complexity of the proposed approach is the sum of all given as:
  - $O(s^2 \times n^4 + s^2 + i\,K!\,n)$ since n is the largest factor here, we can simplify it as $O(n^4)$ as the worst case complexity of the proposed approach.

Algorithm 1: Random subspace and classifier optimization

Training dataset: $X = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, $x \in \mathbb{R}^d, y \in \{1,2, \ldots, c\}$,
Validation dataset: $V = \{(x_1, y_1), \ldots, (x_n, y_n)\}$
Input: A set of base classifiers B, upper bounds of clustering K
Output: Optimized ensemble classifier ξ

1. **for** $k = 1$ to K do
2. $C^S \leftarrow$ partition training dataset into k clusters by minimizing the squared Euclidean distance of each sample from the cluster centroid
3. Increment $s = s + 1$
4. **endfor**
5. Initialize a population of $|s|$ particles
6. **while** termination criteria
7. Map each particle to a cluster in the search space
8. Calculate the fitness of the population using equation (5) by training a base classifier on all clusters in the population on validation dataset V
9. Update the local best and global best of the population
10. Update particle velocity and position
11. **endwhile**
12. Discard redundant/noisy clusters and use the new optimized clusters s'
13. **for** $j = 1\ to\ |s'|$ do
14. $bcp_k(B, C^j) \leftarrow$ train a set of base classifiers B on generated data cluster using training data set X and them to the pool bcp
15. Increment $j = j + 1$
16. **endfor**
17. Initialize a population of $|bcp|$ particles
18. **while** termination criteria
19. Map each particle to a classifier B in the population by generating a binary bit string pop representing Each classifier in the population
20. Generate an ensemble solution ξ of classifiers that have a higher value than the threshold θ
21. Calculate the fitness of the population using equation (7) with validation dataset V
22. Update the local best and global best of the population
23. Update particle velocity and position
24. **endwhile**
25. Use the new optimized 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.

4 Experiments and results

In this section, we have conducted experiments to evaluate the performance of the proposed ensemble classifier. Details of the benchmark datasets, experimental setup, performance analysis and comparative analysis are given.

4.1 Datasets

The generalization performance of the proposed ensemble classifier framework was evaluated on 12 benchmark datasets collected from the UCI [52] dataset repository. The details of the datasets are shown in Table 1.

Table 1: Number of samples, class labels, and classes of the UCI benchmark datasets used in experiments

Dataset	# of samples	# of variables	# of classes
Breast Cancer	699	9	2
Diabetic	768	8	2
E.coli	336	7	8
Haberman	306	3	2
Ionosphere	351	33	2
Iris	150	4	3
Liver	345	6	2
Segment	2310	19	7
Sonar	208	60	2
Thyroid	7200	21	3
Vehicle	946	18	4
Wine	178	13	3

It can be noted from Table 1 that a mix of datasets was selected for experiments. These datasets have been used in other similar works, therefore allowing comparative analysis.

4.2 Experimental Setup

MATLAB 2017 R1 was used to implement the proposed ensemble classifier framework and conduct experiments (soure code available at https://github.com/zohaibjan/ensemble_works_TKDD). A 10-fold cross-validation data partitioning method was applied to incorporate for randomness as in other similar research and 10 independent runs were conducted. For base classifiers a multi-class version of SVM with Gaussian kernel, vanilla DT with leaf size random between c and $\mathrm{c} * 10$ where c is the number of classes in the dataset, default ANN with 10 hidden-layers and random *epochs* between 100 and 5000, vanilla KNN with k being random between 1 and 10, vanilla NB, DISCR with *diaglinear* discriminant type, were utilized while conducting experiments. The random choice of parameters was set optimistically using trial and error that can achieve maximum ensemble accuracy. This was done to incorporate further classifier diversity in the ensemble as different classifier will not only be trained on different clusters but will have different parameter settings as well. For clustering input data default implementation of k-means with 2400 maximum iterations was used. The upper bounds of clusters K was set to $\sqrt[3]{n}$ where n is the number of samples in a dataset similar to [17]. Default implementation of PSO in Matlab "*particleswarm*" was used for optimization. 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 [17] 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.

The classification accuracy was calculated over the testing dataset using the following equation:

$$ac = \frac{\sum_{x_i \in T_s} \{y_i' = y_i^{true}\}}{|T_s|} \qquad (11)$$

where y' represents the predicted class labels of the ensemble using the unseen test set T_s, y represents the true class labels, and x_i represents an input feature vector.

