Optimal Clusters Generation for Maximizing Ensemble Classifier Performance

Zohaib Jan
Center of Intelligent Systems
Central Queensland University
Brisbane 4000, Australia
z.jan@cqu.edu.au

Brijesh Verma
Center of Intelligent Systems
Central Queensland University
Brisbane 4000, Australia
b.verma@cqu.edu.au

***Abstract*— Clustering based ensemble classifiers have seen a lot of focus recently because of their ability to effectively classify real-world noisy datasets. One way of incorporating clustering in ensembles is to utilize clustering algorithms such as *k-means* to generate a pool of data clusters. This is done to generate a random subspace on which base classifiers are trained, as opposed to bagging. One key parameter to clustering algorithms is the number of clusters *i.e.* the number of groups the data should be partitioned into; this is commonly known as the variable *K*. Most of the existing approaches either determine the value of *K* through trial and error or use some derived formulae-based approach. The problem firstly is that using a static value of *K* for different datasets is not ideal and although a certain value may work well for one dataset it may not work well for others. Secondly, calculating the value of *K* using a formulae-based approach using the raw data is not effective either as an unbalanced data can have a negative effect on the derived value. Therefore, in this paper we first segregate the data based on the data classes and then on each data class we perform a Silhouette analysis to determine the optimal number of clusters each data class should be separated into. The generated clusters which are class pure and are balanced by adding samples from other classes that are closest to the cluster centroid. In this manner we generate a random subspace of an augmented data that is composed of class balanced data clusters. On all balanced data clusters, a diverse set of base classifiers is trained, and an ensemble is formed. The proposed ensemble approach is tested on 16 benchmark UCI datasets and results are compared with single classifiers, as well as state-of-the-art ensemble classifier approaches. A set of non-parametric tests are also adopted to further validate the efficacy of the results.**

Keywords—clustering, ensemble classifiers, machine learning.

I. INTRODUCTION

Ensemble classifier is a methodology of combining multiple classifiers suitably in order to surpass the generalization plateau of single classifiers [1]. That is why ensemble classifiers are also used synonymously with "committee of experts". Ensemble classifiers besides being accurate are also robust and can classify real-world noisy datasets, making them vastly applicable in different areas of research/sciences including the financial sector, environmental sciences, medicine, transport, and image processing [2, 3]. Ensemble classifiers are robust in nature because they predominantly benefit from the "perturb and combine" strategy [4] and are able to outperform single classifiers because their decisions are not based on the performance of a single model. Additionally, single classifier performing well on one dataset may not perform well on others as stated in the "no free lunch" theorem [5].

Ensemble classifier generation methodology can be divided into two parts. Firstly, a given input data is "perturbed" or partitioned then base classifiers are trained on different parts. A common methodology of perturbing input data is random subspace method [6, 7]. Essentially random subsamples of the input data are generated with repeating and unique records. Another name for this strategy is "bagging" [8], where "bags" of input data are generated and base classifiers are trained on different bags. The idea behind this is that since each base classifier is trained on a separate bag this provides a means of controlling the variance of different classifiers. We can utilize bagging to generate bags of input samples known as attribute bagging or bags of input features known as feature bagging [9]. Many ensemble strategies have been proposed that utilize these strategies for further details readers can refer to [10]. Boosting is another sub-sampling-based ensemble strategy and many boosting-based ensemble classifiers have been proposed in research for further details readers can refer to [11-15]. Boosting works by training base classifiers on input data samples which were wrongly classified by previously trained base classifiers. Random Forest (RaF) is another example of an ensemble classifier that exploits bagging by training a multitude of Decision Trees (DT) on feature sub-samples of the input data [16]. RaF has been widely accepted as one of the most versatile and robust ensemble classifier that has been able to successfully classify noisy datasets and many variations of RaF have been proposed in research [17].

