Optimizing clustering to promote data diversity when generating an ensemble classifier

Zohaib M Jan
Centre of Intelligent Systems
School of Engineering and Technology
Central Queensland University
Brisbane, Australia
z.jan@cqu.edu.au

Brijesh Verma
Centre of Intelligent Systems
School of Engineering and Technology
Central Queensland University
Brisbane, Australia
b.verma@cqu.edu.au

Sam Fletcher
Centre of Intelligent Systems
School of Engineering and Technology
Central Queensland University
Brisbane, Australia
s.fletcher@cqu.edu.

ABSTRACT

In this paper, we propose a method to generate an optimized ensemble classifier. In the proposed method, a diverse input space is created by clustering training data incrementally within a cycle. A cycle is one complete round that includes clustering, training, and error calculation. In each cycle, a random upper bound of clustering is chosen and data clusters are generated. A set of heterogeneous classifiers are trained on all generated clusters to promote structural diversity. An ensemble classifier is formed in each cycle and generalization error of that ensemble is calculated. This process is optimized to find the set of classifiers which can have the lowest generalization error. The process of optimization terminates when generalization error can no longer be minimized. The cycle with the lowest error is then selected and all trained classifiers of that particular cycle are passed to the next stage. Any classifier having lower accuracy than the average accuracy of the pool is discarded, and the remaining classifiers form the proposed ensemble classifier. The proposed ensemble classifier is tested on classification benchmark datasets from UCI repository. The results are compared with existing state-of-the-art ensemble classifier methods including Bagging and Boosting. It is demonstrated that the proposed ensemble classifier performs better than the existing ensemble methods.

KEYWORDS

Ensemble Classifiers, Evolutionary Algorithms, Particle Swarm Optimization, Clustering, Multi Classifier Systems, Neural Networks

1 INTRODUCTION

Ensemble classifier is a machine learning classification methodology where a number of classifiers are fused together to form a common decision (classification). Ensemble classifiers have shown success in various disciplines like weather forecasting, credit risk analysis, banking, medical diagnosis, and house pricing [1-5]. Essentially, the idea behind ensemble classifier is that a group of classifiers can produce better results than a single classifier. Additionally, a single classifier performing well on one dataset might not perform well on others as well [6]. A great body of research is presented in [1-5, 7-9], which states various methodologies for fusing classifiers. Two key factors to consider when fusing classifiers to form an ensemble classifier are classifier accuracy and classifier diversity. Classifiers that are fused to generate an ensemble classifier should at least be better than random guessing and make uncorrelated errors [9].

A number of previous research has pointed out the merits of diversity in ensemble classifiers [10, 11]. However, a trade-off must be maintained between accuracy and diversity. Precedence given to diversity only will result in an ensemble classifier that is diverse but performs inaccurately. Ultimately, the main objective of generating an ensemble classifier is to increase classification accuracy [3]. To maximise diversity and accuracy simultaneously, several ensemble classifier methods have been proposed. These methods can be generalized into three categories i) random sub sampling of training dataset, ii) feature randomization of training dataset, and iii) parameter randomization. These methodologies are elaborated in [2, 4].

In relation to diversity through sub sampling, two pioneering works to consider are Bagging[10] and Boosting[12]. Bagging works by creating sub samples of data with repeating and unique groups. Classifiers are trained on each sub sample of the dataset, which are then fused together using majority voting. Boosting on the other hand subsequently trains a classifier on the 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 in [13-15]. A renowned work in achieving diversity through feature randomization is Random Forest [16]. 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 [17], secondly are methodologies that use evolutionary algorithms to manipulate features and/or ensemble classifier components [18].

GECCO '18 Companion, July 15–19, 2018, Kyoto, Japan
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-5764-7/18/07…$15.00
https://doi.org/10.1145/3205651.3208245

