# A Novel Method for Creating an Optimized Ensemble Classifier by Introducing Cluster Size Reduction and Diversity

Zohaib Jan, Juan Munoz and Asim Ali[1]

***Abstract*—In this paper, a new method is proposed for creating an optimized ensemble classifier. The proposed method mitigates the issue of class imbalances by partitioning the input data into its various data classes. The partitions are then clustered incrementally to generate a pool of class pure data clusters. The generated data clusters are then balanced by adding samples from all classes which are closest to the cluster centroid. In this manner all generated data clusters are balanced and classifiers trained on such a data cluster are unbiased as well. This creates a diverse input space for training of base classifiers. The pool of clusters is then utilized to train a set of diverse base classifiers to generate the base classifier pool. The pool of classifiers is then treated as a combinatorial problem of optimization and an evolutionary algorithm is incorporated. The proposed approach generates an optimized ensemble classifier that can not only achieve the highest classification accuracy but also has a lower component size as well. The proposed approach is tested on 31 benchmark datasets from UCI machine learning repository and results are compared with existing state-of-the-art ensemble classifiers as well.**

***Index Terms*—Ensemble classifiers, Fusion of classifiers, Clustering, Neural networks.**

## I. Introduction

ENSEMBLE classifier is a notably recent machine learning classification technique for improving the classification accuracy by suitably combining the class label estimates of an individual classifier. 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 defined by Zhou, Dietterich in [1-3]. Ensemble classifiers benefit from the "perturb and combine" strategy and over the years many new methods of ensemble classifiers have been developed. A number of research has pointed to the benefits of diversity in ensemble classifiers however a trade-off between accuracy and diversity must be maintained as indicated by Kuncheva and Whitaker in [4].

Precedence given to diversity only will result in an ensemble classifier that is diverse in nature but performs inaccurately; the main objective is to develop an ensemble classifier that can achieve higher classification accuracy over unseen data. To achieve diversity different measures have been proposed over the years with the prime objective to increase overall ensemble classification accuracy [4]. Generally, diversity in an ensemble classifier can be achieved by 1) sub sampling of data, 2) feature randomization of data and 3) parameter randomizations of classifiers. In regards to achieving diversity through sub sampling two pioneering works to consider are bagging and boosting shown by Breiman, Freund and Schapire in [5, 6]. Bagging works by creating sub samples of data with repeating and unique groups; classifiers are trained on each sub sample which are then suitably combined. Boosting on the other hand subsequently trains a classifier on data patterns where the classifier performed poorly, therefore the name boosting. A popular ensemble classifier methodology based on boosting is AdaBoost. Over the years many variations of AdaBoost have been proposed and they are detailed Vezhnevets, Domingo and Avidan in [7-9]. A renowned work in achieving diversity through feature randomization is random forest stated by Breiman in [10]. Random forest works by training decision trees on random subset of records and features from the training dataset. Ensemble classifier methodologies based on parameter randomization can be classified further into two categories. Firstly, are ensemble classifier methodologies that randomize classifier parameters using kernel functions specified by Gönen and Alpaydın in [11], and secondly are methodologies that use evolutionary algorithms to manipulate features and/or ensemble classifier components suggested by Rahman and Verma [12].

Another key aspect to consider when generating an ensemble classifier is the ensemble size itself. As stated by Wolpert and Macready in [13] no single model can perform well on every dataset and to obtain a better class label estimate multiple classifiers should be suitably combined. Adding classifiers in an ensemble classifier does contribute to classification accuracy, however, according to [14-16] adding more than an optimal number of classifiers only contributes to ensemble size and complexity of the ensemble.

Zohaib Jan and Juan Carloz Munos are with the School of Engineering and Technology at Central Queensland University, 160 Ann Street, Brisbane, Australia, e-mail: {z.jan, j.munoz}@cqu.edu.au. Asim Ali is with Hamdard University, Karachi Pakistan email: {asim82ali@gmail.com}.

This is known as "*the law of diminishing returns*" defined by Shephard and Färe in [17].

The main aim of this research is to address the problem in optimizing ensemble classifier in particular finding a best set of diverse base classifiers to create an optimized ensemble classifier. The original contributions presented in this paper are as follows:

- a novel method that can generate, select and fuse a best set of diverse classifiers to produce an optimal ensemble classifier is presented.
- a new approach to obtain diverse base classifiers for ensemble via class-based clustering.
- a new approach of minimizing class imbalances through the incorporation of clustering balancing based on Euclidean space.

The rest of the paper is organized as follows. Section II entails the current state of the art ensemble classifier techniques. Section III discusses the proposed method for creating an optimized ensemble classifier. Section IV gives details about the experimental setup, experiments, results, and analysis of results. Section V summarizes our findings and lays out future directions.

## II. Related work

Ensemble classifiers can outperform individual classifiers, however there are many problems with ensemble classifiers such as finding appropriate diverse base classifiers and their fusion. Many individual classifiers such as neural networks, decision trees, *etc.* trained by searching in the local search space suffer from converging to a local optimum. Ensemble classifiers overcome this by having various learners search in the problem space in parallel and find a better estimation by converging to a global optimum. It has been pointed out by various researchers that in order to traverse the search space from multiple angles one key factor is to have a diverse set of classifiers [2, 18-20]. Diversity in classifiers can be achieved in two ways: 1) training classifiers on a diverse input sub space generated from the dataset, and 2) training of structurally different classifiers on dataset. This way trained classifiers will not produce correlated errors and the ensemble classifier will perform well on unseen data. The bulk of this section will discuss recent ensemble classifier methods that incorporate diversity by generating a diverse input space, train classifiers on a subset of features, and utilize evolutionary algorithms to optimize diverse sets of classifiers.

A significant amount of research has been conducted in creating different methods to promote diversity in an ensemble classifier [21-23]. Random Forest (RaF) is a very robust and versatile ensemble classifier method and many variations have been proposed over the years. In a study [24] authors exhaustively compared 179 classifiers on entire UCI classification dataset repository and concluded that the best ensemble classifier overall is a variation of RaF known as parallel RaF. Similarly in [25] authors conducted a benchmark study of different ensemble classifiers on 121 datasets from UCI repository and concluded that oblique decision tree ensemble out performed other ensemble classifier methods. The oblique decision tree ensemble was originally proposed in [26] and is a variation of RaF known as MPRaF.