As for the second part different classifier selection methodologies have been incorporated to generate ensemble classifiers. Instead of utilizing all trained base classifiers from the pool a subset of base classifiers is utilized to generate an ensemble that can achieve the maximum generalisation performance. Since selecting a subset of classifiers is a combinatorial problem and large scale combinatorial problems are classified as NP-hard problems [18]. Therefore, researchers have proposed either rule-based classifier selection methodologies [19] or have utilized various optimization algorithms to optimize the pool of classifiers [20]. Ensemble classifier methodologies that exploit both the feature space and input sample space are classified as hybrid ensemble classifier approaches. These approaches exploit both input sample space, and feature space; and some even incorporated evolutionary sample selection processes to optimize the input space which can maximize the classification accuracy of the ensemble [21-23].

Some authors have utilized clustering as a substitute to bagging to generate a random subspace. Such ensemble classifiers are categorized as clustering-based ensembles [24-26]. In clustering-based ensemble classifiers the input data is first partitioned into several sparse data clusters and a set of diverse base classifiers is trained on each data cluster. This way ensemble incorporates diversity in two folds: firstly, since a set of diverse base classifier is utilized which contains classifiers that are structurally different (for example,

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

Artificial Neural Networks (ANN), Discriminant Analysis (DISCR), k-Nearest Neighbour (KNN), Naïve Bayes (NB), Support Vector Machine (SVM), *etc.*) therefore, each classifier brings with it different learning capabilities; secondly since each set of diverse classifiers is trained on a different data cluster using which they learn different aspects of the input data and, therefore, are different from each other. Another added benefit of clustering-based ensembles is that since each data cluster represents a dense local region of the decision boundary the classifiers trained on such data clusters have local expertise. Essentially this process is the reverse of a kernel function, instead of training a complex classifier to learn a decision boundary in higher dimensions we can break a complex decision boundary into smaller simpler decision boundary regions which can be learned by classifiers locally. Due to its robustness and versatility many clustering based ensemble classifier approaches have been proposed in research.

Although, clustering has been successfully employed by ensemble classifiers to either generate a random subspace or to group classifiers together from the pool and have proven to be very applicable with noisy datasets; there are some lurking limitations in clustering-based ensembles that need further consideration. Firstly, due to randomness and noise in the datasets the data clusters generated might not guarantee a noise free data cluster; secondly using a general rule of fixed upper bounds of clustering for different dataset is not an ideal methodology because datasets have different intrinsic and extrinsic characteristics and one rule which may work in one instance may not work well in every other instance; thirdly, due to class imbalances in the datasets the generated data clusters will also be class imbalanced and any base classifier trained on such a data cluster will eventually be biased. Therefore, in this research we propose a novel ensemble classifier approach that generates a random subspace by clustering the input data based on their classes. For example, if there are two classes in the input data {1, 2} (malignant or benign as in case of Wisconsin Breast Cancer dataset) then there will be two subsets of feature vectors x^1 and x^2, with x^1 being the feature vector belonging to class 1, and x^2 being the feature vector belonging to class 2. Each vector is then clustered to generate a random subspace, but instead of generating data clusters incrementally we conduct a *Silhouette* [27] analysis to identify the optimal number of clusters. Very briefly *Silhouette* analysis determines how well a data point is associated to its cluster. For further details readers can refer to the respective paper. This is done so that only optimal number of data clusters are generated of each class instead of using a fixed upper bound that is constant for different datasets. Additionally, data is clustered based on classes to avoid class imbalances lurking into data clusters as each data cluster will now be balanced by adding samples from other classes that are closest to its centroid. This ensures that class imbalances are accounted for and generated data clusters are balanced and suitable for the training of base classifiers. The novel contributions of this paper are:

- A methodology of determining the optimum number of data clusters for each data class.
- A methodology of generating balanced data clusters.
- A methodology of generating an ensemble classifier using all class balanced data clusters.

The rest of the paper is organized as follows: Section II discusses the proposed methodology, Section III entails the experiments and the results, and Section IV provides conclusion and gives future directions.

