A Novel Layered Clustering based Approach for Generating Ensemble of Classifiers

Ashfaqur Rahman and Brijesh Verma

***Abstract*— This paper introduces a novel concept for creating an ensemble of classifiers. The concept is based on generating ensemble of classifiers through clustering of data at multiple layers. The ensemble classifier model generates a set of alternative clustering of a data set at different layers by randomly initializing the clustering parameters and trains a set of base classifiers on the patterns at different clusters in different layers. A test pattern is classified by first finding the appropriate cluster at each layer and then using the corresponding base classifier. The decisions obtained at different layers are fused into a final verdict using majority voting. As the base classifiers are trained on overlapping patterns at different layers, the proposed approach achieves diversity among the individual classifiers. Identification of difficult–to–classify patterns through clustering and achievement of diversity through layering leads to better classification results as evidenced from the experimental results.**

***Index Terms*—ensemble classifiers, committee of experts, multiple classifier systems, cluster oriented ensemble classifier**

I. INTRODUCTION

IN an ensemble classifier approach, multiple base classifiers learn decision boundaries on the training patterns and their decisions on a test pattern are fused to reach the final classification verdict. Ensemble classifiers are also known as committee of classifiers, mixture of experts and multiple classifier systems. Many ensemble classifier generation methods are presented in the literature. The generation methods aim to produce the base classifiers in a way that they differ from each other in terms of the errors they make on identical patterns. This phenomenon is also known as *diversity* [1]–[5]. The fusion methods on the other hand explore ways to merge the decisions from the base classifiers into a final verdict.

Ensemble of classifiers can be generated using *clustering*. The data can be partitioned into multiple non–overlapping segments using clustering. The segments may contain overlapping patterns from multiple classes and these patterns are difficult to classify otherwise. A classifier like a neural network can be trained on the patterns within a cluster at this stage to learn boundaries among the patterns. A number of research efforts are observed in this direction in the literature [19]–[26] and we use the term *clustered ensemble* to refer to them. Disjoint segments of difficult–to–classify patterns are identified in this process and different base classifiers are well trained on different segments. Each pattern in this process, however, is involved in the training of only one classifier and the decision on a test pattern is dictated by only one base classifier. The objective of obtaining multiple decisions on a test pattern does not happen with clustered ensembles. In order to achieve diversity it is required that the training sets for the different classifiers are different and at the same time there is overlapping between the training sets so that identical patterns can be learned by multiple classifiers.

In this regard we make use of the fact that the final content of the segments in some clustering algorithms depends on the initialization of clustering parameters (e.g. seeds in *k*–means clustering algorithm). In order to achieve diversity the data set can be independently partitioned *n* times using different initial clustering parameters and identical patterns will belong to *n* alternate clusters. We use the terminology *n layers* to refer to *n* alternative clusterings of the data set in this paper. The decision provided by the base classifiers trained on the *n* alternate clusters at *n* layers can be fused to obtain the final verdict on the pattern. With clustering we can generate the base classifiers and with layers we can achieve the diversity. Based on the above philosophy, in this paper we present a novel approach towards generating ensemble of classifiers using *layered clustering*.

The research presented in this paper aims to: (i) develop a novel method for generating ensemble of classifiers using cluster layers, (ii) investigate the impact of number of layers on classification accuracy, (iii) investigate the impact of number of clusters at different number of layers on classification accuracy, and (iv) obtain a comparative analysis on how well the proposed approach performs compared to the commonly used approaches for ensemble classifier generation.

The paper is organized as follows. Section II reviews existing approaches for ensemble classifier generation and decision fusion. Section III presents the proposed approach for generating ensemble classifiers. The experimental platform is presented in Section IV. Section V presents the experimental results and discussion. Finally, Section VI concludes the paper.

II. RELATED WORKS

Two major streams of works are observed in the literature

Manuscript received May 6, 2010.

Ashfaqur Rahman is with the CQUniversity, Australia (phone: +61749306508; e-mail: a.rahman@cqu.edu.au).