4.3 Performance Analysis of the Proposed Ensemble Classifier Framework

The average classification accuracy and standard deviation of the proposed ensemble classifier on 12 benchmark datasets are given in Table 2.

Table 2: Classification accuracy and standard deviation of the proposed ensemble classifier on 12 UCI benchmark datasets

Datasets	Proposed Approach	Std. Deviation
Breast Cancer	0.9640	0.00248
Diabetic	0.7700	0.00145
E.coli	0.9510	0.00032
Haberman	0.7480	0.00202
Ionosphere	0.9170	0.00328
Iris	0.9730	0.01429
Liver	0.7180	0.00993
Segment	0.9860	0.00314
Sonar	0.8320	0.00543
Thyroid	0.9640	0.00025
Vehicle	0.9030	0.00303
Wine	0.9920	0.00226

It can be noted from Table 2, that the proposed ensemble classifier achieved an average classification accuracy of 90.30% on 12 datasets, with most of the results lying in the first standard deviation. We have

also investigated the effect of incorporating a set of diverse base classifiers, and clustering on the overall ensemble diversity. To compute diversity of the ensemble we have used "*disagreement*" measure originally proposed in [3] (for further details please refer). The measure is calculated as follows:

$$Dis_{i,j} = \frac{N_{01} + N_{10}}{N_{11} + N_{10} + N_{01} + N_{00}} \qquad (12)$$

where N_{ij} is the dissimilarity between classifiers i and j. 0 signifies wrongly classified sample, and 1 signifies correctly classified label on a dataset. The classifiers in the pool bcp' were utilized to generate predicted class labels and the average is computed as given in equation (13). The results are shown in Table 3.

$$avgDiv = \sum_{i=1}^{bcp-1} \sum_{j=i+1}^{bcp} Dis_{i,j} \qquad (13)$$

Table 3: Diversity comparison of using clustering and diverse base classifiers to generate an ensemble classifier

Datasets	Diversity without clustering	Diversity with clustering
Breast Cancer	0.021	0.422
Diabetic	0.403	0.463
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
Segment	0.081	0.408
Sonar	0.261	0.425
Thyroid	0.046	0.163
Vehicle	0.253	0.465
Wine	0.105	0.454

It can be noted from Table 3 that the average diversity without clustering and a diverse set of base classifiers is 0.17 and average diversity with clustering and a diverse set of base classifiers is 0.39, which is a significant increase in the diversity of the ensemble. However, the increase in diversity is only beneficial if it has a positive effect on the overall accuracy. Therefore, Table 4 entails the classification accuracy of the proposed ensemble framework with and without the incorporation of optimization.

Table 4: Classification accuracy of the proposed ensemble classifier with and without the incorporation of an evolutionary algorithm

Datasets	Optimized ensemble	Non-optimized ensemble
Breast Cancer	96.77	90.40
Diabetic	77.89	76.70
E.coli	95.10	91.70
Haberman	76.70	74.50
Ionosphere	91.70	90.00
Iris	97.30	79.60
Liver	71.80	70.10
Segment	98.60	92.10
Sonar	91.90	78.40
Thyroid	96.40	96.20
Vehicle	90.30	89.80
Wine	99.20	95.20

It can be noted from Table 4 that the average classification accuracy of the optimized ensemble classifier is 90.30%, whereas the average classification accuracy of the non-optimized ensemble is 85.39%. It is evident from the results that the optimized ensemble not only achieved 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 was also affecting the overall classification accuracy of the ensemble to suffer from "*diminishing returns*".

We have also evaluated the effect of incorporating an evolutionary algorithm on reducing the number of clusters that are utilized 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 base classifiers trained on such clusters will not perform well on the unseen dataset (hold-out sample) and eventually will negatively impact the ensemble. Therefore, such data clusters are discarded from the generated input space. The results are given in Table 5 and it can be noted from Table 5 that on average a total of 44.66 clusters were generated of which 21.83 clusters were utilized which means an average of 15.83% cluster reduction was achieved.