II. Proposed Methodology

A. Prilimanaries

To generate data clusters from the input data consider a data set $X = \{(\mathrm{x}_1, \mathrm{y}_1), (\mathrm{x}_2, \mathrm{y}_2), \dots, (\mathrm{x}_\mathrm{n}, \mathrm{y}_\mathrm{n})\}$ having feature vectors $\mathrm{x}_\mathrm{i} \in \mathbb{R}^\mathrm{D}$ each associated with a class label $\mathrm{y}_\mathrm{i} \in 1, .., V$ where V is the number of discrete class labels in a dataset. Then clustering is achieved by minimizing the squared Euclidean distance from a given centroid c given as:

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

where x is a feature vector, k is the number of data clusters and $euc(x, c)$ denotes the squared Euclidean distance given as:

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

To validate or assess the quality of generated data clusters, also to identify how many data clusters k should be generated for each data class. Firstly, the dataset is partitioned into its various classes as $X^1, X^2, \dots, X^V \in X$ where $X^V \subseteq X$, then the following holds true $X^1 \cup X^2, \dots \cup X^V = X$ and $X^1 \cap X^V = \emptyset$. This is done by performing a *Silhouette* analysis of each subset calculated as:

$$s(i) = \begin{cases} 1 - \frac{a(i)}{b(i)}, & if a(i) < b(i) \\ 0, & if (ai) = b(i) \\ \frac{b(i)}{a(i)} - 1, & if a(i) > b(i) \end{cases} \tag{3}$$

where $a(i)$ is the similarity of a data point i, and b(i) is the dissimilarity of a data point i in a cluster C. For further details please refer to [33].

The *Silhouette* score ranges between {-1, 1} with a small value of s meaning that a datapoint i is well matched to its given cluster and a larger value meaning otherwise. By conducting a *Silhouette* analysis of each data subset X^n we identify the optimum number of data clusters that must be generated to partition the data subset efficiently.

B. Ensemble generation framework

The proposed ensemble framework starts off with a training input data. The input data is then partitioned into its respective classes and optimal number of data clusters are generated of each class after conducting a *Silhouette* analysis. The benefit of clustering here is in three folds: firstly, by clustering the input data we are breaking a rather complex decision boundary into its smaller constituents which is essentially the opposite of what a kernel function does. This enables us to train simple classifiers with local expertise by training them on data clusters which represents dense local regions. Secondly, by clustering dataset based on the number of classes we are identifying how many variations exists between data patterns of each class; put simply, metaphorically speaking if we have a dataset of two classes cats and dogs then we are identifying how many different types of cats and dogs exist and which type of dogs is like cats. Lastly by clustering we generate a rich and diverse input space

that not only enables us to train a multitude of base classifiers on a single input data but also provides a mean of managing the bias and variance of classifiers in an ensemble. A flow chart of the proposed ensemble classifier generation framework is given in figure 1.