In recent works, authors have created a diverse input space from the training dataset using clustering [19]. In [20] authors proposed an incremental ensemble classifier process. Input data were partitioned into a number of clusters on which base classifiers were trained. Classifier accuracy and diversity were then calculated for all classifiers, and with incremental layered approach, each new classifier with higher accuracy was added to the ensemble. A classifier is added to the pool if, either accuracy is greater than the last classifier in the pool, or accuracy is the same, but diversity has increased, otherwise the classifier is discarded. The number of data clusters generated are increased iteratively, which are generated by k-means. Similarly, in [21] data diversity was achieved by not only clustering, but also discarding similar clusters using Jaccard index. Any cluster having Jaccard index value higher than a given threshold was discarded. It was also suggested that using a maximum value of clustering with k-means should be $\sqrt[3]{n}$, where n is the number of records in the training dataset. This not only ensured that the computational complexity of the algorithm remains $O(n^2)$, but also kept the algorithm from creating clusters with few records in them. In another research [22] data diversity was achieved through clustering using k-means algorithm by partitioning the dataset into atomic and non-atomic clusters. An atomic cluster is class pure, whereas a non-atomic cluster is not. Every non-atomic cluster is converted to an atomic cluster through a 2-layer neural network classifier. The process is repeated till every non-atomic cluster is converted to an atomic cluster. When all clusters are atomic, decisions can be formed as every cluster is class pure.

There has also been a lot of research into the benefits of using evolutionary algorithms in order to generate ensemble classifiers. In [18] the authors suggested that achieving high diversity with accuracy can be classified as a multi objective optimization problem, and that using evolutionary algorithms such as Genetic Algorithms (GA) can be beneficial in this regard. Data are partitioned into distinct clusters incrementally in a layered fashion and at each layer, a different value of K clusters were generated with a maximum value of K set to be n (number of records). GA is used to find the optimum trade-off between accuracy and diversity, and the process of incrementally clustering terminates when a global optimum is achieved.

In [23] authors suggested adding a layer of entropy between meta classifiers in stack generalization. They suggested that using diversity as a measure to select the best set of base classifiers will have a lesser impact on computational resources. Oracle output and entropy measures were used as inputs for optimization algorithm. In [24] authors proposed clustering ensemble classifiers using Particle Swarm Optimization (PSO). The weight of each cluster was calculated using PSO. Each cluster was treated as a particle in n dimensions, which was then given a relative weight using classical PSO. This proved not only efficient in terms of generalization error but also effective in lowering the complexity of the ensemble. Similarly, in [25] authors suggested using PSO as a model selection tool in order to select the best set of classifiers to form an ensemble classifier. It was argued that traditional model selection methodologies tend to focus on maximising an individual model's accuracy rather than promoting diversity amongst different models in the pool consequently, promoting higher global ensemble accuracy. Popular model selection approach, which overcomes this problem, is known as Particle Swarm Model Selection (PSMS). In recent studies [25-28], PSMS has shown a remarkable success and proved to be a good contender for optimizing a binary search space. PSMS was also used to find the best set of features, as a model selection tool, and for parameter optimization for classification dataset.

Although several methods use clustering to create a diverse input space [18, 20-22, 29], the need for an optimum value of K to create a diverse input space needs careful consideration. A single value of K for different datasets is not an ideal solution; a dynamic way to adjust K which is reflective of dataset's dimensions should be investigated. Additionally, various methodologies have been proposed, which argue the benefit of combining ensemble classifier with evolutionary algorithms [25-27] to optimize various hyper-parameters. However, a careful consideration in the use of evolutionary algorithms to create a diverse input space is essential.

In this paper, we use a set of heterogeneous classifiers to generate an accurate and diverse ensemble classifier. Furthermore, certain classifier parameters are randomized to further promote structural classifier diversity. We generate a diverse input space by clustering dataset and filtering redundant clusters. The input space is optimized with an evolutionary algorithm, giving precedence to the overall classification accuracy of the ensemble classifier by dynamically optimizing K.

The original contributions of this paper are (i) identifying the optimum value of K to generate heterogeneous data clusters, which can increase the classification accuracy of the ensemble classifier; (ii) analysing the impact of using an evolutionary algorithm to optimize the diverse input space on overall classification accuracy of the ensemble; and (iii) comparing the proposed method with existing state of the art ensemble approaches including Bagging and Boosting.

The next sections of the paper are outlined as follows. Section 2 presents the proposed ensemble classifier methodology. Section 3 entails the experimental setup, results and comparative analysis. Section 4 concludes the paper and provides future directions.