Brijesh Verma is with the CQUniversity, Australia (phone: +61749309058; e-mail: b.verma@cqu.edu.au).

towards ensemble classifiers: construction of base classifiers and fusion methods for combining the decisions of the base classifiers. The fusion methods map the base classifier outputs into class decisions. The mapping can be done on discrete class decisions or continuous class confidence values produced by the base classifiers. The commonly used fusion methods [3] for combining class labels are majority voting, weighted majority voting, behaviour knowledge space, and Borda count. The fusion methods for combining continuous outputs include algebraic combiners including mean rule, weighted average, trimmed mean, min/max/median rule, product rule, and generalized mean. The base classifiers in the proposed approach produce discrete valued class decisions and we use majority voting based fusion method. As this paper presents an ensemble classifier construction method, we refrain from discussing the fusion methods further. Some recent papers on fusion methods are available in [6]–[9].

Ensemble classifiers can be constructed by using identical (e.g. using neural network only) as well as different base classifiers (e.g. using neural network, SVM and *k*–NN classifier) although the former is found more in practice. Ensemble classifier generation methods using identical classifiers can be broadly classified into four groups that are based on (i) manipulation of the training parameters, (ii) manipulation of the feature space, (iii) manipulation of the training data labels, and (iv) manipulation of the training examples.

Ensemble classifiers can be created by *manipulating the training parameters* of the base classifiers. The authors in [11] propose a neural ensemble classifier where different network weights are used to initialise the base neural network learning process in order to diversify the base classifiers. The paper presented an approach to initialise neural networks that uses competitive learning to intelligently create networks that are originally located far from the origin of weight space. In [12] a set of neural networks with different initial weights were trained to classify land surface images obtained from the sensors housed in satellites setup by NASA. These methods are shown to achieve better generalisation.

The second group of ensemble classifier generation methods that we discuss here generates base classifiers by *manipulating the input feature space* [10]. Thirty two neural networks were trained in [13] based on eight different subsets of 119 available input features to identify volcanoes. The resulting ensemble classifier was able to match the performance of human experts. Multiple decision trees were constructed systematically in [14] by pseudorandomly selecting subsets of components of the feature vector. Random subspace ensembles of SVMs were used in [15] for Bio–molecular cancer classification and in [16] for classification of brain images obtained through functional magnetic resonance imaging (FMRI). As shown in [7] these *random subspace* ensemble classifiers perform relatively inferior to other ensemble classifiers.

The third group of ensemble classifiers is constructed by *manipulation of the output targets*. In class switching ensemble [17] each base classifier is generated by switching the class labels of a fraction of training examples that are selected at random from the original training set. In Error Correcting Output Coding (ECOC) method [18] the learning problem is constructed by randomly partitioning the K classes into two subsets A_l and B_l and the input data is then relabeled such that the original classes in A_l and B_l are given new labels 0 and 1 respectively. A classifier h_l learns the relabeled data and this process is repeated L times to obtain the classifiers $h_1,\dots,h_L$. Given a new pattern $\boldsymbol{x}$ each classifier h_l predicts either 0 or 1. If $h_l(\boldsymbol{x}) = 0$, then each class in A_l receives a vote. Otherwise each class in B_l receives a vote. On reception of votes from all the classifiers, the class with highest number of votes is selected as the prediction of the ensemble classifier.

The fourth group of methods generates ensemble classifiers by *manipulating the training examples*. The base classifiers are trained on different subsets of the training examples. The methods differ in generation of the subsets and can be broadly classified into three subgroups:

- *Clustered Ensembles* [19]–[24][47][48] where the subsets are generated by partitioning the training examples into non–overlapping clusters. The method identifies the difficult–to–classify patterns that tend to stay close in Euclidean space in a cluster and aims to build specialized base classifiers on each cluster. A pattern can belong to one cluster only and decision on a test pattern is governed by only one base classifier. A selection rather than fusion approach is followed for obtaining the ensemble classifier decision. The concept of diversity thus does not apply to these ensembles and clustered ensembles rely on the performance of each base classifier. These methods were actually designed to reduce the complexity of learning large data sets [19]. Some ensemble classifiers use soft clustering. The hierarchical mixture of experts method [27] divides the input space into nested regions and fits simple surfaces to the data that fall into these regions. The regions have *soft* boundaries, meaning that data points may lie simultaneously in multiple regions. The boundaries between regions are themselves simple parameterized surfaces that are adjusted by the learning algorithm. A tree with *expert networks* at the leaves and *gating networks* at the non–terminals forms the architecture of the hierarchical mixture of experts method. The gating networks provide the soft partitioning and the expert networks provide local regression surfaces within the partition.
- Bootstrap aggregating or *bagging* [28] is one of the earliest ensemble classifier generation methods. Diversity in bagging is achieved by training the base classifiers on different subsets of the training data. The subsets are randomly drawn (with replacement) from the training set. The base classifiers are homogeneous in nature. The decisions of the individual classifiers are fused using majority voting i.e. the class chosen by most base classifiers is the final verdict of the ensemble classifier. Bagging is suitable for small data sets. For large data sets however the sampling scheme based on the bootstrap with replicates of the training set is infeasible. Bagging provides a mechanism to achieve diversity but does not mention any