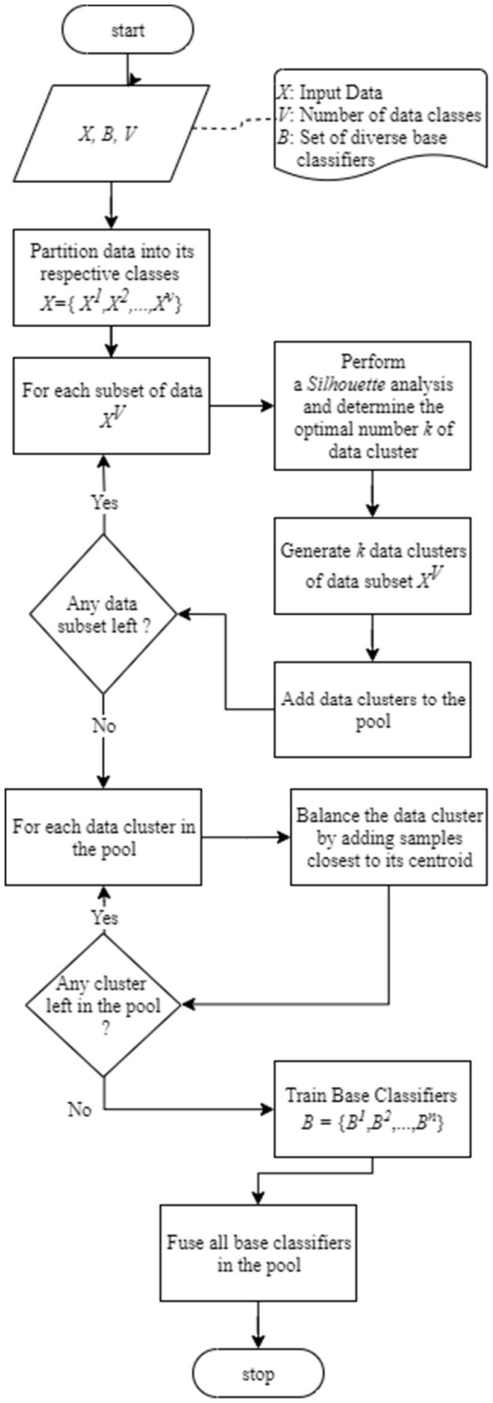

Figure 1: Proposed ensemble classifier generation framework

The generated data clusters of each data class are first balanced before they are utilized for training. The balancing process ensures that any class imbalances that exists in the input data to not lurk into the generated clusters. The balancing of generated data clusters is done by adding data samples from other input classes that are closest to the cluster centroid. For example, if a data cluster C having a centroid c belonging to input data class v has n number of samples in it then it is balanced by adding n samples of each class in V that is not v. This is done by calculating the Euclidean distance of each sample from the cluster centroid c that does not belong to class v as:

$$dist = \left\| x_n^i - c \right\| \quad \forall \; i \in V \; and \; i \notin v \tag{4}$$

The Euclidean distances are stored in a vector $dist$ and the vector is sorted in ascending order. The first n samples from the vector $dist$ are added to the cluster C. The process is repeated for data patterns from each class besides v. At the end of balancing, the cluster C ideally has n number of samples from each data class which have some spatial dependency with each other.

A set of diverse base classifiers $B = \{\zeta^1, \zeta^2, .., \zeta^n\}$ for example Support Vector Machine (SVM), Artificial Neural Network (ANN), Discriminant Analysis (DISCR), Naïve Bayes (NB), K-Nearest Neighbour (KNN), Decision Trees (DT), *etc.* is trained on all balanced data clusters. The type of base classifier chosen is independent of this research and any combination or type of classifier can be chosen. In future we will investigate further what type or number of classifiers can maximise the ensemble accuracy.

A classifier takes in input training data and produces the predicted class labels of the unseen test set T having feature vector x and class labels y. Ensemble classifier is generated by utilizing all classifiers in the pool and conducting a majority vote to fuse their class decisions. This is done as:

$$\xi = \{\zeta^1(x_i^T), \zeta^2(x_i^T), .., \zeta^n(x_i^T)\} \tag{5}$$

where x_i^T is the i^{th} data pattern from the unseen test set X^T and ξ is the ensemble

To get the final predicted output of the ensemble ξ mode is taken as:

$$y' = mode(\xi) \tag{6}$$

where y' is a vector of predicted class labels of the ensemble and $mode$ is a mathematical operator that depicts majority voting here and it simply returns the most frequent value row wise.

Lastly, the ensemble classifier accuracy is computed using the predicted class labels from (6) as follows:

$$acc = \frac{\sum_{x_i \in T} I(y_i', y_i)}{n} \tag{7}$$

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

III. Experiments

This section details the experiments that were conducted to gage the performance of the proposed ensemble classifier. The datasets that are used in the experiments are taken from University of California Irvine (UCI) benchmark machine learning classification datasets repository [28]. A summary of these datasets is given in Table 1.