Table 5: Cluster reduction achieved through incorporation of an evolutionary process

Datasets	Average clusters generated	Average clusters utilized
Breast Cancer	37.0	23.8
Diabetic	46.0	27.8
E.coli	22.0	11.7
Haberman	22.0	14.7
Ionosphere	29.0	17.8
Iris	16.0	10.0
Liver	29.0	19.5
Segment	79.0	33.0
Sonar	22.0	14.1
Thyroid	172	51.7
Vehicle	46.0	27.7
Wine	16.0	10.2

4.4 Comparative Analysis

We have analyzed and compared the classification accuracy of the proposed ensemble with single classifier approaches, legacy state-of-the-art ensemble classifiers such as bagging and boosting, existing clustering-based ensemble approaches and recent state-of-the-art ensemble classifiers. The classification accuracies were taken directly from the respective papers.

The average classification accuracy on 12 benchmark datasets of the proposed ensemble classifier and single classifier approaches is given in figure 3. From figure 3 it can be noted that the proposed ensemble classifier was able to achieve 8.58% performance improvement over ANN, 5.96% over SVM, 6.80% over KNN, 5.02% over DISCR, and 7.39% over DT. For the sake of simplicity, the average classification accuracy of the single classifier is given which was taken from the paper [19].

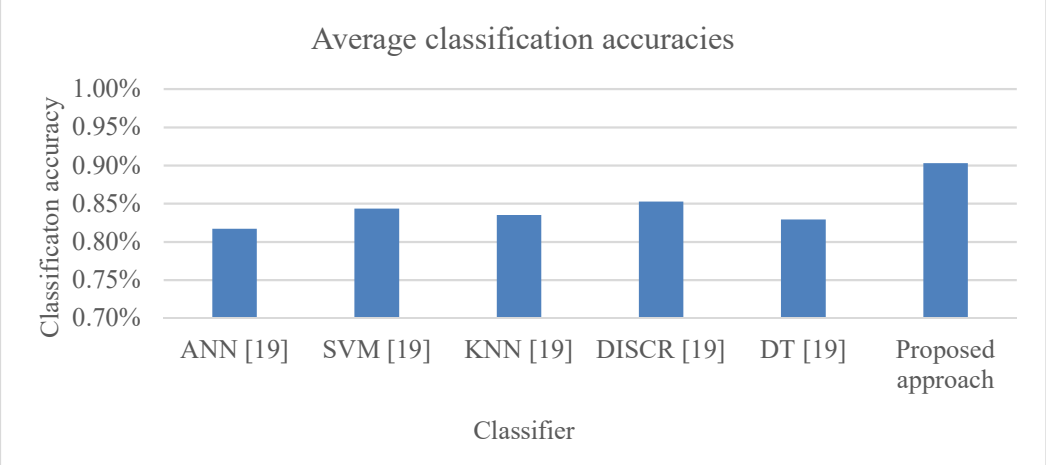

Figure 3: Average classification accuracy of the proposed approach, and single classifier approaches

The performance of the proposed ensemble classifier is also compared with legacy ensemble classifiers bagging and boosting. The results are given in Table 6 and it can be noted from table 6 that the proposed ensemble achieved higher classification accuracy in 8 out of 12 datasets gaining an average performance increase of 2.35% over bagging and 3.78% over boosting.

Table 6: Classification performance of the proposed approach, bagging, and boosting

Dataset	Proposed approach	Bagging [19]	Boosting [19]
Breast cancer	0.9677	0.9709	0.9694
Diabetic	0.7789	0.7602	0.7462
E.coli	0.9510	0.8867	0.8890
Haberman	0.7670	0.7420	0.6637
Ionosphere	0.9170	0.9136	0.9000
Iris	0.9730	0.9667	0.9733
Liver	0.7180	0.7024	0.7071
Segment	0.9860	0.9680	0.9572
Sonar	0.9190	0.8551	0.8266
Thyroid	0.9640	0.9682	0.9682
Vehicle	0.9030	0.8424	0.8096
Wine	0.9920	0.9778	0.9722

The classification performance of the proposed ensemble classifier was also compared with a recent ensemble classifier that generates a random subspace through clustering and incorporates a rule-based accuracy and diversity comparison OEC-ILC. The results are given in figure 4 with classification accuracies taken from the respective paper [19].