2 PROPOSED METHOD

The proposed method starts with the training data, validation data, a set of base classifiers, and an initial value of K. The training data is passed to the optimization process, which generates data clusters from $k = 1, \dots, \mathrm{K}$. All generated clusters are filtered based on similarity, and on all remaining clusters, a set of base classifiers are trained. An ensemble classifier is then generated using all of the trained classifiers; in the fitness function of the optimization process, generalization error of the ensemble is calculated using validation data. All of the trained classifiers along with value of K, and error are stored. The process of optimization repeats with a different random value of K in each cycle until error can no longer be minimized or no relative change in error occurs. The method then passes all of the trained classifiers and the value of K, of the cycle, which had the lowest error to the next stage. The average accuracy of the pool of classifiers is calculated and any classifier performing lower than a threshold gets discarded. The remaining classifiers are then utilized to generate the proposed ensemble classifier and classification accuracy is calculated with an unseen testing dataset.

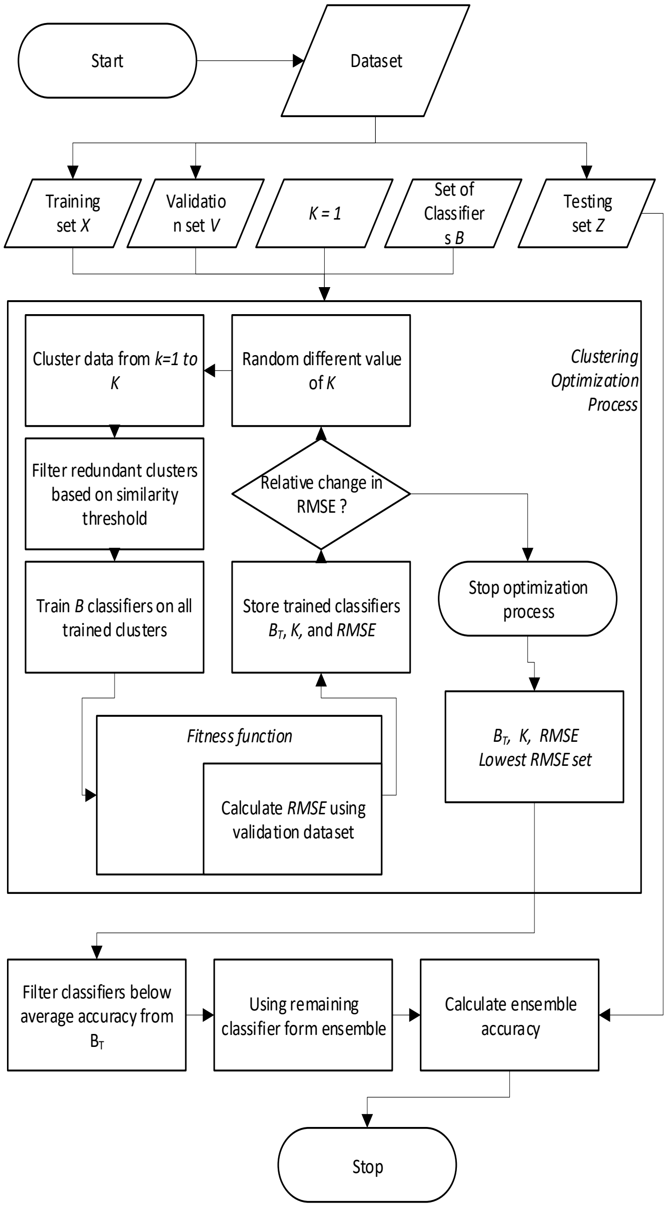
Start
Dataset
Training set X
Validation set V
K = 1
Set of Classifiers B
Testing set Z
Cluster data from k=1 to K
Random different value of K
Clustering Optimization Process
Filter redundant clusters based on similarity threshold
Relative change in RMSE ?
Train B classifiers on all trained clusters
Store trained classifiers B_T, K, and RMSE
Stop optimization process
Fitness function
Calculate RMSE using validation dataset
B_T, K, RMSE Lowest RMSE set
Filter classifiers below average accuracy from B_T
Using remaining classifier form ensemble
Calculate ensemble accuracy
Stop

Figure 1: Flowchart of the proposed ensemble classifier method

Fig. 1 shows the flow chart of the proposed method and the subsequent subsections provide details on the different components involved in the proposed ensemble classifier methodology.

2.1 Generating diverse input space

In contrast with generating random sub samples of the dataset as in Bagging, the proposed methodology incrementally generates data clusters from the training data in each cycle. The process starts with an initial value of $K = 1$ and goes up to a maximum of $K = n$, where n is the number of records in the training data. In each cycle, the number of generated data clusters are $j = K(K + 1)/2$.