mechanism to identify difficult–to–classify patterns that leaves space for improvement. There are a number of variants of bagging and aggregation approaches including random forests [29],large scale bagging [30], ordered aggregation [31], adaptive generation and aggregation approach [32], and fuzzy bagging [32][34].

- *Boosting* [35][36] creates data subsets for base classifier training by re-sampling the training examples and providing the most informative training example for each consecutive base classifier. Each of the training examples is assigned a weight that determines how well the instance was classified in the previous iteration. The patterns in the current subset of the training data that are badly classified are included in the training subset for the next iteration. This way the different base classifier errors are made uncorrelated. The subsets in boosting not necessarily contain examples that are difficult to classify when combined together. AdaBoost [37] is a more generalized version of boosting. A number of variants of boosting can be observed in the literature including weighted instance selection method [38], boosting recombined weak classifiers [39], Learn++ [40] and its variant Learn++.NC [41].

The proposed ensemble classifier generation method belongs to the last group and a careful scrutiny of these existing works reveals that (i) cluster ensembles identify overlapping patterns that are difficult to classify but do not provide any mechanism to incorporate diversity and (ii) bagging and boosting provide a mechanism to achieve diversity but do not set a direction for identifying clusters of patterns that deserve more attention than others. We are motivated to develop a layered cluster oriented approach for generating ensemble of classifiers that can identify clusters of difficult–to–classify patterns for improving accuracy and train base classifiers at different layers on overlapping clusters to improve diversity. The following sections detail the proposed ensemble classifier.

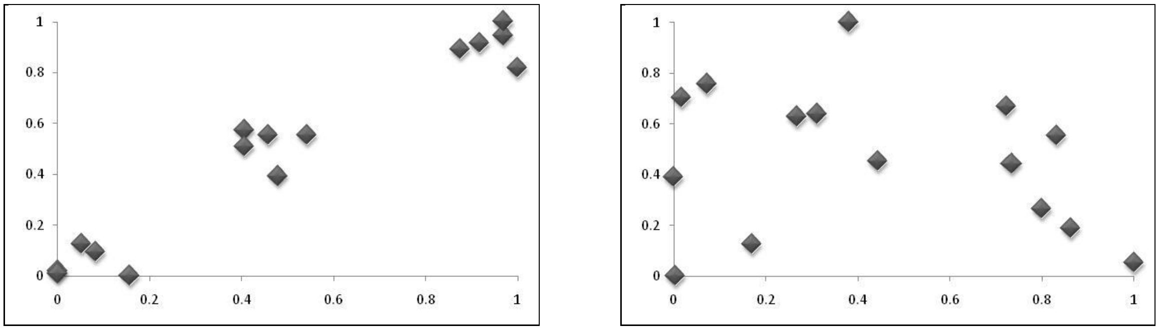
III. PROPOSED APPROACH

A. Philosophy

The proposed approach for generating ensemble of classifiers is based on the concept of clustering. The primary task is to cluster the data set into multiple segments and engage a set of base classifiers to learn the decision boundaries among the patterns within each cluster. The process of clustering partitions a data set into segments that contains highly correlated data points. These correlated data points tend to stay very close geometrically. They are difficult to classify especially when patterns from multiple classes overlap within a cluster. When clustering is applied on labelled data sets (i.e. data where each pattern is associated with a class), the produced segments can be of two types – *atomic* and *non–atomic*. An atomic cluster contains patterns that belong to the same class whereas a non–atomic cluster is composed of patterns from multiple classes.