Clustering has received much focus when it comes to generating ensembles [12, 20, 27-31]. Some authors used clustering algorithms to form clusters of classifiers whereas others have used clustering to partition training data into distinct data clusters [32]. As such in [33] a cluster oriented ensemble classifier was proposed. Dataset was first partitioned into distinct data clusters and on each data cluster a set of base classifiers were trained. Trained classifiers were combined using an artificial neural network to generate the ensemble classifier. It was argued that using a fusion classifier rather than a typical algebraic method to fuse decisions from different base classifier is a better way to generate an ensemble. Similarly in [21] a methodology of incremental ensemble learning processes was proposed. The input data was partitioned into a number of data clusters incrementally and a set of base classifiers were trained on all generated data clusters. Classifier accuracy and diversity was calculated for all trained classifiers and they were added to the pool on the basis of *accuracy precedence diversity*. Meaning, a classifier will be added to the pool if it achieved higher classification accuracy than the one already in the pool; if accuracy is the same then diversity was compared. The classifier was discarded if neither the accuracy nor the diversity was increased. In another research [34] diversity was achieved by clustering data into distinct clusters and discarding redundant clusters whose Jaccard index was higher than a given threshold. It was also suggested that using a maximum value of $K$ for generating clusters incrementally should be $K = \sqrt[3]{n}$, where $n$ is the number of records in the training dataset. This not only ensured that the computational complexity of clustering remains $O(n^2)$, but also kept the algorithm from creating clusters with few records in them.

In [29] authors achieved diversity by clustering data into atomic and non-atomic clusters. An atomic cluster is class pure whereas a non-atomic cluster has multiple class labels in it. Every non-atomic cluster is fed into a neural network classifier to transform it into an atomic cluster; the process repeats till every non-atomic cluster is converted into an atomic cluster. When all clusters are class pure decisions can be formed. In [30] a hybrid sample based clustering ensemble was proposed which is based on an extension to boosting and bagging. It was suggested to partition input data iteratively using a hybrid sampling procedure, inspired by the nature of boosting and bagging. On all generated partitions a novel consensus function is applied, this encodes the local and global cluster structure into a single representation which is then consolidated into a single partition by a clustering algorithm.

A significant amount of research has been conducted in utilizing evolutionary algorithms such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO), *etc.*

to optimize different ensemble classifier hyper-parameters, ensemble classifier selection, and or optimizing input space [35, 36]. As such in [12] it was suggested that achieving high diversity with accuracy can be classified as a multi objective optimization problem and GA can be utilized to find a balance between accuracy and diversity. Dataset is partitioned into distinct clusters incrementally iteratively and in each iteration a different value of *K* clusters was generated. Authors employed GA to search for the optimum value of *K* which was able to achieve the highest classification accuracy. In [36] authors employed multi objective optimization to search for the best hyper parameters that can generate an optimal pareto front and as an instance selection tool to select the best of classifiers from the pool. In [37] a multi objective genetic programming framework for classification of data with unbalanced majority and minority classes as a learning objective. In [38], authors used a divide and conquer based hierarchical method to first create a diverse input space and train a set of heterogeneous classifiers. The trained classifiers are then suitably combined by means of a multi objective optimization algorithm. The proposed method was not only able to perform well but also generated an ensemble classifier with an optimal size. Similarly, according to Wu and Bongard in [39] a set of heterogeneous classifiers are combined through active learning. The proposed method optimizes ensemble size iteratively to find the best size and highest classification accuracy.

In [40] authors proposed a method for ensemble classifiers using clustering and PSO. Clusters of classifiers are generated and are assigned weights which are calculated using PSO. In PSO, each cluster was treated as a particle in *k* dimensions. Relative weights are given to clusters using classical PSO. This proved not only efficiency in terms of generalization error but also effective in lowering the complexity of the ensemble. Similarly, in [41] PSO was used as a model selection tool to select the best set of classifiers for ensemble classifiers. Authors argued that traditional model selection methodologies focus on maximising individual models' accuracy rather than promoting global accuracy and diversity of the ensemble. A popular model selection approach which overcomes this problem is known as Particle Swarm Model Selection (PSMS). In recent studies [41-44] PSMS has shown great success and proved to be a good contender for optimizing a binary search space. PSMS was used to find the best set of features, as a model selection tool, and to optimize parameters for classification dataset.

In another research [22] Attractive and Repulsive PSO (ARPSO) was proposed which selects the best set of classifiers from an initial pool of classifiers. This is done by considering both classification accuracy and diversity. The proposed approach improved generalization performance by considering diversity of classifiers and adaptively selects the number of classifiers. As discussed, PSO has proven to be a good optimization tool for ensemble learning methodologies. PSO is a metaheuristic optimization algorithm originally proposed by Kennedy in 1995 and Eberhart [27]. PSO helps in optimizing continuous and unconstrained nonlinear optimization problems. PSO mimics the social behaviour of a flock of birds. It is a population-based algorithm in which each population member becomes a possible solution. The population consists of a set of particles and each particle has a personal best and a global best. PSO searches in the nonlinear search space to find a solution that minimizes the global error whilst also minimizing the individual particle's personal error as well.

According to Feurer et al. [50] there is evidence that no single method of AutoML outperform on all datasets under experimentation and secondly some methods in machine learning essentially depend on hyperparameter optimization. However, the successful use of AutoML frameworks which effectively uses Bayesian optimization has corrected the latter problem. The former problem is interconnected with the latter problem since the ranking of calculations rely upon whether their hyperparameters are fine-tuned. These two problems can be tackled as separated, organized joint optimization problem. The use of Bayesian optimizer for meta learning corresponds to a probabilistic model that compiles, assess, and connects hyperparameter settings and its respectively model result that derives from this iteration.

Meta-learning approach work hand to hand to Bayesian optimization when performing a ML framework and suggest some instantiations for the framework. However, it is unable to provide fine-grained information on performance. The positive side of Bayesian optimization is that can fine-tune performance successfully overtime by selecting k configurations based on meta learning. While these methodologies are promising, they can be restricted to a few meta-features and unable to adapt to the highdimensional configurations faced in AutoML.

While Bayesian hyperparameter optimization is efficient when processing data or when finding the best-performing hyperparameter setting. However, it is an inefficient strategy when the objective is essentially to make great forecasts as models trained during the span of the search are lost. Feurer et al. proposes a post-processing method to store and use the models as an automatic ensemble construction using multiple hyperparameter setting making it more robust and reliable compared to individual models.