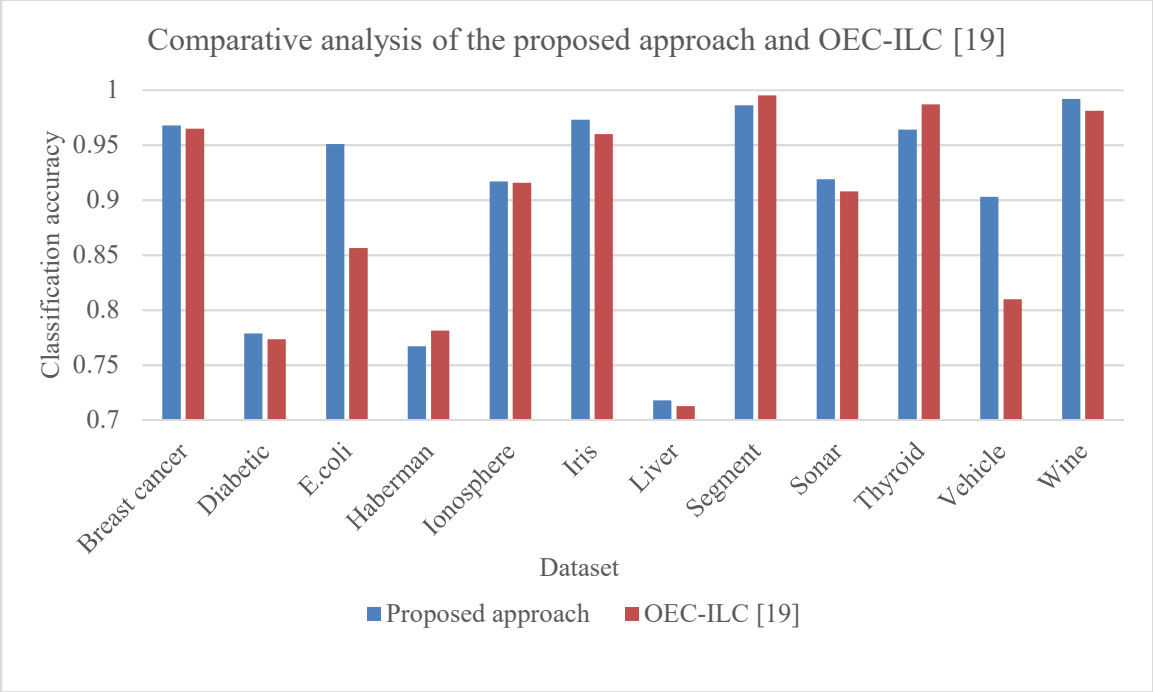

Figure 4: Classification accuracy of the proposed ensemble classifier and OEC-ILC

It can be noted from figure 4 that the proposed ensemble classifier outperformed OEC-ILC in 9 out of the 12 datasets achieving an average of 1.59% performance gain. Additionally, the classification performance of the proposed ensemble approach is compared with other clustering-based ensemble approaches given in [54]. The results are given in Table 7 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.

Table 7: Classification accuracy of the proposed ensemble classifier and OEC-ILC

Dataset	Proposed approach	STLR [54]	CBCA [54]	MV [54]	STJ48 [54]	CL [54]
Breast cancer	0.967	0.960	0.970	0.965	0.963	0.970
Haberman	0.767	0.738	0.765	0.725	0.728	0.734
Iris	0.973	0.945	0.972	0.951	0.952	0.924
Segment	0.986	0.965	0.949	0.960	0.961	0.853
Sonar	0.919	0.849	0.790	0.781	0.756	0.756

Lastly, we have also compared the classification performance of the proposed ensemble classifier with a progressive semisupervised ensemble of multiple classifiers proposed in [55]. The classification accuracies are taken directly from the respective paper and results are given in Table 8 and it can be noted that on average the proposed ensemble achieved 15.58% performance improvement.

Table 8: Classification accuracy of the proposed ensemble classifier and PSEMISEL

Dataset	Proposed approach	PSEMISEL [55]
Diabetic	0.7789	0.6311
Iris	0.9730	0.8961
Segment	0.9860	0.9226
Vehicle	0.9030	0.6179
Wine	0.9920	0.9366

4.6 Significance Testing

To further validate the efficacy of the results we have adopted a parametric test [56] to compare the performance of the proposed ensemble classifier approach with single classifier approaches, clustering based ensemble approaches, a rule-based ensemble approach and a semi-supervised ensemble of multiple classifiers. The Wilcoxon Signed Rank test [56] can be used to answer the question whether the experimental results obtained are statistically significant or not. The tests are conducted with an alpha value of 0.05 and the results are listed in Table 9.