TABLE 1: UCI BENCHMARK DATASETS USED IN EXPERIMENTATION

Dataset	Number of Samples	Number of Columns/Attributes	Number of Classes
Breast Cancer	699	9	2
Diabetic	768	8	2
Ecoli	336	7	8
Glass	214	10	7
Haberman	306	3	2
Heart-s	270	13	2
Hepatitis	155	19	2
Ionosphere	351	33	2
Iris	150	4	3
Liver	345	6	2
Segment	2310	19	7
Sonar	208	60	2
Spectfheart	267	22	2
Thyroid	7200	21	3
Vehicle	946	18	4
Wine	178	13	3

It can be noted from Table 1, that the datasets chosen have mixed attributes ranging from 150 samples to 7200 allowing for a thorough analysis of the proposed ensemble classifier. Moreover, the same datasets have been used by a number of existing researches [19] allowing for comparative analysis.

The proposed approach is implemented in MATLAB 2019b [29], default implementation of base classifiers SVM, ANN, DISCR, DT, NB, and KNN are used without any parameter optimization. For clustering input data default implementation of K-Means in MATLAB is used. To accommodate for randomness a 10-fold cross validation is conducted and classification accuracy and standard deviation over 10 independent runs is calculated for analysis. For cluster validation default implementation of "*evalclusters"* in MATLAB used with "*Silhouette"* as criterion parameter. The range of clusters for each dataset is measured from 2 to 20, and the optimal number of clusters identified are used as a parameter to K-means.

The results of the proposed ensemble classifiers are first compared with single classifier approaches to evaluate the effectiveness of generating an ensemble of structurally different classifiers, then the results are compared with popular ensemble approaches known as Bagging and Boosting. A set of non-parametric signed ranked tests [30] are adopted with a significance level of 0.05.

A. Experiment results

The average classification accuracies of the proposed ensemble approach on the benchmark datasets over 10 independent runs with 10-fold cross validation is given in Table II.

TABLE II: CLASSIFICAITON ACCURACIES AND AVERAGE CLUSTERS GENERATED PER DATASETS OF THE PROPOSED ENSEMBLE CLASSIFIER

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

It can be noted from Table II that for each dataset a different number of clusters are generated. This is predominantly due to equation (3), as only optimal number of clusters are generated for each dataset. This adds to the fact that each dataset has different characteristics and using the same upper bounds for different datasets is not an ideal strategy.

B. Comparision with single classifiers

The classification performance of the proposed ensemble classifier is also compared with single classifier approaches. For fair comparisons the experiments are run in the same environment with 10-fold cross validation and 10 independent runs. For base classifiers default implementations are used without any parameter optimization. The results are given in Table III with highest classification accuracies given in bold. It can be noted that the proposed ensemble outperformed 6 base classifiers in 10 out of 16 datasets and on average achieved performance gains of approximately 2.0% over SVM, 5.0% over DT, 12.0% over ANN, 2.0% over DISCR, and 6.0% over NB, and KNN. These results are summarized in figure 2.

It can be noted from figure 2, that on average SVM and DISCR are the highest performing classifiers with SVM being slightly better than DISCR achieving a performance of boost of 0.58% over DISCR. Empirically it can be said that SVM is the most robust base classifier compared to others. Although with parameter optimization the results might differ but for default case this holds true.