As discussed in the literature a number of methods have been proposed to promote diversity in ensemble classifiers, and optimize ensemble classifier components, however, a careful consideration should be given to create an optimal ensemble size by selecting not only diverse but accurate classifiers as well. The challenges involved in creating a diverse set of classifier involves i) identifying the best set of heterogeneous classifiers which can achieve highest classification accuracy, ii) identifying the best set of data clusters that can contribute to data diversity which in turn will result in achieving higher classification accuracy, and iii) identifying the means of using optimization algorithm to optimize the pool of classifiers to generate a diverse and accurate ensemble classifier. This paper contributes to the above-mentioned areas by proposing an ensemble classifier

generation method by i) introducing data diversity through clustering and balancing, and ii) optimizing the pool of trained classifiers to achieve higher classification accuracy and optimum ensemble size.

## III. Proposed Method

### A. Preliminaries

The focus of this paper is to develop a novel method of generating an optimized ensemble classifier by selecting the best set of classifiers through incorporating an evolutionary algorithm. This is achieved by suitably combining a subset of base classifiers from the pool that can achieve the highest classification accuracy. The proposed method starts off by creating a diverse training space that minimizes the class imbalances and classifier biasness by clustering input data incrementally. The benefit of creating a diverse search space is in two folds 1) it allows the proposed ensemble classifier approach to train multiple classifiers on different subsets of data samples as in bagging therefore allowing smaller datasets especially to have a sufficient base classifier pool, and 2) since classifiers are trained on dense data clusters therefore, they have local expertise and are able to bring with them different learning capabilities. All trained classifiers are then passed to an optimization algorithm which selects a subset of diverse classifiers and fuses them to generate an optimized ensemble classifier.

### B. Diverse training space generation

The proposed method generates a diverse search space by first partitioning the input data into its various data classes. Let us assume that $X = \{(x_1, y_1), (x_2, y_2) \ldots, (x_n, y_n)\}$ is the training dataset and $n$ is the number of records, then $x$ is a feature vector and $y_n$ is the respective class label. Data partitions are generated which are class pure as for example subsets $X^1, X^2, .., X^v \subseteq X$ such that $X^v$ will contain data samples from only class $v$. If for example in the case of *breast cancer* dataset which has only 2 data classes malignant and benign, only 2 subsets are generated. Each data partition is clustered using k-means algorithm with the value of $k$ starting from 2 going up to a maximum of $K$. Clustering is achieved by grouping similar data samples into different groups (clusters) with a common mean. This is achieved by minimizing the squared Euclidean distance from the centroid of a data cluster given as:

$$\text{argmin}\left(\sum_{j=1}^{k}\sum_{i=1}^{n}\left|x_i - c_j\right|^2\right) \quad (1)$$

where $x$ is a feature vector, $k$ is the number of 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. 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 maximize the classification accuracy of the ensemble.

Since the generated data clusters are class pure therefore, they must be balanced for a classifier that is trained on them to be unbiased as well. 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 true for data clusters $C^1$ and 2: $C^1 \cap C^2 = \emptyset$ and $C^1 \cup C^2 = X^i$. 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 $v$, let us say $v'$, then data samples from classes besides $v'$ 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:

$$dist = norm\big(d(x_i, c)\big) \;\; \forall \; i \in V \; and \; i \notin v' \quad (2)$$

where $d(x_i, c) = \sum |x_i - c|$. 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 $v'$ 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.

### C. Train base classifiers

A set of diverse base classifiers which include Naïve Bayes (NB), Discriminant Analysis (DISCR), k-Nearest Neighbour (kNN), Decision Trees (DT), Artificial Neural Networks (ANN), and Support Vector Machines (SVM) are trained on all balanced data clusters. These classifiers are structurally different with different learning capabilities and contribute to classifier diversity. Additionally, classifier parameters are randomized to further increase diversity. The hyper parameters and their respective selection criteria are given in Table II.

### D. Optimizing the classifier pool

The search space $\varphi$ for an optimization problem for ensemble classifier solution $\varepsilon$ is defined as follows:

$$argmin\big(f_1(\varepsilon), f_2(\varepsilon), \ldots, f_S(\varepsilon)\big) \coloneqq \{\varepsilon \mid \varepsilon \in \varphi\} \quad (3)$$

where $S$ is the number of possible ensemble classifier solutions. The $i^{th}$ solution $\varepsilon^i(\chi) = \{\zeta^1, \zeta^{10}, \zeta^{bcp}\}$ is formed by combing 1st, 10th, and $bcp^{th}$ classifiers from the pool of generated trained classifiers $\zeta$. The objective of the optimization algorithm is formulated by adding the ensemble classifier root mean square error(RMSE) over validation dataset and the ensemble component size. The RMSE of the ensemble is calculated using the class label estimates of the ensemble generated over the feature vector $x \in V$ of the validation dataset. The class estimates of $i^{th}$ solution is obtained as follows:

$$\mathrm{y}_i = mode(\varepsilon^i(\chi)) \quad (4)$$

where *mode* is a mathematical operator which depicts majority voting and is used to fuse decisions of different classifiers. Finally, the RMSE is calculated as follows:

$$RMSE_i = \frac{\sqrt{\sum_{j=1}^{|V|}(y_j - y'_j)^2}}{|V|} \quad \forall y \in V \qquad (5)$$

where $y'$ is the predicted class labels obtained from (4).

**Algorithm 1: Optimized Ensemble Classifier**

**Input:** Training data, $X = \{x_i, y_i\}_{i=1}^{n}$
$n$ - Data size
$K$ - Number of data clusters/class
**Output**: Optimized ensemble classifier

1. Generate $X^n$ subsets of the input data such that $X^1, X^2, \ldots X^\kappa \subseteq X$
2. **foreach** $X^n$ do
3. $sp^i \leftarrow$ partition $X^n$into $K$ data clusters by minimizing the squared Euclidean distance of each sample from the cluster centroid
4. **endfor**
5. **foreach** class-pure cluster $C$ in $sp$ do
6. determine the dominant class $y_d$in cluster
7. $y_{nd} \leftarrow$ determine nondominant classes in the cluster
8. foreach $y_{nd}$ do:
9. add feature vectors $x$ from remaining clusters in the pool closest to cluster centroid
10. **endfor**
11. **endfor**
12. **foreach** data clusters $C$ in pool $sp$ do
13. $bcp^i \leftarrow$ train a base classifier $\zeta$ on the data cluster $C$ and add to the pool
14. **endfor**
15. Initialise a population of $|bcp|$ particles
16. **while** termination criteria
17. Map each particle to a trained base classifier in the population by generating a binary bit string pop representing each classifier in the population
18. Generate an ensemble solution $\xi$ of classifiers that have a higher value than the threshold $\theta$
19. Calculate the fitness of the population using equation (6) with validation dataset
20. Update the local best and global best of the population
21. Update particle velocity and position
22. **endwhile**
23. Use the optimized pool $bcp'$ of classifiers predict the class labels of the unseen dataset, the test set and calculate ensemble classification accuracy.