At the end of the clustering process, classifiers can be trained on the patterns of non–atomic clusters whereas the class label can be memorized for the atomic clusters. The class of a test pattern can be predicted by first finding the appropriate cluster based on its distance from the cluster centres and then using the corresponding classifier (for a non–atomic cluster) or the class label (for an atomic cluster). A pattern can belong to one cluster only and the decision on a test example is based on the prediction of a single classifier. Clustering identifies difficult–to–classify patterns but the decision making process based on a single classifier prediction leaves space for improvement.

The final composition of the partitions in some clustering algorithms depends on the initialization of clustering parameters. For example, the final contents of clusters in k–means clustering algorithm depend on the initialization of the seeds (i.e. the initial state of the cluster centres) when k is not equal to the actual number of clusters. We can clarify this with an example. Consider two artificial data sets with two attributes as shown in Fig 1. Data set D_1 is well clustered into three partitions whereas the number of actual clusters in D_2 is more than three. Now let's consider applying k–means clustering algorithm on these data sets with $k = 3$ with the cluster centres initialized randomly. The three independent outcomes of the clustering algorithm on these data sets are presented in Table I. Note that the patterns always belong to the same cluster for the data set D_1. Patterns in D_2 belong to different clusters as the k–means clustering algorithm is applied for a number of times. Identical situation will occur for data set D_1 if k–means clustering algorithm is applied with $k \neq 3$. Note that the change is at its minimum when k is equal to the actual number of clusters.



(a) Data set D_1

(b) Data set D_2

Fig 1: Two artificial data sets. D_1 is well clustered into three clusters whereas D_2 is not.

Table I: Application of *k*–means clustering algorithms on artificial data sets D_1 and D_2 with *k*=3

Data Set D1		*k*–means clustering			Data Set D2		*k*–means clustering		
x	*y*	*Run* 1	*Run* 2	*Run* 3	*x*	*y*	*Run* 1	*Run* 2	*Run* 3
0.91667	0.91489	1	1	1	0.73413	0.44099	3	2	1
0.875	0.89362	1	1	1	0	0.38846	1	3	2
1	0.81915	1	1	1	0.26687	0.6272	1	3	3
0.96875	0.94681	1	1	1	0.015612	0.70219	1	3	3
0.96875	1	1	1	1	0.071102	0.75611	1	3	3
0.40625	0.57447	2	2	2	0.86198	0.18678	2	2	1
0.54167	0.55319	2	2	2	0.72191	0.66691	3	1	1
0.40625	0.51064	2	2	2	0.31062	0.63765	1	3	3
0.45833	0.55319	2	2	2	1	0.051871	2	2	1
0.47917	0.39362	2	2	2	0.002845	0	1	3	2
0	0.021277	3	3	3	0.44307	0.45122	3	3	3
0.15625	0	3	3	3	0.38081	1	3	1	3
0.083333	0.095745	3	3	3	0.79889	0.26333	2	2	1
0	0.010638	3	3	3	0.8312	0.55459	3	2	1
0.052083	0.12766	3	3	3	0.16881	0.12466	1	3	2

We aim to incorporate the above observation to improve the decision making process in ensemble of classifiers. The idea is to train multiple classifiers on similar patterns. At this point let's introduce the concept of layers. A layer indicates the partitioning of the data set based on one set of seed parameters. Fig 2 demonstrates an example of a data set (Fig 2(a)) divided into three clusters. Fig 2(b) presents layer one clustering where the data set in Fig 2(a) is divided into three clusters based on the initial values of the clustering parameters Υ_1. The clusters are indexed by the layer number followed by the cluster number. For example the second cluster at layer one is represented by $\Omega_{1,2}$. An alternate clustering of the same data set into three segments is presented in Fig 2(c) based on another set of initial clustering parameters Υ_2. Note that patterns belong to different clusters at different layers (Fig 2(d)).

Base classifiers are now trained on the non–atomic clusters at different layers. The clusters at different layers overlap and the same pattern is included in the training of multiple classifiers at different layers. Different subsets of the data are used in the training of the base classifiers and thus diversity is achieved. A test pattern belongs to different clusters at different layers and thus gets decisions from different base classifiers that can be combined to obtain the ensemble classification verdict. We use this idea for generating the ensemble of classifiers. The *novelty* of the proposed approach lies in the use of a layered clustering approach towards achieving diversity among the base classifiers. Note that the proposed approach is significantly different from the clustered ensemble that fails to achieve diversity as a single base classifier always provides the decision on a pattern.