Table 9: P-values of non parametric signed rank tests

Classifier	p-values
ANN	0.0014
SVM	0.0018
KNN	0.0011
DISCR	0.0018
DT	0.0011
Bagging	0.0030
Boosting	0.0048
OEC-ILC	0.1196
STLR	0.0215
CBCA	0.1124
MV	0.0215
STJ48	0.0215
CL	0.0398
PSEMISEL	0.0138

The null hypothesis can be rejected at a *p*-value less than 0.05 signifying that the experimental results are not merely by chance but are statistically significant and the alternate hypothesis can be accepted with 95% confidence that the proposed approach is significantly better. It can be noted from Table 9 that most of the results are statistically significant besides that of OEC-ILC and CBCA. Although the proposed ensemble classifier did achieve performance improvement of 1.59% and 5.24% respectively, more experiments are required to further validate the efficacy of the results.

5 CONCLUSIONS

This research proposed a novel ensemble classifier framework that incorporates an evolutionary algorithm to generate an optimized ensemble classifier. Firstly, a random subspace is generated through clustering data incrementally generating a pool of data clusters. All clusters are then optimized by incorporating an evolutionary algorithm and only clusters which can maximize the classification accuracy of a base classifier are selected. Redundant and noisy data clusters are discarded. The optimized pool of clusters is then utilized to train a set of diverse base classifiers to generate the base classifier pool. All classifiers are optimized by incorporating a multi-constrained PSO which generates an optimized ensemble classifier that cannot only achieve higher classification accuracy but also has a lower component size. To evaluate the performance of the proposed ensemble classifier, 12 benchmark datasets from UCI repository were used and results were compared with single classifier approaches as well as state-of-the-art ensemble classifier approaches.

From the results following observations can be made: i) by optimizing the generated subspace a 15% cluster reduction was achieved, which means that not all generated data clusters were suitable for training a base classifier such as redundant and noisy data clusters; ii) adding more classifiers to the ensemble does not increase the overall classification accuracy of the ensemble and optimizing the base classifier pool on the basis of classification accuracy and ensemble size resulted in an ensemble that comprises of optimum number of classifiers but can achieve the same if not higher classification accuracy than an ensemble generated without optimization; iii) using a diverse set of classifiers to generate a base classifier pool is more efficient than using classifiers of the same type, as this contributes to the overall diversity of the ensemble which can achieve higher classification performance; iv) random subspace generation through clustering is not only a methodology which enables the training of multiple base classifiers on a single dataset but also contributes to overall diversity of the ensemble. Since base classifiers are trained on different data clusters, they have different learning capabilities.

In the future, we would like to extend our work by conducting experiments on more real world and benchmark datasets. We would also like to experiment with different optimization algorithms and compare the effect on ensemble classifier accuracy if a different optimizer is used.

ACKNOWLEDGMENTS

This research was supported by the Australian Research Council's Discovery Project funding scheme ARC-DP-160102639.

REFERENCES

[1] D. H. Wolpert and W. G. Macready (1997). "No free lunch theorems for optimization". IEEE Transactions on Evolutionary Computation, 1(1), 67-82.

[2] Z.-H. Zhou (2012), Ensemble methods: foundations and algorithms. CRC Press.

[3] T. G. Dietterich (2000). "Ensemble methods in machine learning". Multiple Classifier Systems, vol. 1857, pp. 1-15, 2000.

[4] T. G. Dietterich (1997). "Machine-learning research". AI Magazine, 18(4), 97.

[5] Y. Ren, L. Zhang, and P. N. Suganthan (2016). "Ensemble classification and regression-recent developments, applications and future directions". IEEE Computational Intelligence Magazine, 11(1), 41-53.

[6] R. Kohavi and D. H. Wolpert (1996), "Bias plus variance decomposition for zero-one loss functions". International Conference on Machine Learning, (96), 275-83.