The objective function of the optimization algorithm is given as follows:

$$f(\xi) = RMSE_1 + |\varepsilon^i|_2 \qquad (6)$$

Which essentially is the sum of the root mean square error and the component size of ensemble solution. Therefore, the solution that can achieve the lowest error and has the smallest component size is selected. Particle Swarm Optimization is used here as a black box optimization toolbox. Population in PSO is represented is represented as follows as:

$$pop = [\varphi^1, \varphi^2, \ldots., \varphi^{|bcp|}] \qquad (7)$$

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

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 that utilized clustering and optimization to generate an ensemble is given in Algorithm 1.

### *E. Theoretical analysis of the proposed approach*

The time complexity of the proposed approach can be computed as a sum of four tasks i)complexity of generating data clusters, ii) complexity of balancing data clusters, iii) generate base classifier pool and iv) optimize the pool of classifiers.

- $T = T_{clustering} + T_{b\_clusters} + T_{bcp} + T_{O\_bcp}$
- where $T_{clustering}$, $T_{b\_clusters}$, $T_{bcp}$and $T_{O\_bcp}$ is the time complexity of generating data clusters, balancing data clusters, generating base classifier pool, and optimizing base classifier pool.
- The time 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.
- The number of generated data clusters is a factor of $K$ and the number of data classes in the input data. Therefore, to balance the data clusters if there are $m$ samples in a data cluster and $1 - v$ classes exist therefore the complexity of balancing data clusters is $O(m * v)$.
- On each balanced data cluster a set of base classifier is trained. Assuming the worst-case complexity of training a classifier is $O(n^2)$ therefore if there are $sp$ data clusters in the pool the overall complexity of generating the base classifier pool will be $O(sp \times n^2)$.
- Lastly, the cost of optimization is $O((sp \times n^2)^2)$
- Therefore, the complexity of the proposed approach is the sum of all given as
  - $O(ikn + m * v + n^2 + s^2 \times n^4)$ the number of sapmles $n$ is the largest coefficient here, therefore the worst-case complexity is $O(n^5)$.

## IV. Experiments And Analysis

In this section, we present the benchmark datasets, experiments, results and comparative analysis to evaluate the proposed method.

### *A. Datasets*

Benchmark datasets from UCI Machine Learning Repository

[45] were used for experimentation. The details of these datasets are given in Table I. We have used UCI benchmark datasets so that the results can be compared with other researchers as many researchers use UCI benchmark datasets. A mix of small and large datasets with few and many classes are chosen for experiments. It can be noted from Table I that 10 datasets have more than 10 classes, and 3 datasets have more than 1000 records with *Adult* having 48000 records. There is a good mix of datasets to test the performance of the proposed method thoroughly.

TABLE I: DETAILS OF UCI DATASETS USED IN EXPERIMENTATION

| Datasets | # of features | # of records | # of classes |
|---|---|---|---|
| *Adult* | 14 | 48842 | 2 |
| *Australian* | 14 | 690 | 2 |
| *Balance* | 4 | 625 | 3 |
| *Banknote* | 4 | 748 | 2 |
| *Breast Cancer* | 9 | 683 | 2 |
| *Bupa* | 6 | 345 | 2 |
| *DNA* | 61 | 3190 | 3 |
| *E.coli* | 7 | 336 | 2 |
| *Fertility* | 9 | 100 | 2 |
| *Haberman* | 3 | 306 | 2 |
| *Hayes Roth* | 5 | 160 | 3 |
| *Heart* | 13 | 270 | 2 |
| *Hepatitis* | 19 | 80 | 2 |
| *Ionosphere* | 33 | 351 | 2 |
| *Iris* | 4 | 150 | 3 |
| *Letter Recognition* | 16 | 2000 | 26 |
| *Liver* | 6 | 345 | 2 |
| *Page Blocks* | 10 | 5473 | 5 |
| *Pima Diabetic* | 8 | 768 | 2 |
| *Segment* | 19 | 2310 | 7 |
| *Sonar* | 60 | 208 | 2 |
| *Stat-Image* | 13 | 270 | 2 |
| *Teaching* | 5 | 151 | 3 |
| *Thyroid* | 5 | 215 | 3 |
| *Transfusion* | 4 | 748 | 2 |
| *Vehicle* | 18 | 946 | 4 |
| *Vowel* | 13 | 528 | 11 |
| *WDBC* | 30 | 569 | 2 |
| *Wine* | 13 | 178 | 3 |
| *Zoo* | 17 | 101 | 7 |

### B. Experimental setup

The proposed methodology is implemented in MATLAB [46], a 10-fold cross validation is conducted to accommodate for randomness as in other similar works. A set of base classifiers (ANN, SVM, DT, DISCR, KNN, NB *etc.*) are used to train on all generated data clusters. Mostly default parameters are used besides the ones mentioned in Table II, which are randomized to accommodate for further classifier diversity.