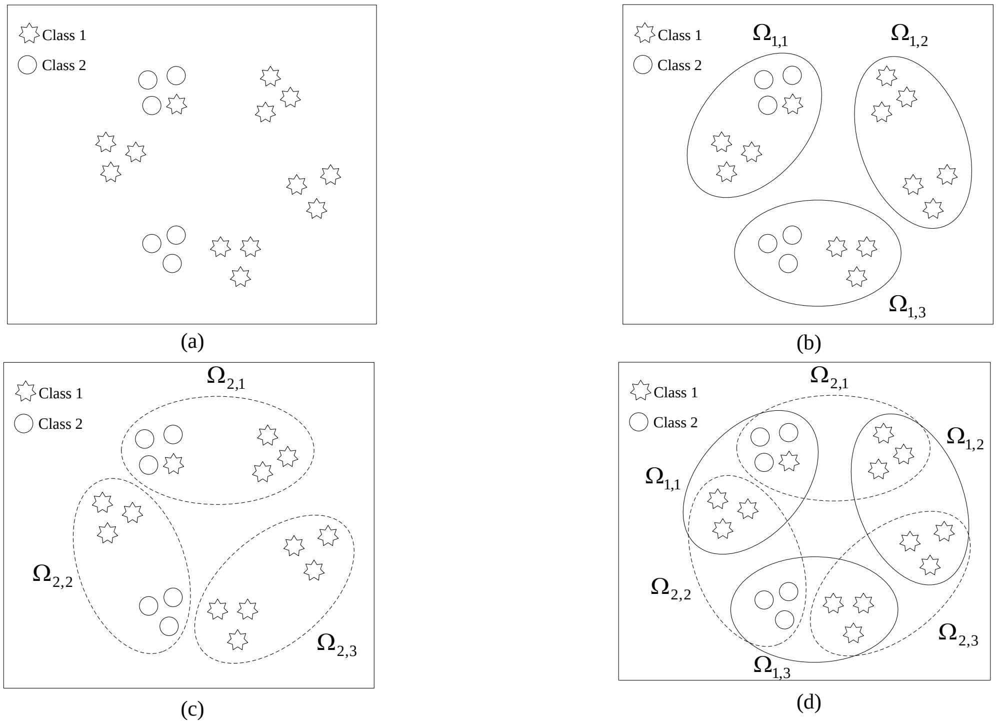


Fig 2: Clustering labelled data: (a) Data set with patterns from two classes, (b) Layer one partitioning – Data set partitioned into three clusters where $\Omega_{1,2}$ is an atomic cluster whereas the other two are non–atomic clusters, (c) Layer two partitioning – Data set partitioned into three clusters where $\Omega_{2,3}$ is an atomic cluster whereas the other two are non–atomic clusters, (d) Layer one and layer two clusters superimposed indicating patterns belonging to alternate clusters.

B. Ensemble Classifier Model

Let the training patterns in the data set be represented by $\Gamma = \{(\boldsymbol{x}_1, t_1), (\boldsymbol{x}_2, t_2), \dots, (\boldsymbol{x}_{|\Gamma|}, t_{|\Gamma|})\}$ where each pattern is described by a vector of n continuous valued features $\boldsymbol{x}_j =< x_{j1}, x_{j2}, \dots, x_{jn} >$ and a class label t_j with $t_j \in \{class_1, class_2, \dots, class_{N_{class}}\}$. A layer is denoted by l and the K clusters at layer l are denoted by $\Omega_{l,1}, \Omega_{l,2}, \dots, \Omega_{l,K}$ where $1 \le l \le N_{layers}$.

A pattern in the training set can be considered as a point in the Euclidean space of dimension n. The objective of the clustering algorithm is to group data points that are geometrically close. Given two patterns $(\boldsymbol{x}_i, t_i)$ and $(\boldsymbol{x}_j, t_j)$ in the training set a distance function d between them is defined in terms of their Euclidean distance as

$$d(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sqrt{\textstyle\sum_{k=1}^{n} (x_{ik} - x_{jk})^2}, \qquad (1)$$