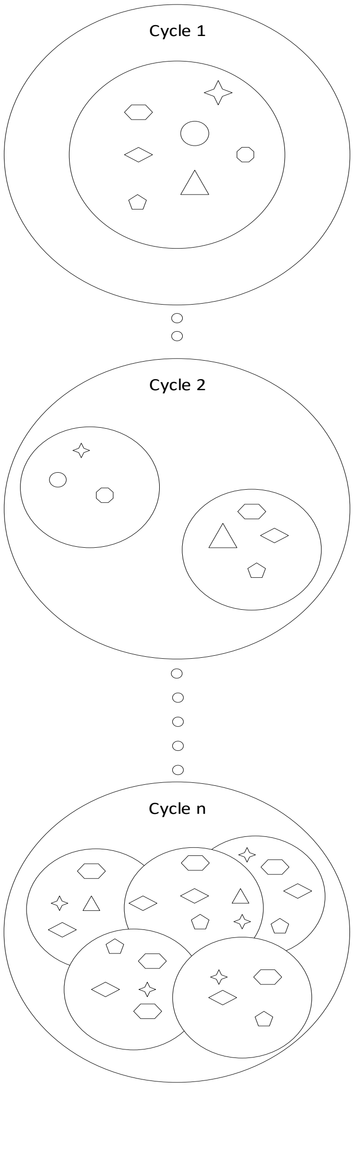

Figure 2: Incrementally clustering dataset. Different shapes are different data patterns.

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 each x is a feature vector containing m discrete or continuous valued features $< x_{n,1}, x_{n,2}, \ldots, x_{n,m} >$ and y_n is the respective discrete class label. Clustering is done without the inclusion of class labels in the dataset so that the decision boundaries are not reflective of class labels. The process of clustering is summarized in Fig. 2. It can be noted from Fig. 2 that in each cycle a different number of clusters are generated and, in some cycles, as in the nth cycle, cluster overlap occurs.

Each cluster Ω will contain a set of observations and is given as $\Omega_n = \{x_1, x_2, \ldots, x_n\}$. In each cycle, the total generated clusters are $\Omega = \{\Omega_1, \Omega_2, \ldots, \Omega_j\}$. Clusters are checked for inter cluster similarity in order to discard redundant clusters using Equation (1).

$$s\left(\Omega_i,\ \Omega_j\right) = \frac{\left|\Omega_i, \cap\ \Omega_j\right|}{\left|\Omega_i \cup \Omega_j\right|} \quad \forall\, i,j\ \in\ \Omega \quad and\, i\ \neq j \qquad (1)$$

The value of s varies from 0 to 1, with 1 implying the two clusters are 100% identical to each other and 0 implying there are no repeating records.

2.2 Classifier training

In the proposed method, a set of heterogeneous base classifiers are used, which include Naïve Bayes (NB), Discriminant Analysis (DISCR), k-Nearest Neighbour (kNN), Decision Trees (DT), Neural Networks (NN), and Support Vector Machines (SVM). These classifiers have different learning capabilities and certain classifier parameters are randomized per cluster, which introduces classifier structural diversity in the ensemble. The hyper parameters and their respective selection criteria are given in Table 2. Therefore, if the total number of clusters after optimization and filtering is k ($k < j$), and on each cluster a set of B base classifiers are trained, then the total trained classifiers are $l = \ k \times B$.

2.3 Optimizing the input space

The process of optimization takes in all trained classifiers $B_T = \{\ \varphi_1, \varphi_2, \dots, \varphi_l\}$, and the validation dataset V having feature vector x and class labels y. Each classifier φ_l from the set of trained classifiers is used to classify the feature vector of validation dataset using classification function $p(\varphi, x) = \ y$, and their respective results are stored in a decision matrix d as shown below in Equation (2).