TABLE II: PARAMETERS USED IN EXPERIMENTS

| Algorithm / Classifier | Parameter | Values |
|---|---|---|
| *Neural network* | Hidden neuron | Random between 10-30 |
| | Training function | Levenberg-Marquardt backpropagation / Bayesian regularization 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 neighbor* | Number of neighbors | 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 optimization* | Maximum iteration | 100 |
| | Stall iteration | 10 |
| | Swarm size | 100 |

Particle Swarm Optimization (PSO) from the global optimization toolbox in MATLAB (function = "*particleswarm"*)is used as a block box optimization tool. In order to binarize the operation of classifier selection from the pool the value of $\theta$ is set to 0.5 as in other similar works [52]. The number of particles is set to 100, with a maximum stall iteration time set to 10 to accommodate for dead locks. The number of variables is set to the number of classifiers in the pool. The classifiers are represented as particles in the search space as a binary row vector with index value (positions) limited to 0 and 1. 1 means a classifier is selected in the search space and 0 means otherwise. The proposed methodology is uploaded on GitHub for reuse purposes [53]. For various classifier implementation the following functions from MATLAB were used:

- *fitcecoc* = multi class Support Vector Machine
- *fitcnb* = Naïve Bayes
- *fitcdiscr*= Discriminant Analysis
- *fitctree* = Decision Tree
- *train* = Neural Network
- *fitcknn* = K-Nearest Neighbor

Instead of using a single value for the upper bounds of clustering $K$, a range of values are tested [2, 10]. The highest average classification accuracy over 10 folds for a given input value of $K$ for a given dataset is reported.

### C. Results

The highest average classification accuracy that can be achieved for a given value of $K$ for a given dataset with and without the incorporation of optimization is reported in Table III, along with standard deviation and the value of $K$. It can be noted that for each dataset a different value of the upper bounds of clustering $K$ achieved the highest classification accuracy. Moreover, in all cases the optimized ensemble classifier outperformed the non-optimized ensemble classifier further adding to the fact that not all classifiers in the pool are suitable to be included as a part of ensemble and thus proving that more is not necessarily better. Different values of $K$ adds to the fact that each dataset has different intrinsic spatial characteristics and therefore, when exploited properly can results in creating a diverse input space for training. Some datasets are sparse, and others are dense in nature which in turn results in the requirement of different values of $K$.

Through the incorporation of optimization, not only the ensemble classifier which can achieve the highest classification accuracy is selected but also the ensemble with the lowest component size is selected. This certainly adds to the fact that "less is more" [54], and can be seen from figure 1 that of all the trained classifiers in the pool on average 50% were selected to generate the final optimized ensemble classifier.

TABLE III: HIGHEST AVERAGE CLASSIFICATION ACCURACY OF THE PROPOSED ENSEMBLE CLASSIFIER APPROACH WITH AND WITHOUT AND INCORPORATION OF OPTIMIZATION

| **Dataset** | **Classification accuracy without optimization** | **Std. dev** | **Classification accuracy with optimization** | **Std. dev** | ***K*** |
|---|---|---|---|---|---|
| *Adult* | 0.6175 | 0.099 | 0.8426 | 0.002 | 3 |
| *Australian* | 0.8449 | 0.044 | 0.8652 | 0.042 | 10 |
| *Balance* | 0.5439 | 0.056 | 0.8878 | 0.011 | 6 |
| *Banknote* | 0.9920 | 0.011 | 0.9998 | 0.004 | 2 |
| *Breast Cancer* | 0.9685 | 0.020 | 0.9742 | 0.017 | 6 |
| *Bupa* | 0.6777 | 0.060 | 0.7184 | 0.079 | 9 |
| *DNA* | 0.5931 | 0.053 | 0.8178 | 0.023 | 4 |
| *E.coli* | 0.7529 | 0.066 | 0.8977 | 0.067 | 9 |
| *Fertility* | 0.5200 | 0.270 | 0.8300 | 0.133 | 8 |
| *Haberman* | 0.4731 | 0.132 | 0.7139 | 0.033 | 4 |
| *Hayes Roth* | 0.7500 | 0.150 | 0.7625 | 0.149 | 7 |
| *Heart* | 0.8481 | 0.079 | 0.8396 | 0.070 | 3 |
| *Hepatitis* | 0.8375 | 0.060 | 0.8750 | 0.102 | 4 |
| *Ionosphere* | 0.8832 | 0.043 | 0.8944 | 0.015 | 6 |
| *Iris* | 0.9000 | 0.056 | 0.9483 | 0.063 | 6 |
| *Letter Recognition* | 0.6475 | 0.013 | 0.8092 | 0.003 | 2 |
| *Liver* | 0.6731 | 0.093 | 0.7185 | 0.099 | 9 |
| *Page Blocks* | 0.0941 | 0.008 | 0.9564 | 0.002 | 3 |
| *Pima Diabetic* | 0.6588 | 0.049 | 0.7525 | 0.042 | 8 |
| *Segment* | 0.9329 | 0.003 | 0.9796 | 0.005 | 2 |
| *Sonar* | 0.5914 | 0.128 | 0.8012 | 0.019 | 10 |
| *Stat-Image* | 0.7355 | 0.021 | 0.8871 | 0.004 | 9 |
| *Teaching* | 0.4763 | 0.117 | 0.5767 | 0.017 | 7 |
| *Thyroid* | 0.1843 | 0.046 | 0.9988 | 0.002 | 10 |
| *Transfusion* | 0.2606 | 0.077 | 0.7236 | 0.014 | 7 |
| *Vehicle* | 0.6254 | 0.039 | 0.7776 | 0.008 | 2 |
| *Vowel* | 0.5404 | 0.037 | 0.8837 | 0.008 | 6 |
| *WDBC* | 0.9157 | 0.025 | 0.9671 | 0.004 | 8 |
| *Wine* | 0.9438 | 0.053 | 0.9991 | 0.014 | 7 |
| *Zoo* | 0.9700 | 0.067 | 0.9700 | 0.067 | 4 |

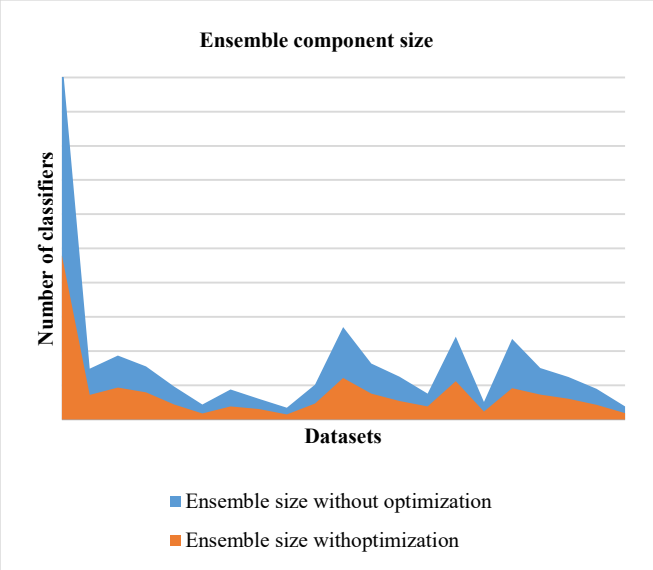


Figure 1: Effect on ensemble component size with the incorporation of optimization

### *D. Comparative analysis*

The classification accuracy of the proposed ensemble classifier is compared with two existing state-of-the-art ensemble classifiers namely boosting, and random forest. For fair comparisons these ensembles were implemented in the same environment using the *fitcensemble* function for boosting in MATLAB with "adaboostM1" for binary classification problems and "adaboostM2" for multi class classification problems. As for random forest the function *treebagger* in MATLAB was used with 50 trees given as parameter. The average classification accuracies over 10-folds are reported in the table and used for analysis.