TABLE III: CLASSIFICAITON ACCURACIES OF THE PROPOSED ENSEMBLE CLASSIFIER IN COMPARISON WITH SINGLE CLASSIFIERS

Dataset	Proposed approach	SVM	DT	ANN	DISCR	NB	KNN
Breast Cancer	**0.9700**	0.9642	0.9434	0.8866	0.9571	0.9648	0.9670
Diabetic	**0.7722**	0.7676	0.7046	0.2239	0.7696	0.7364	0.7147
Ecoli	0.8513	0.7914	0.8116	0.8354	**0.8733**	0.8307	0.8627
Glass	0.9673	**0.9944**	0.9794	0.7561	0.9182	0.9032	0.9757
Haberman	**0.7652**	0.7298	0.6778	0.7337	0.7474	0.5120	0.7125
Heart-s	0.8374	0.8389	0.7533	0.7689	**0.8430**	0.8011	0.6796
Hepatitis	**0.8625**	0.8313	0.8175	0.8113	0.8150	0.8550	0.8113
Ionosphere	**0.9262**	0.8701	0.8855	0.9071	0.8613	0.9071	0.8379
Iris	0.9667	0.9767	0.9427	0.9433	**0.9800**	0.9580	0.9660
Liver	**0.7246**	0.6870	0.6400	0.6832	0.6809	0.6423	0.6649
Segment	0.9525	**0.9626**	0.9548	0.9552	0.9155	0.8977	0.9391
Sonar	**0.7945**	0.7716	0.7082	0.7736	0.7414	0.7608	0.7817
Spectfheart	**0.8023**	0.7741	0.7422	0.0746	0.7520	0.7340	0.7386
Thyroid	0.9113	0.9307	**0.9954**	0.9790	0.9374	0.9400	0.9395
Vehicle	**0.8026**	0.8009	0.7038	0.7869	0.7771	0.6089	0.6414
Wine	**0.9887**	0.9557	0.9099	0.9122	0.9848	0.9770	0.7004

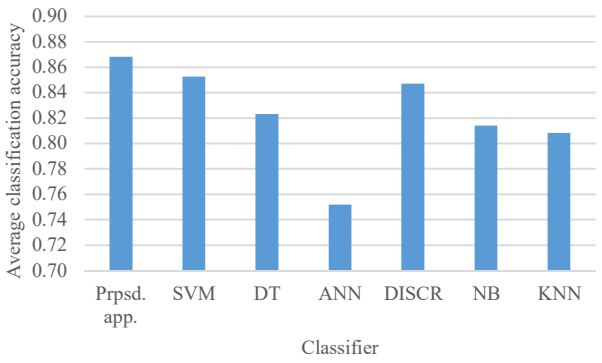

Figure 2: Comparative analysis of the average classification accuracy of the proposed approach and base classifiers

C. *Comparisons with other ensemble approaches*

The classification performance of the proposed ensemble approach is also compared with two state-of-the-art ensemble classifiers namely Random Forest and Adaboost. The default implementation of ensembles was used in MATLAB with the following parameters for Raf:

- *fitcensemble*
- *Method = bag*
- *Learners = tree*

and for Adaboost:

- *fitcensemble*
- *Method = AdaboostM2* (for datasets with more than two classes)
- *Method = AdaboostM1* (for datasets with two classes)

The datasets were partitioned using 10-fold cross validation and classification accuracies over 10 independent runs are reported. The results are given in Table IV, with highest accuracies given in bold.

TABLE IV: CLASSIFICAITON ACCURACIES OF THE PROPOSED ENSEMBLE CLASSIFIER IN COMPARISON WITH STATE-OF-THE-ART ENSEMBLE CALSSIFIERS

Dataset	Proposed approach	ADABOOST	Random Forest
Breast Cancer	**0.9700**	0.9638	0.9621
Diabetic	**0.7722**	0.7301	0.7288
Ecoli	**0.8513**	0.7902	0.8310
Glass	**0.9673**	0.9053	0.9341
Haberman	**0.7652**	0.6676	0.6987
Heart-s	**0.8374**	0.8052	0.7785
Hepatitis	0.8625	**0.8650**	0.8200
Ionosphere	0.9262	**0.9353**	0.9026
Iris	**0.9667**	0.9460	0.9327
Liver	**0.7246**	0.6951	0.6780
Segment	0.9525	**0.9770**	0.9729
Sonar	0.7945	**0.8575**	0.7505
Spectfheart	**0.8023**	0.8015	0.8005
Thyroid	0.9113	**0.9968**	0.9801
Vehicle	**0.8026**	0.7461	0.7334
Wine	**0.9887**	0.9618	0.9533

It can be noted from Table IV, that the proposed ensemble outperformed Adaboost, and RaF in 11 out of 16 datasets and achieved an average of 1.52% performance gains over Adaboost, and 2.71% over RaF.