$$d = \begin{bmatrix} y'_{1,1} & \dots\dots & y'_{1,l} \\ y'_{2,1} & \dots.. & y'_{2,l} \\ \vdots & & \vdots \\ y'_{n,1} & \dots.. & y'_{n,l} \end{bmatrix} \qquad (2)$$

where $y'_{n,l}$ is the classification result of $p(\varphi_l, x_n)$

The proposed approach uses PSO to optimize the process of clustering by decreasing overall generalization error. To understand this process, we first define the PSO search space. In the PSO, we have a population of particles, with each particle having a personal best and a global best. Each particle in the proposed approach is the index of the respective classifier's column in the decision matrix d, so if there are l classifiers then there are l particles. Therefore, the personal best is the possible inclusion of that particular column of responses y' of a classifier. This is given by a row vector as:

$$rv\ =\ [\mathbf{1}(\mathbf{c_1}), \mathbf{1}(\mathbf{c_2}), \dots, \mathbf{1}(\mathbf{c}_n)] \qquad (3)$$

$$\text{where } \mathbf{1}(\mathbf{c}_n) \colon \begin{cases} 0\ if \quad n \notin q \\ 1\, if \quad n \in q \end{cases}$$

where q is the pool of classifiers

The row vector has an upper bound of $[1, 1, \dots, 1_l]$, and a lower bound of $[0,0, \dots\ , 0_l]$. Any element in the row vector, which has a 0 value indicates that the classification labels of that column will not be selected, and 1 meaning otherwise. Class labels y' of only those columns which have a respective 1 in the row vector are selected and a mode is taken or simply majority voting. Using the result, we can calculate the root mean square error of the ensemble as given in Equation (4).

$$RMSE\ = \frac{\sqrt{\sum_{i=1}^{|V|}(y_i - y'_i)^2}}{|V|} \qquad \forall\, y\ \in V \qquad (4)$$

where y' is the predicted class label and y is the actual class label of the validation dataset.

Error in (4) is used as the fitness function and the optimization process searches in the binary search space to find the best set of trained classifiers that can minimize (4) globally. The process of optimization terminates when (4) can no longer be minimized. At this stage, the proposed approach yields the set of trained classifiers that had the lowest RMSE and the value of K chosen in that particular cycle.

2.4 Classifier filtering

Average accuracy of the set of the optimized trained classifiers is calculated using Equation (5).

$$avg\_accuracy = \frac{1}{c}\sum_{i=1}^{c} acc_i \qquad (5)$$

Where acc is the individual classification accuracy of each classifier in the pool on validation dataset and is calculated using Equation (6).

$$acc = \frac{1}{|V|}\sum_{i=1}^{m} \mathbf{1}(p(\varphi, x_i)) \ \ \forall\, x\ \in V \qquad (6)$$

$$\text{where } \mathbf{1}\big(p(\varphi, x)\big) \colon = \begin{cases} 0\ \ if\ p(\varphi, x_i)\ =\ y_i \\ 1\ \ if\ p(\varphi, x_i)\ \neq\ y_i \end{cases}$$

Any classifier in the pool, that has lower classification accuracy than the average classification accuracy of the pool, gets discarded.

2.5 Decision

The classification is done using the set of filtered classifiers and the feature vector x from the unseen dataset test set Z . All classifications labels y' from each classifier are then combined via majority voting. Generalization error is calculated using (4) and the resultant error is the error of the proposed ensemble classifier.

3 EXPERIMENTAL STUDY AND ANALYSIS

In this section, we present several experiments to test the efficacy of the proposed ensemble classifier method on a set of benchmark datasets from UCI Machine Learning repository [30]. The details of these datasets are given in Table 1. We also compared our experimental results with existing state of the art ensemble techniques.

Table 1: Datasets

Dataset	Number of Features	Number of Records	Number of Class Labels
Breast Cancer	9	683	2
Ecoli	7	336	2
Glass	10	214	7
Haberman	3	306	2
Ionosphere	33	351	2
Iris	4	150	3
Vehicle	18	946	4

Table 2: Experimentation Parameters

Algorithm / Classifier	Parameter	Values
Neural network	Hidden neuron	Random between: 10 to 30
	Training function	Conjugate gradient descent back propagation
	Number of epochs	1000
	Error goal	1e-5
Multi class support vector machine	Kernel function	Radial basis function
	Iteration limit	Random between: 1000 to 5000
Naïve Bayes	Distribution function	Kernels
K-Nearest neighbour	Number of neighbours	Random between: 4 to 10
Decision tree	Minimum leaf size	No of class labels
Discriminant analysis	Kernel function	Polynomial
K-means	Number of iteration	2400
Particle swarm optimization	Maximum iteration	100
	Stall iteration	10
	Swarm size	|classifiers|

3.1 Experimental setup

We used 10-fold cross validation for experimentation purposes. A set of heterogeneous classifiers i.e. ANN, kNN, SVM, DT, NB, and DISCR were used. MATLAB R2017a was used for experimentation using default implementations of *k*-Means algorithm for clustering, PSO for optimization, and base classifiers. Mostly, default parameters were used in addition to that mentioned in Table 2. These parameters were chosen on a trial and error basis.