It can be noted from Table IV, that the proposed ensemble classifier outperformed both boosting, and random forest in 17 out of 30 datasets. In 3 datasets boosting outperformed both random forest and the proposed approach and in 9 datasets random forest outperformed both boosting and the proposed ensemble classifier. The highest classification accuracies are given in bold in Table IV.

TABLE IV: COMPARATIVE ANALYSIS OF THE PROPOSED ENSEMBLE APPROACH WITH BOOSTING AND RANDOM FOREST. THE HIGHEST CLASSIFICATION ACCURACIES ARE GIVEN IN BOLD

| Dataset | Proposed approach | Boosting | RaF |
|---|---|---|---|
| *Adult* | 0.8426±0.002 | **0.8644±0.001** | 0.8616±0.001 |
| *Australian* | **0.8652±0.042** | 0.8565±0.007 | 0.8646±0.004 |
| *Balance* | **0.8878±0.011** | 0.8799±0.008 | 0.8410±0.004 |
| *Banknote* | **0.9998±0.004** | 0.9976±0.001 | 0.9926±0.001 |
| *Breast Cancer* | **0.9742±0.017** | 0.9642±0.003 | 0.9669±0.002 |
| *Bupa* | **0.7184±0.079** | 0.6917±0.026 | 0.7164±0.015 |
| *DNA* | 0.8178±0.023 | 0.8040±0.001 | **0.8760±0.002** |
| *E.coli* | **0.8977±0.067** | 0.8368±0.018 | 0.8572±0.011 |
| *Fertility* | 0.8300±0.133 | 0.8320±0.022 | **0.8660±0.018** |
| *Haberman* | **0.7139±0.033** | 0.6540±0.009 | 0.6774±0.014 |
| *Hayes Roth* | 0.7625±0.149 | 0.8012±0.010 | **0.8087±0.008** |
| *Heart* | **0.8396±0.070** | 0.7851±0.015 | 0.8266±0.009 |
| *Hepatitis* | **0.8750±0.102** | 0.8375±0.000 | 0.8650±0.029 |
| *Ionosphere* | 0.8944±0.015 | **0.9327±0.004** | 0.9322±0.002 |
| *Iris* | **0.9483±0.063** | 0.3333±0.000 | 0.9473±0.009 |
| *Letter Recognition* | 0.8092±0.003 | 0.6527±0.001 | **0.9635±0.001** |
| *Liver* | **0.7185±0.099** | 0.6928±0.010 | 0.7181±0.016 |
| *Page Blocks* | 0.9564±0.002 | 0.9699±0.001 | **0.9739±0.001** |
| *Pima Diabetic* | **0.7525±0.042** | 0.7374±0.005 | 0.7516±0.005 |
| *Segment* | **0.9796±0.005** | 0.9773±0.001 | 0.9691±0.001 |
| *Sonar* | 0.8012±0.019 | 0.8676±0.019 | **0.8245±0.016** |
| *Stat-Image* | 0.8871±0.004 | 0.8742±0.004 | **0.9157±0.001** |
| *Teaching* | 0.5767±0.0175 | 0.6047±0.017 | **0.6582±0.022** |
| *Thyroid* | **0.9988±0.002** | 0.9972±0.001 | 0.9961±0.001 |
| *Transfusion* | 0.7236±0.014 | **0.7679±0.013** | 0.7174±0.006 |
| *Vehicle* | **0.7776±0.008** | 0.7406±0.013 | 0.7486±0.008 |
| *Vowel* | 0.8837±0.008 | 0.7339±0.007 | **0.9647±0.003** |
| *WDBC* | **0.9671±0.004** | 0.9708±0.003 | 0.9613±0.003 |
| *Wine* | **0.9991±0.014** | 0.5720±0.056 | 0.9791±0.002 |
| *Zoo* | **0.9700±0.067** | 0.4054±0.000 | 0.9594±0.008 |

TABLE V: COMPARATIVE ANALYSIS OF THE PROPOSED ENSEMBLE APPROACH WITH A WEIGHTED VOTING FRAMEWORK (WMV) ENSEMBLE

| Dataset | Proposed approach | WMV [48] |
|---|---|---|
| *Balance* | **0.887** | 0.829 |
| *Breast Cancer* | **0.974** | 0.958 |
| *Bupa* | **0.718** | 0.688 |
| *DNA* | 0.818 | **0.932** |
| *E.coli* | **0.898** | 0.822 |
| *Heart* | **0.839** | 0.807 |
| *Hepatitis* | **0.875** | 0.810 |
| *Ionosphere* | 0.894 | **0.915** |
| *Iris* | **0.948** | 0.933 |
| *Letter Recognition* | 0.809 | **0.909** |
| *Page Blocks* | 0.956 | **0.971** |
| *Pima Diabetic* | 0.753 | **0.757** |
| *Segment* | **0.980** | 0.961 |
| *Sonar* | **0.801** | 0.759 |
| *Stat-Image* | 0.887 | **0.895** |
| *Vehicle* | **0.778** | 0.724 |
| *Vowel* | 0.884 | **0.913** |
| *Zoo* | **0.970** | 0.830 |