The proposed ensemble is also compared with various clustering-based ensemble classifiers namely Sacking with Logistic Regression (STLR), Classification by Cluster Analysis (CBCA), standard classification with Clustering (CL), and Stacking with J48 as a combination function (STJ48) proposed in [24]. The classification accuracies are taken directly from the respective papers and are listed in Table V below with highest classification accuracies given in bold. It can be noted that on average the proposed ensemble approach achieved 7.98% performance gains over STJ48 having a significance p value of 0.005 at 95% confidence ($alpha = 0.05$), 5.36% over STLR with a p-value of 0.08, 7.64% over CL with a p-value of 0.005, and 3.68% over CBCA with a p-value of 0.28.

TABLE V: CLASSIFICAITON ACCURACIES OF THE PROPOSED ENSEMBLE CLASSIFIER IN COMPARISON WITH OTHER CLUSTERING BASED STATE-OF-THE-ART ENSEMBLE CALSSIFIERS

Dataset	Proposed approach	STJ48 [24]	STLR [24]	CL [24]	CBCA [24]
Breast Cancer	**0.9700**	0.9480	0.9600	**0.9700**	**0.9700**
Glass	**0.9673**	0.6380	0.6240	0.6480	0.6910
Haberman	**0.7652**	0.7280	0.7380	0.7310	0.7650
Heart-s	0.8374	0.7400	0.8420	0.8230	**0.8450**
Hepatitis	**0.8625**	0.7930	0.8330	0.7870	0.8600
Iris	0.9667	0.9520	0.9450	0.9240	**0.9720**
Segment	0.9525	0.9610	**0.9650**	0.8530	0.9490
Sonar	0.7945	0.7560	**0.8490**	0.7560	0.7900
Spectfheart	**0.8023**	0.6800	0.6760	0.7350	0.7410

IV. Conclusion

In this paper a novel ensemble classifier approach was proposed. The proposed approach utilized clustering to generate a pool of data clusters. This is done to "perturb" the input data and generate a diverse input space using which a pool of diverse base classifiers is trained. Dataset is first partitioned based on the number of classes and optimal number of data clusters are generated for each subset. The optimal number of data clusters are determined by conducting a Silhouette analysis. All generated clusters are then balanced by adding samples from other classes that are closest to the cluster centroids. In this manner we generate an augmented and perturbed input space which not only alleviates the class imbalance problem but also exploits any spatial dependencies that exist in the data.

From experiments, it was evident that each dataset has different characteristic which in turn required a different number of optimal data clusters to be generated to achieve the highest classification accuracy. It can be noted from figure 3 that a fixed value of K is not an effective strategy as some datasets are sparse such as *Thyroid* with samples spread from each other whereas others are dense such as *Diabetic* in which samples are close to each other. The proposed ensemble approach not only performed better than other single classifier approaches but also performed well in comparison to other state-of-the-art ensemble classifier approaches as evident by results in Table IV and Table V.

In future we will run further experiments on more large scale real-world and benchmark datasets to further validated the efficacy of the proposed approach. We will also test with other cluster validation techniques and investigate the effect of different validation techniques on the classification accuracy of the ensemble.

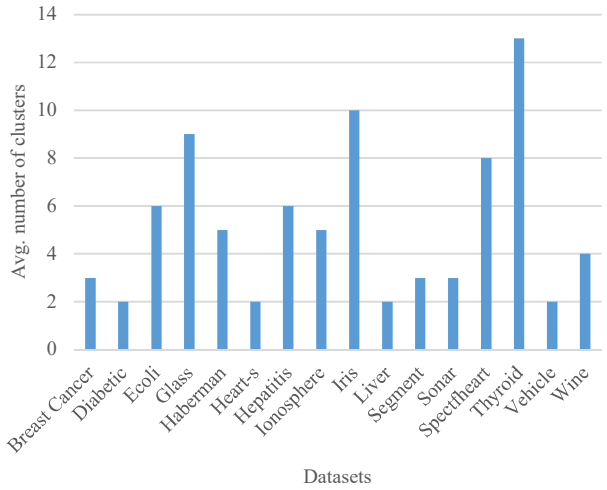

Figure 3: Average number of data clusters generated per dataset

Acknowledgment

This research was supported under the Australian Research Council's Discovery Project funding scheme (Project Number DP160102369).