3.2 Results

In Table 3, experimentation results entailing average classification accuracy, standard deviation, total clusters generated by the algorithm, and total clusters utilized by the algorithm per dataset are given. The proposed ensemble classifier was tested on Breast Cancer, Ecoli, Glass, Haberman, Ionosphere, Iris, and Vehicle datasets, and it achieved classification accuracies of 0.967, 0.957, 0.989, 0.757, 0.923, 0.977, and 0.903 respectively.

Across all datasets, 66% of clusters were utilized and the remaining 34% clusters were discarded on the basis of similarity. The similarity threshold of s chosen for these experimentations was 0.9, and this was empirically calculated on trial and error basis as it resulted in the highest classification accuracy. The proposed technique dynamically calculated the optimum value of K for each dataset; however, it is crucial to observe that K is approximately 3.5 for most of the datasets.

Table 3: Classification accuracy, and clustering information of the proposed approach

Datasets	Avg. Classification Accuracy	Standard deviation	Total Clusters	Clusters Utilized
Breast cancer	0.967	0.0029	17	12
Ecoli	0.957	0.0007	3	2
Glass	0.989	0.0009	2	1
Haberman	0.757	0.0046	9	7
Ionosphere	0.923	0.0016	4	3
Iris	0.977	0.0025	2	1
Vehicle	0.903	0.0016	11	8

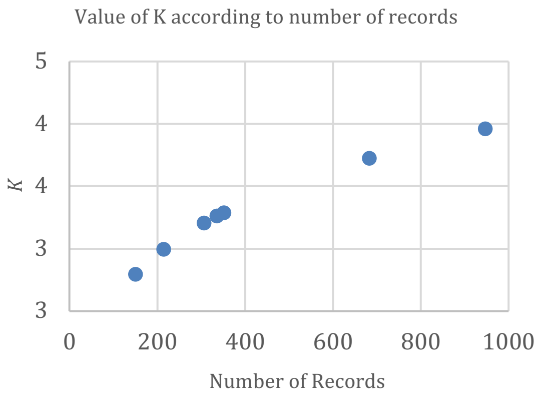

Figure 3: K with respect to number of records

We can see from Fig. 3 that a larger value of K is selected as the number of records in the dataset increases. It, is however apparent that the average value of K is 3.8 for datasets having number of records ranging from 300 to 950.

3.3 Comparative analysis

We compared the proposed approach with existing state of the art ensemble classifier methods presented in OEC-ILC [20], MPRaF-T [31], REC [32], Bagging [29], Boosting [29] and Random Forest. Default implementation of Random Forest in MATLAB[33] was used for comparison using the “*bag*” parameter for *fitcensemble* function. The results are given in Table 4, with the highest accuracies given in bold. It can be noted that the proposed approach performed significantly better than other approaches in 4 out of 7 datasets. Also observed was that the proposed ensemble classifier method had approximately 1.03% performance improvement over OEC-ILC, 1.05% over MPRaF-T, 1.09% over REC, 1.03% over Bagging, 1.05% over Boosting, and 1.05% over Random Forest. The results therefore, denote the efficacy of the proposed ensemble classifier. In order to test the significance of the results, we conducted non-parametric Wilcoxon Signed Rank Test for paired samples. The ranks given are (-) if the proposed approach outperformed the given approach, (+) if the proposed approach did not perform better, and (=) if there is no difference in the performance. The p values of t-test are also given in Table 4, which were calculated with an alpha value of 0.05 and null hypothesis is rejected at a p value lower than 0.05.

Table 4: Comparison of classification accuracy of the proposed approach with OEC - ILC, MPRaF- T, REC, Bagging, Boosting, and Random Forest. Best results shown in bold.