TABLE VI: COMPARATIVE ANALYSIS OF THE PROPOSED ENSEMBLE APPROACH WITH A RANDOM FOREST BASED ENSEMBLE (MPROF-T) ENSEMBLE

| Dataset | Proposed approach | MPRoF-T [26] |
|---|---|---|
| *Adult* | **0.843** | 0.839 |
| *Australian* | **0.865** | 0.864 |
| *Balance* | 0.887 | **0.893** |
| *Banknote* | 1.000 | 1.000 |
| *Breast Cancer* | **0.974** | 0.963 |
| *DNA* | 0.818 | **0.912** |
| *E.coli* | **0.898** | 0.852 |
| *Fertility* | 0.830 | **0.880** |
| *Haberman* | **0.732** | 0.712 |
| *Heart* | **0.840** | 0.831 |
| *Hepatitis* | **0.875** | 0.847 |
| *Ionosphere* | 0.894 | **0.933** |
| *Iris* | 0.948 | **0.967** |
| *Page Blocks* | 0.956 | **0.972** |
| *Pima Diabetic* | **0.753** | 0.749 |
| *Segment* | **0.980** | 0.956 |
| *Sonar* | 0.801 | **0.835** |
| *Teaching* | **0.577** | 0.547 |
| *Vehicle* | **0.778** | 0.770 |
| *Wine* | **0.999** | 0.979 |

The classification accuracies of the proposed ensemble classifier approach is also compared with already published state-of-the-art ensemble classifiers presented in WMV [26], and MPRoF-T [48]. WMV is a weighted voting framework-based ensemble classifier that combines suitable classifiers from the pool by assigning them weights. MPRoF-T is a rotation forest-based ensemble classifier that uses multi proximal rotation forest with Tikhonov regularization to generate an ensemble classifier.

The classification accuracies are taken directly from their respective papers and the results are given in Table V, and Table VI, with the highest classification accuracies given in bold. It can be noted from Table V that the proposed approach outperformed WMV in 11 out of 18 common datasets, and from Table VI it can be noted that the proposed approach outperformed in 12 out of 20 common datasets against MPRoF-T.

### *E. Significance test*

To test the significance of the results a series of non-parametric signed rank tests with Holme Bonferroni correction [49] were conducted with alpha significance value of 0.05. The *p* values are listed in Table VII below.

TABLE VII: P-VALUES OF WILCOXON SIGNED RANK TEST WITH HOLM BONFERRONI CORRECTION

| Classifiers | p-values |
|---|---|
| *Random Forest* | 0.333 |
| *AdaBoost* | 0.017 |
| *WMV* | 0.080 |
| *MPRoF-T* | 0.490 |

It can be noted from Table VII that the proposed approach performed significantly better than AdaBoost with significance performance gains and the null hypothesis can be rejected at an alpha significance of 0.01. In comparison to WMV the null hypothesis can be rejected at a significance of 0.08, 0.33 in comparison to Random Forest, and 0.49 in comparison to MPRoF-T.

## V. CONCLUSION

In this paper we proposed an ensemble framework that utilizes clustering and an evolutionary algorithm. The proposed ensemble classifier mitigates the problem of class imbalances by partitioning input data into its constituent classes and then by clustering the partitions into various class pure data clusters. The generated data clusters are then balanced by adding samples from all remaining classes that are closest to the centroid. A set of diverse base classifiers is trained on all generated data clusters and a base classifier pool

is generated. The generated pool of base classifier is then optimized to generate the final ensemble that can not only achieve the highest classification accuracy but also has a lower component size.

The proposed technique has been tested on a variety of benchmark classification datasets from the UCI machine learning repository. The highest classification accuracies that can be achieved with a given value of $K$ were reported. This proved to the fact that a single static input value of $K$ for different datasets is not ideal to generate an ensemble classifier and a dynamic way of finding the optimal value would be a better approach. The results of the proposed ensemble were also compared with existing state-of-the-art ensemble classifiers and significance testing is conducted to further validate the efficacy.

Although the results from the proposed technique can improve the accuracy on selected benchmark datasets, it may not perform well in cases where the training and test patterns follow a denser distribution. We will further experiment and analyze the proposed technique for the improvement of such