where $\boldsymbol{x}_i =< x_{i1}, x_{i2}, \dots, x_{in} >$ and $\boldsymbol{x}_j =< x_{j1}, x_{j2}, \dots, x_{jn} >$. Assuming a set of K clusters $\{\Omega_{l,1}, \Omega_{l,2}, \dots, \Omega_{l,K}\}$ at layer l, the associated cluster centres $\Upsilon_l = \{\boldsymbol{\kappa}_{l,1}, \boldsymbol{\kappa}_{l,2}, \dots, \boldsymbol{\kappa}_{l,K}\}$ are initialized randomly and the clustering algorithm aims to minimize an objective function

$$J_{L_i} = \textstyle\sum_{k=1}^{K} \sum_{\forall x_j \in \Omega_{i,k}} d(\boldsymbol{x}_j, \boldsymbol{\kappa}_{i,k}), \qquad (2)$$

for all the data points in the training set Γ.

At the end of the clustering process at layer l each pattern $(\boldsymbol{x}_i, t_i)$ belongs to a cluster $\Omega_{l,k}$ where $1 \le k \le K$. At this point the clusters are separated into atomic and non–atomic clusters. A class distribution vector is defined for each cluster $\Omega_{l,k}$ in this regard as –

$$\Phi_{\Omega_{l,k}}(c_i) = \textstyle\sum_{\forall (x_j, t_j) \in \Omega_{l,k}} \phi(t_j, c_j), \text{ where} \qquad (3)$$

$$\phi(t_j, c_i) = \begin{cases} 1 & if\ t_j = c_i \\ 0 & otherwise \end{cases} \quad \text{and} \qquad (4)$$

$c_i \in \{class_1, class_2, \dots, class_{N_{class}}\}$. A cluster $\Omega_{l,k}$ is defined atomic *if*

$$\frac{\max(\Phi_{\Omega_{l,k}})}{\sum_{\forall c_i} \Phi_{\Omega_{l,k}}(c_i)} = 1. \qquad (5)$$

A cluster not satisfying (5) indicates the presence of patterns from multiple classes and is declared non–atomic. A class label is memorized for an atomic cluster $\Omega_{l,k}$ as

$$\beta_{l,k} = \text{argmax}_{c_i} \Phi_{\Omega_{l,k}}(c_i) \qquad (6)$$

where $c_i \in \{class_1, class_2, \dots, class_{N_{class}}\}$.

A neural network $\theta_{l,k}$ is trained at this stage on the patterns in each non–atomic cluster $\Omega_{l,k}$ to learn the decision boundaries. A test pattern $\boldsymbol{x}$ is classified by first finding the appropriate cluster at each layer. For this purpose, the distance between $\boldsymbol{x}$ and the centre of each cluster $\boldsymbol{\kappa}_{l,k}$ is computed using (1) and the appropriate cluster at layer l is selected as

$$\widehat{\Omega}_{l,k} = \text{argmin}_{\boldsymbol{\kappa}_{l,k}} d(\boldsymbol{x}, \boldsymbol{\kappa}_{l,k}). \qquad (7)$$

If $\widehat{\Omega}_{l,k}$ is an atomic cluster the class label $\beta_{l,k}$ learned using (6) is predicted at layer l. If $\widehat{\Omega}_{l,k}$ is a non–atomic cluster the corresponding neural network $\theta_{l,k}$ trained on $\widehat{\Omega}_{l,k}$ is used to predict the class label $\beta_{l,k}$ at layer l. Upon receiving the prediction set $\{\beta_{l,k}\}$ from all the N_{layers} layers the decisions are fused into a final verdict using the majority voting fusion rule.

The learning and prediction phase of the proposed approach based on the above philosophy is presented in Fig 3 and Fig 4 respectively. The training data set is clustered in N separate layers. At each layer the data is segmented into K clusters based on clustering parameters (e.g. initial state of the cluster centres). A cluster analyser then identifies atomic and non–atomic clusters. The class label is recorded for atomic clusters. A neural network is trained on the patterns of a non–atomic cluster. During prediction the appropriate cluster for the test pattern is identified at each layer. If the selected cluster is atomic the pre–recorded class is predicted as $\beta_{l,k}$. If the cluster is non–atomic the corresponding neural network predicts the class $\beta_{l,k}$. Once the prediction is received from all the N layers the final verdict is obtained from $\{\beta_{l,k}\}$ using the majority voting.