Datasets	**Proposed approach**	**OEC - ILC [20] *p*=0.064**	**MPRaF –T [31] *p*=0.0373**	**REC [32] *p*=0.0139**	**Bagging [29] *p*=0.031**	**Boosting [29] *p*=0.014**	**Random Forest [33] *p*=0.0232**
Breast cancer	0.967	0.965(-)	0.967(=)	0.960(-)	**0.9778(+)**	0.9694(+)	0.967(=)
Ecoli	**0.957**	0.856(-)	0.855(-)	0.829(-)	0.8867(-)	0.8890(-)	0.865(-)
Glass	**0.989**	0.966(-)	0.942(-)	0.730(-)	0.9591(-)	0.9545(-)	0.977(-)
Haberman	0.757	**0.782(+)**	0.723(-)	N/A	0.7420(-)	0.6637(-)	0.686(-)
Ionosphere	0.923	0.916(-)	0.926(+)	0.917(-)	0.9136(-)	0.9000(-)	**0.934(+)**
Iris	**0.977**	0.960(-)	0.976(-)	0.937(-)	0.9667(-)	0.9733(-)	0.946(-)
Vehicle	**0.903**	0.810(-)	0.763(-)	0.725(-)	0.8424(-)	0.8096(-)	0.757(-)

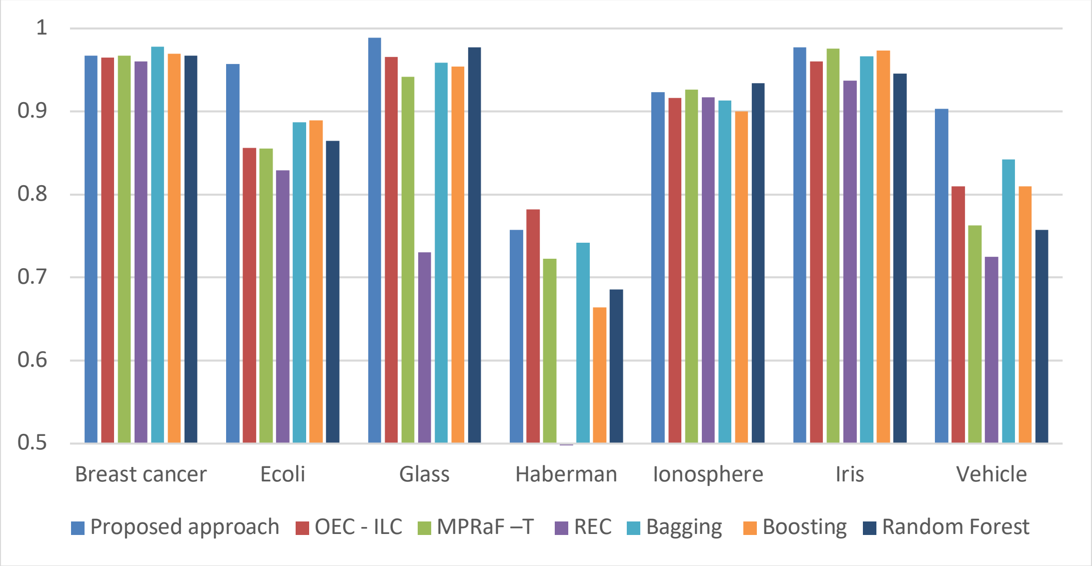

Figure 4: Comparative analysis of classification accuracy among proposed method, OEC-ILC, MPRaf-T, REC, Bagging, Boosting, and Random Forest.

4 CONCLUSIONS

This paper presented a method for creating an optimized ensemble classifier. The method dynamically generated a diverse input space by clustering data into a set of heterogeneous clusters. Data clusters are generated in a cyclical approach incrementally. In each cycle, a different value of upper bound of clustering was selected and data clusters were generated. A set of heterogeneous classifiers, with random parameters, were trained on each cluster and an ensemble classifier was generated in each cycle. Generalization error of the ensemble classifier was calculated against validation dataset. The process was repeated in each cycle until the generalization error can no longer be minimized or no relative change can occur. The cycle with the lowest error was selected by the proposed method and all trained classifiers of that cycle were transferred to the next stage. The average accuracy of the pool of classifier was calculated and any classifier in the pool, having accuracy value lower than the average accuracy of the pool was discarded. The remaining classifiers form the ensemble classifier. In contrast with selecting a single value of K for clustering, the proposed method dynamically selects the optimum K , which can achieve the highest classification accuracy.