References

[1] Y. Ren, L. Zhang, and P. N. Suganthan, "Ensemble classification and regression-recent developments, applications and future directions," *IEEE Computational Intelligence Magazine,* vol. 11, no. 1, pp. 41-53, 2016.

[2] A. Danesh, B. Moshiri, and O. Fatemi, "Improve text classification accuracy based on classifier fusion methods," in *International Conference on Information Fusion*, 2007, pp. 1-6.

[3] Z. M. Jan and B. Verma, "Ensemble classifier optimization by reducing input features and base classifiers," in *Proceedings of the Congress on Evolutionary Computation*, 2019, pp. 1580-1587.

[4] 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.

[5] 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.

[6] Z. Yu *et al.*, "Hybrid incremental ensemble learning for noisy real-world data classification," *IEEE Transactions on Cybernetics,* vol. 49, no. 2, pp. 403-416, 2019.

[7] I. Barandiaran, "The random subspace method for constructing decision forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 20, no. 8, 1998.

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

[9] R. Bryll, R. Gutierrez-Osuna, and F. Quek, "Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets," *Pattern Recognition,* vol. 36, no. 6, pp. 1291-1302, 2003.

[10] G. Brown, J. L. Wyatt, and P. Tino, "Managing diversity in regression ensembles," *Journal of Machine Learning Research,* vol. 6, no. 9, pp. 1621-1650, 2005.

[11] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "RUSBoost: A hybrid approach to alleviating class imbalance," *IEEE Transactions on Systems Man and Cybernetics Part a-Systems and Humans,* vol. 40, no. 1, pp. 185-197, 2010.

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

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

[14] G. Ratsch, T. Onoda, and K. R. Muller, "Soft margins for adaboost," *Machine Learning,* vol. 42, no. 3, pp. 287-320, 2001.

[15] C. Domingo and O. Watanabe, "MadaBoost: a modification of adaboost," in *Conference on Computational Learning Theory*, 2000, pp. 180-189.

[16] L. Zhang and P. N. Suganthan, "Random forests with ensemble of feature spaces," *Pattern Recognition,* vol. 47, no. 10, pp. 3429-3437, 2014.

[17] 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. 2165-76, 2015.

[18] A. Nemirovskii, "Several np-hard problems arising in robust stability analysis," *Mathematics of Control, Signals and Systems,* vol. 6, no. 2, pp. 99-105, 1993.

[19] 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.

[20] Z. M. Jan, B. Verma, and S. Fletcher, "Optimizing clustering to promote data diversity when generating an ensemble classifier," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 2018, pp. 1402-1409.

[21] Y. Yang and J. Jiang, "Hybrid sampling-based clustering ensemble with global and local constitutions," *IEEE Transactions on Neural Networks and Learning Systems,* vol. 27, no. 5, pp. 952-65, 2016.

[22] Z. Yu, L. Li, J. Liu, and G. Han, "Hybrid adaptive classifier ensemble," *IEEE Transactions on Cybernetics,* vol. 45, no. 2, pp. 177-90, 2015.

[23] 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, 2015.

[24] 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.

[25] 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.

[26] 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.

[27] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics,* vol. 20, pp. 53-65, 1987.

[28] K. Bache and M. Lichman. "UCI machine learning repository." http://archive.ics.uci.edu/ml/ (accessed on November 2019).

[29] MATLAB, *Statistics and machine learning toolbox*. Natick, Massachusetts: The MathWorks Inc., 2013.

[30] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics bulletin,* vol. 1, no. 6, pp. 80-83, 1945.