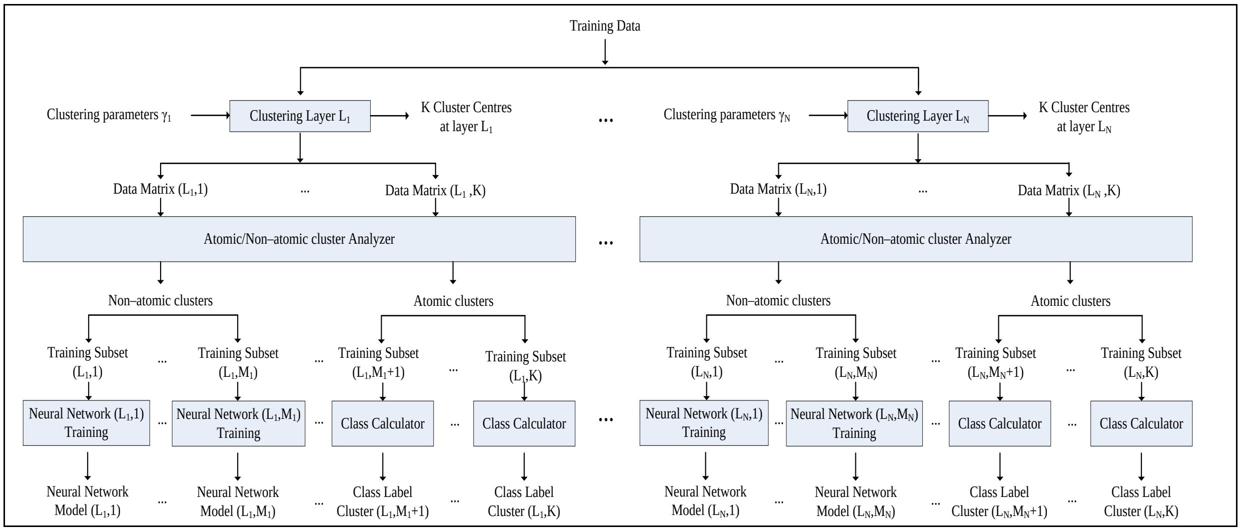

Training Data
Clustering parameters γ_1
Clustering Layer L_1
K Cluster Centres at layer L_1
Clustering parameters γ_N
Clustering Layer L_N
K Cluster Centres at layer L_N
Data Matrix (L_1,1)
Data Matrix (L_1 ,K)
Data Matrix (L_N,1)
Data Matrix (L_N ,K)
Atomic/Non–atomic cluster Analyzer
Atomic/Non–atomic cluster Analyzer
Non–atomic clusters
Atomic clusters
Non–atomic clusters
Atomic clusters
Training Subset (L_1,1)
Training Subset (L_1,M_1)
Training Subset (L_1,M_1+1)
Training Subset (L_1,K)
Training Subset (L_N,1)
Training Subset (L_N,M_N)
Training Subset (L_N,M_N+1)
Training Subset (L_N,K)
Neural Network (L_1,1) Training
Neural Network (L_1,M_1) Training
Class Calculator
Class Calculator
Neural Network (L_N,1) Training
Neural Network (L_N,M_N) Training
Class Calculator
Class Calculator
Neural Network Model (L_1,1)
Neural Network Model (L_1,M_1)
Class Label Cluster (L_1,M_1+1)
Class Label Cluster (L_1,K)
Neural Network Model (L_N,1)
Neural Network Model (L_N,M_N)
Class Label Cluster (L_N,M_N+1)
Class Label Cluster (L_N,K)

Fig 3: Training method of the proposed ensemble classifier with L_N cluster layers.