The proposed method was compared against popular and recent ensemble classifier methods including Bagging, Boosting, Random Forest, MPRaF-T, OEC-ILC, and REC, on 7 classification benchmark datasets from UCI repository. It was experimentally demonstrated that the ensemble classifier created using the proposed method performed significantly better on 4 out of 7 datasets. The higher results achieved by the proposed ensemble classifier method are attributed to the incorporation of data diversity by optimizing clustering and structural diversity by using a set of heterogeneous classifiers, and randomizing classifier parameters.

Although the proposed ensemble classifier method achieved high classification accuracy, we will further evaluate the proposed method on additional benchmark and real-world datasets. We will also evaluate the impact of classifier types and parameters in future research.

ACKNOWLEDGMENTS

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

REFERENCES

[1] T. K. Ho, J. J. Hull, and S. N. Srihari, "Decision combination in multiple classifier systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 16, no. 1, pp. 66-75, 1994.

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

[3] 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.

[4] Z.-H. Zhou, *Ensemble methods: foundations and algorithms*. CRC Press, 2012.

[5] M. Woźniak, M. Graña, and E. Corchado, "A survey of multiple classifier systems as hybrid systems," *Information Fusion,* vol. 16, pp. 3-17, 2014.

[6] 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.

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

[8] J. Abellán and J. G. Castellano, "A comparative study on base classifiers in ensemble methods for credit scoring," *Expert Systems with Applications,* vol. 73, pp. 1-10, 2017.

[9] M.-J. Kim and D.-K. Kang, "Classifiers selection in ensembles using genetic algorithms for bankruptcy prediction," *Expert Systems with Applications,* vol. 39, no. 10, pp. 9308-9314, 2012.

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

[11] G. Rätsch, T. Onoda, and K.-R. Müller, "Soft margins for AdaBoost," *Machine Learning,* vol. 42, no. 3, pp. 287-320, 2001.

[12] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *International Conference on Machine Learning*, Bari, Italy, 1996, vol. 96, pp. 148-156.

[13] A. Vezhnevets and V. Vezhnevets, "Modest AdaBoost-teaching AdaBoost to generalize better," in *Graphicon*, 2005, vol. 12, no. 5, pp. 987-997.

[14] C. Domingo and O. Watanabe, "MadaBoost: A modification of AdaBoost," in *Conference on Learning Theory*, 2000, pp. 180-189.

[15] S. Avidan, "Spatialboost: Adding spatial reasoning to adaboost," in *European Conference on Computer Vision*, 2006, pp. 386-396: Springer.

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

[17] M. Gönen and E. Alpaydın, "Multiple kernel learning algorithms," *Journal of Machine Learning Research,* vol. 12, no. Jul, pp. 2211-2268, 2011.

[18] 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.

[19] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks,* vol. 16, no. 3, pp. 645-678, 2005.

[20] M. Asafuddoula, B. Verma, and M. Zhang, "An incremental ensemble classifier learning by means of a rule-based accuracy and diversity comparison," in *International Joint Conference on Neural Networks*, 2017, pp. 1924-1931.

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

[22] 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, May 2011.

[23] H. Kadkhodaei and A. M. E. Moghadam, "An entropy based approach to find the best combination of the base classifiers in ensemble classifiers based on stack generalization," in *International Conference on Control, Instrumentation, and Automation*, 2016, pp. 425-429.

[24] L.-y. Yang, J.-y. Zhang, and W.-j. Wang, "Cluster ensemble based on particle swarm optimization," in *WRI Global Congress on Intelligent Systems*, 2009, vol. 3, pp. 519-523.

[25] H. J. Escalante, M. Montes, and E. Sucar, "Ensemble particle swarm model selection," in *International Joint Conference on Neural Networks*, 2010, pp. 1-8.

[26] 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*, 2007, pp. 678-683.

[27] H. J. Escalante, M. Montes, and L. E. Sucar, "Particle swarm model selection," *Journal of Machine Learning Research,* vol. 10, pp. 405-440, 2009.

[28] H. J. Escalante, M. Montes, and L. Villaseñor, "Particle swarm model selection for authorship verification," in *Iberoamerican Congress on Pattern Recognition*, 2009, pp. 563-570.

[29] 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.

[30] K. Bache and M. Lichman. (2013). *UCI machine learning repository*. Available: http://archive.ics.uci.edu/ml/

[31] 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.

[32] 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.

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