[7] T. M. Oshiro, P. S. Perez, and J. A. Baranauskas (2012). "How many trees in a random forest?". International Conference on Machine Learning and Data Mining, 154-168.
[8] R. W. Shephard and R. Färe (1974). "The law of diminishing returns". in Production Theory, 287-318, Springer, Berlin, Heidelberg.
[9] L. Breiman (1006). "Bagging predictors". Machine learning, 24(2), 123-140.
[10] Y. Freund and R. E. Schapire (1996). "Experiments with a new boosting algorithm". International Conference on Machine Learning, 96, 148-156.
[11] C. Domingo and O. Watanabe (2000). "MadaBoost: A modification of AdaBoost". International Conference on Computational Learning Theory, 180-189.
[12] G. Rätsch, T. Onoda, and K.-R. Müller (2001). "Soft margins for AdaBoost". Machine learning, 42(3), 287-320.
[13] A. Vezhnevets and V. Vezhnevets (2005). "Modest AdaBoost-teaching AdaBoost to generalize better". Graphicon, 12(5), 987-997.
[14] L. Breiman (2001), "Random forests," Machine learning, 45(1), 5-32.
[15] L. Zhang and P. N. Suganthan (2015). "Oblique decision tree ensemble via multisurface proximal support vector machine". IEEE Transactions on Cybernetics, 45(10), 2165-2176.
[16] L. Zhang and P. N. Suganthan (2017). "Benchmarking Ensemble Classifiers with Novel Co-Trained Kernal Ridge Regression and Random Vector Functional Link Ensembles". IEEE Computational Intelligence Magazine, 12(4), 61-72.
[17] B. Zhang, A. Qin, and T. Sellis (2018). "Evolutionary feature subspaces generation for ensemble classification". Proceedings of the Genetic and Evolutionary Computation Conference, 577-584.
[18] J. A. Hartigan and M. A. Wong (1979). "Algorithm AS 136: A k-means clustering algorithm". Journal of the Royal Statistical Society. Series C (Applied Statistics), 28(1), 100-108.
[19] M. Asafuddoula, B. Verma, and M. Zhang (2017), "An incremental ensemble classifier learning by means of a rule-based accuracy and diversity comparison". International Joint Conference on Neural Networks, 1924-1931.
[20] Z. M. Jan, B. Verma, and S. Fletcher (2018). "Optimizing clustering to promote data diversity when generating an ensemble classifier". Proceedings of the Genetic and Evolutionary Computation Conference Companion, 1402-1409.
[21] S. Fletcher and B. Verma (2017). "Removing Bias from Diverse Data Clusters for Ensemble Classification". International Conference on Neural Information Processing, 140-149.
[22] B. Verma and A. Rahman (2012). "Cluster-oriented ensemble classifier: Impact of multicluster characterization on ensemble classifier learning". IEEE Transactions on Knowledge and Data Engineering, 24(4), 605-618.
[23] A. Rahman and B. Verma (2011). "Novel layered clustering-based approach for generating ensemble of classifiers". IEEE Transactions on Neural Networks, 22(5), 781-92.
[24] A. Rahman and B. Verma (2010). "A novel ensemble classifier approach using weak classifier learning on overlapping clusters". International Joint Conference on Neural Networks, 1-7.
[25] J. Zhao, L. Jiao, S. Xia, V.B. Fernandes, I. Yevseyeva, Y. Zhou, and M.T. Emmerich (2018). “Multiobjective sparse ensemble learning by means of evolutionary algorithms”. Decision Support Systems, 111, 86-100.
[26] M. Asafuddoula, B. Verma, and M. Zhang (2017). "A Divide-and-Conquer Based Ensemble Classifier Learning by Means of Many-Objective Optimization". IEEE Transactions on Evolutionary Computation, 762-777.
[27] A. Peimankar, S.J. Weddell, T. Jalal and A.C. Lapthorn (2018). “Multi-objective ensemble forecasting with an application to power transformers”. Applied Soft Computing, 68, 233-248.
[28] Peimankar, A., Weddell, S.J., Jalal, T. and Lapthorn, A.C. (2017). "Evolutionary multi-objective fault diagnosis of power transformers". Swarm and Evolutionary Computation, 36, 62-75.
[29] V.K. Ojha, Abraham A, and Snášel V. (2017). "Ensemble of heterogeneous flexible neural trees using multiobjective genetic programming". Applied Soft Computing. 52, 909-24.
[30] U. Bhowan, M. Johnston, M. Zhang, and X. Yao (2013). "Evolving diverse ensembles using genetic programming for classification with unbalanced data". IEEE Transactions on Evolutionary Computation, 17(3), 368-386.
[31] M.-J. Kim and D.-K. Kang (2012). "Classifiers selection in ensembles using genetic algorithms for bankruptcy prediction". Expert Systems with Applications, 39(10), 9308-9314.