## REFERENCES

[1] Z.-H. Zhou, Ensemble methods: foundations and algorithms. CRC Press, 2012.
[2] T. G. Dietterich, "Ensemble Methods in Machine Learning," Multiple Classifier Systems, vol. 1857, pp. 1-15, 2000.
[3] T. G. Dietterich, "Machine-Learning Research," AI Magazine, vol. 18, no. 4, p. 97, 1997.
[4] 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.
[5] L. Breiman, "Bagging Predictors," Machine learning, vol. 24, no. 2, pp. 123-140, 1996.
[6] Y. Freund and R. E. Schapire, "Experiments with a new Boosting Algorithm," International Conference of Machine Learning, vol. 96, pp. 148-156, 1996.
[7] A. Vezhnevets and V. Vezhnevets, "Modest AdaBoost-teaching AdaBoost to generalize better," Graphicon, vol. 12, no. 5, pp. 987-997, 2005.
[8] C. Domingo and O. Watanabe, "MadaBoost: A Modification of AdaBoost," Conference on Computational Learning Theory, pp. 180-189, 2000.
[9] S. Avidan, "Spatialboost: Adding Spatial Reasoning to Adaboost," European Conference on Computer Vision, pp. 386-396, 2006.
[10] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5-32, 2001.
[11] M. Gönen and E. Alpaydın, "Multiple Kernel Learning Algorithms," Journal of Machine Learning Research, vol. 12, no. Jul, pp. 2211-2268, 2011.
[12] 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.
[13] D. H. Wolpert and W. G. Macready, "No Free Lunch Theorems for Optimization," IEEE Transactions on Evolutionary Computation, vol. 1, no. 1, pp. 67-82, 1997.
[14] H. R. Bonab and F. Can, "A Theoretical Framework on the Ideal Number of Classifiers for Online Ensembles in Data Streams," International Conference on Information and Knowledge Management, pp. 2053-2056, 2016
[15] T. M. Oshiro, P. S. Perez, and J. A. Baranauskas, "How Many Trees in a Rrandom Forest?," Internation Conference on Machine Learning and Data Mining, pp. 154-168, 2012.
[16] P. Latinne, O. Debeir, and C. Decaestecker, "Limiting the Number of Trees in Random Forests," Multiple Classifier Systems, pp. 178-187, 2001.
[17] R. W. Shephard and R. Färe, "The Law of Diminishing Returns," Zeitschrift für Nationalökonomie, vol. 34, no. 1-2, pp. 69-90, 1974.
[18] E. K. Tang, P. N. Suganthan, and X. Yao, "An Analysis of Diversity Measures," Machine Learning, vol. 65, no. 1, pp. 247-271, 2006.
[19] E. M. Dos Santos, R. Sabourin, and P. Maupin, "A Dynamic Overproduce-and-Choose Strategy for the Selection of Classifier Ensembles," Pattern Recognition, vol. 41, no. 10, pp. 2993-3009, 2008.
[20] H. Beigy, "Dynamic Classifier Selection Using Clustering for Spam Detection," Computational Intelligence and Data Mining Symposium, pp. 84-88, 2009.
[21] 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, pp. 1924-1931, 2017.
[22] 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 Optimization," International Joint Conference on Neural Networks, pp. 1-7, 2015.
[23] J. A. Lustosa Filho, A. M. Canuto, and J. C. Xavier, "An Analysis of Diversity Neasures for the Dynamic Design of Ensemble of Classifiers," International Joint Conference on Neural Networks, pp. 1-8, 2015.
[24] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do We Need Hundreds of Classifiers to Solve Real World Classification Problems," J. Mach. Learn. Res, vol. 15, no. 1, pp. 3133-3181, 2014.
[25] L. Zhang and P. N. Suganthan, "Benchmarking Ensemble Classifiers with Novel Co-Trained Kernal Ridge Regression and Random Vector Functional Link Ensembles," IEEE Computational Intelligence Magazine, vol. 12, no. 4, pp. 61-72, 2017.
[26] L. Zhang and P. N. Suganthan, "Oblique Decision Tree Ensemble via Multisurface Proximal Support Vector Machine," IEEE Transactions on Cybernetics, vol. 45, no. 10, pp. 2165-2176, 2015.
[27] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," in Proceedings of International Conference on Neural Networks, vol. 4, pp. 1942-1948, 1995.
[28] A. Jurek, Y. Bi, S. 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.
[29] A. Rahman and B. Verma, "Novel Layered Clustering-Based Approach for Generating Ensemble of Classifiers," IEEE Transactions on Neural Networks, vol. 22, no. 5, pp. 781-92, 2011.
[30] 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. 952-965, 2016.
[31] Y. Yang and J. Jiang, "Adaptive Bi-Weighting Toward Automatic Initialization and Model Selection for HMM-Based Hybrid Meta-Clustering Ensembles," IEEE Transactions on Cybernetics, vol. 49, no. 5, pp. 1657-1668, 2018.
[32] R. Xu and D. Wunsch, "Survey of Clustering Algorithms," IEEE Transactions on Neural Networks, vol. 16, no. 3, pp. 645-678, 2005.
[33] B. Verma and A. Rahman, "Cluster-Oriented Ensemble Classifier: Impact of Multicluster Characterization on Ensemble Classifier Learning," IEEE Transactions on Knowledge and Data Engineering, vol. 24, no. 4, pp. 605-618, 2012.
[34] S. Fletcher and B. Verma, "Removing Bias from Diverse Data Clusters for Ensemble Classification," International Conference on Neural Information Processing, pp. 140-149, 2017.
[35] Z. Jan, B. Verma, and S. Fletcher, "Optimizing Clustering to Promote Data Diversity when Generating an Ensemble Classifier," Proceedings of the Genetic and Evolutionary Computation Conference Companion, pp. 1402-1409, 2018.
[36] 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.
[37] U. Bhowan, M. Johnston, M. 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.
[38] M. Asafuddoula, B. Verma, and M. Zhang, "A Divide-and-Conquer Based Ensemble Classifier Learning by Means of Many-Objective

Optimization," IEEE Transactions on Evolutionary Computation,vol. 22, no. 5, pp. 762-777, 2017.
[39] Z. Lu, X. Wu, and J. C. Bongard, "Active Learning through Adaptive Heterogeneous Ensembling," IEEE Transactions on Knowledge and Data Engineering, vol. 27, no. 2, pp. 368-381, 2015.
[40] L.-y. Yang, J.-y. Zhang, and W.-j. Wang, "Cluster Ensemble based on Particle Swarm Optimization," WRI Global Congress on Intelligent Systems, vol. 3, pp. 519-523, 2009.
[41] H. J. Escalante, M. Montes, and E. Sucar, "Ensemble Particle Swarm Model Selection," International Joint Conference on Neural Networks, pp. 1-8, 2010.
[42] H. J. Escalante, M. M. y Gómez, and L. E. Sucar, "PSMS for Neural Networks on the IJCNN 2007 Agnostic vs Prior Knowledge challenge," in International Joint Conference on Neural Networks, pp. 678-683, 2007.
[43] H. J. Escalante, M. Montes, and L. E. Sucar, "Particle Swarm Model Selection," Journal of Machine Learning Research, vol. 10, pp. 405-440, 2009.
[44] H. J. Escalante, M. Montes, and L. Villaseñor, "Particle Swarm Model Selection for Authorship Verification," Iberoamerican Congress on Pattern Recognition, pp. 563-570, 2009.
[45] K. Bache and M. Lichman. (2013). UCI Machine Learning Repository. Available: http://archive.ics.uci.edu/ml/
[46] MATLAB, Statistics and Machine Learning Toolbox. Natick, Massachusetts: The MathWorks Inc., 2013.
[47] M. Eliasziw and A. Donner, "Application of the McNemar test to non-independent matched pair data," Statistics in medicine, vol. 10, no. 12, pp. 1981-1991, 1991.
[48] L. I. Kuncheva and J. J. Rodríguez, "A Weighted Voting Framework for Classifiers Ensembles," Knowledge and Information Systems, vol. 38, no. 2, pp. 259-275, 2014.
[49] F. Wilcoxon, "Individual Comparisons by Ranking Methods," Biometrics bulletin, vol. 1, no. 6, pp. 80-83, 1945.
[50] M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum and F. Hutter, "Efficient and Robust Automated Machine Learning," Advances in Neural Information Processing Systems, pp. 2962-2970, 2015.
[51] 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.
[52] B. Zhang, A. Qin, and T. Sellis, "Evolutionary feature subspaces generation for ensemble classification," in Proceedings of the Genetic and Evolutionary Computation Conference, ACM, pp. 577-584, 2018.
[53] I. Github, "IEEE_TKDE," URI: https://github.com/zohaibjan/IEEE_TKDE, 2020.
[54] H. Bonab and F. Can, " Less is More: A Comprehensive Framework for the Number of Components of Ensemble Classifiers", IEEE Transactions on Neural Network and Learning Systems, vol. 30, no. 9, pp. 2735-2745, 2019.