[32] H. Bhasin and S. Bhatia (2011). "Application of genetic algorithms in machine learning". International Journal of Computing Science and Information Technology, 2(5), 2412-2415.
[33] V. H. A. Ribeiro and G. Reynoso-Meza (2018). "A multi-objective optimization design framework for ensemble generation". Proceedings of the Genetic and Evolutionary Computation Conference Companion, 1882-1885.
[34] C. Zhang, P. Lim, A. Qin, and K. C. Tan (2017). "Multiobjective deep belief networks ensemble for remaining useful life estimation in prognostics". IEEE Transactions on Neural Networks and Learning systems, 28(10), 2306-2318.
[35] A. Peimankar, S. J. Weddell, T. Jalal, and A. C. Lapthorn (2016). "Ensemble classifier selection using multi-objective PSO for fault diagnosis of power transformers". IEEE Congress on Evolutionary Computation, 3622-3629.
[36] J. Zhao, L. Jiao, S. Xia, V. B. Fernandes, I. Yevseyeva, Y. Zhou, and M. T. Emmerich (2018). "Multiobjective sparse ensemble learning by means of evolutionary algorithms". Decision Support Systems, 111, 86-100.
[37] A. Chandra and X. Yao (2006). "Ensemble learning using multi-objective evolutionary algorithms". Journal of Mathematical Modelling and Algorithms, 5(4), 417-445.
[38] F. Han, D. Yang, Q.-H. Ling, and D.-S. Huang (2015). "A novel diversity-guided ensemble of neural network based on attractive and repulsive particle swarm optimization". International Joint Conference on Neural Networks, 1-7.
[39] J. Wu, J. Long, and M. Liu (2015). "Evolving RBF neural networks for rainfall prediction using hybrid particle swarm optimization and genetic algorithm". Neurocomputing,148, 136-142.
[40] B. Xue, M. Zhang, and W. N. Browne (2013). "Particle swarm optimization for feature selection in classification: A multi-objective approach". IEEE Transactions on Cybernetics, 43(6), 1656-1671.
[41] Z. Md. Jan, and B. Verma (2019). "Ensemble classifier Optimization by reducing input features and base classifiers". International Congress on Evolutionary Computation, 1276-1282.
[42] A. H.R. Ko, R. Sabourin, and A. S. B. Jr (2008). "From dynamic classifier selection to dynamic ensemble selection". Pattern recognition, 41(5), 1718-1731.
[43] R. M.O. Cruz, R. Sabourin, G. D.C. Cavalcanti, and T.I. Ren (2015). "META-DES: A dynamic ensemble selection framework using meta-learning". Pattern recognition, 48(5), 1925-1935.
[44] H. J. Escalante, M. Montes, and E. Sucar (2010). "Ensemble particle swarm model selection". International Joint Conference on Neural Networks, 1-8.
[45] H. J. Escalante, M. Montes, and L. E. Sucar (2009). "Particle swarm model selection". Journal of Machine Learning Research, 10, 405-440.
[46] I. Barandiaran (1998). "The random subspace method for constructing decision forests". IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(8).
[47] R. Bryll, R. Gutierrez-Osuna, and F. Quek (2003). "Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets". Pattern Recognition, 36(6), 1291-1302.
[48] A. Lazarevic and V. Kumar (2005). "Feature bagging for outlier detection". Proceedings of the 11th International Conference on Knowledge Discovery in Data Mining, 157-166.
[49] A. Di Marco and R. Navigli (2013). "Clustering and diversifying web search results with graph-based word sense induction". Computational Linguistics, 39(3), 709-754.
[50] J. Kennedy (2011). "Particle swarm optimization". Encyclopedia of machine learning,760-766.
[51] S. Mirjalili and A. Lewis (2013). "S-shaped versus V-shaped transfer functions for binary particle swarm optimization". Swarm and Evolutionary Computation, 9, 1-14.
[52] K. Bache and M. Lichman (2013), "UCI machine learning repository", http://archive.ics.uci.edu/ml/
[53] L. I. Kuncheva and C. J. Whitaker (2003). "Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy". Machine Learning, 51(2), 181-207.
[54] A. Jurek, Y. Bi, S. Wu, and C. D. Nugent (2014). "Clustering-based ensembles as an alternative to stacking," IEEE Transactions on Knowledge and Data Engineering, 26(9), 2120-2137.
[55] Z. Yu, Y. Lu, J. Zhang, J. You, H.S. Wong, Y. Wang and G. Han (2017). “Progressive semisupervised learning of multiple classifiers”. IEEE Transactions on Cybernetics, 48(2), 689-702.
[56] F. Wilcoxon (1945). "Individual comparisons by ranking methods". Biometrics bulletin, 1(6), 80-83.