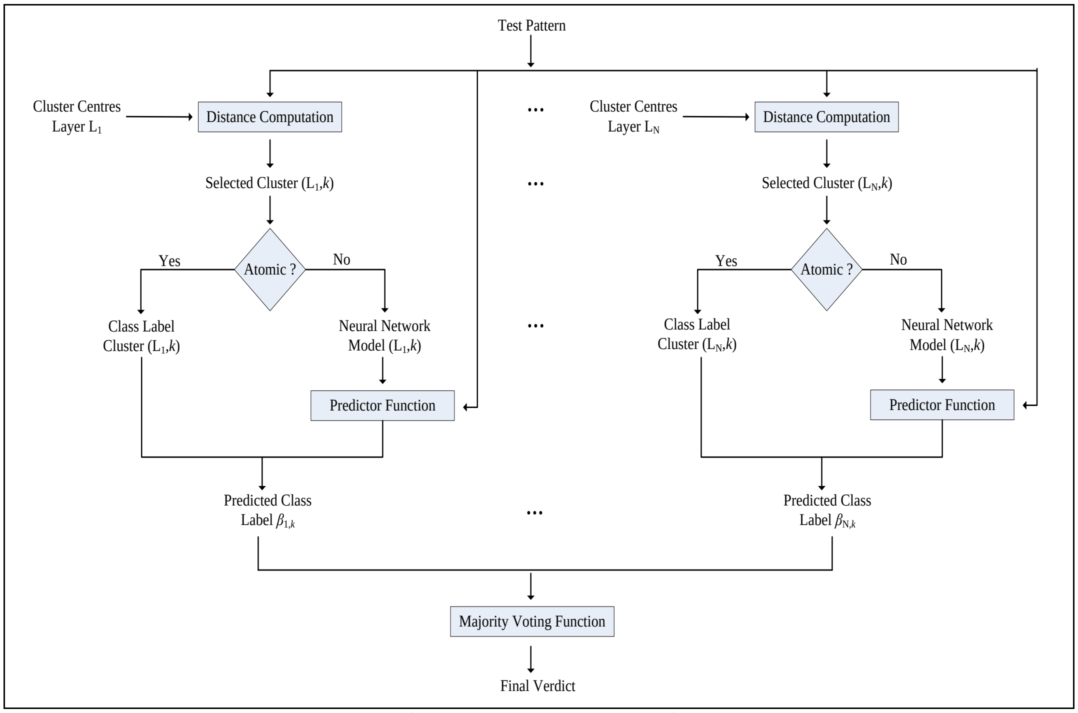


Fig 4: Prediction method of the proposed ensemble classifier with L_N clusters layers.

IV. EXPERIMENTAL PLATFORM

We have conducted a number of experiments on benchmark data sets to evaluate the strength of the proposed ensemble classifier. We have compiled the datasets as used in contemporary research works [7][31][39] from the UCI Machine Learning Repository [42]. A summary of the data sets is presented in Table II. We used 10–fold cross validation approach for reporting the results for all the data sets. We used the *k*–means clustering algorithm for partitioning the data sets.

A neural network (MLP) with two hidden layers and tan sigmoid activation function was used in the experiment. In the first layer five hidden units were used and in the second layer the number of hidden units is set equal to the number of classes. Training of the weights was achieved using backpropagation learning algorithm. The following parameter setting was used during the training process for all data sets – (a) *Learning rate* = 0.01, (b) *Momentum* = 0.4, (c) Epochs i.e. *# of iterations* = 25, and (d) *RMS goal* = 0.00001. Note that the main objective of the experiment was to find the impact of layered clustering and we thus restrict ourselves to best parameter settings on data sets found by trial–and–error.

Table II: Data sets used in the experiments.

Dataset	**# instances**	**# attributes**	**# classes**	**Test process**
Breast Cancer	699	9	2	10–fold cv
Diabetes	768	8	2	10–fold cv
Ecoli	336	7	8	10–fold cv
German	1000	20	2	10–fold cv
Glass	214	10	7	10–fold cv
Ionosphere	351	33	2	10–fold cv
Iris	150	4	3	10–fold cv
Liver	345	6	2	10–fold cv
Parkinsons	197	23	2	10–fold cv
Pendigits	10992	16	10	10–fold cv
Satellite	6435	36	6	10–fold cv
Segment	2310	19	7	10–fold cv
Sonar	208	60	2	10–fold cv
Spam	4601	57	2	10–fold cv
Spect	267	23	2	10–fold cv
Thyroid	215	5	3	10–fold cv
Transfusion	748	5	2	10–fold cv
Vehicle	946	18	4	10–fold cv
Vowel	528	13	11	10–fold cv
Wine	178	13	3	10–fold cv

We used majority voting for decision fusion. Experiments are conducted by partitioning data sets in one to ten layers. At each layer, data is partitioned into one to ten clusters to observe the impact of clustering on classification accuracy. We have computed diversity of the proposed ensemble classifier using Kohavi–Wolpert (KW) variance [43]. Given a set of $|\Gamma|$ examples $\{(\boldsymbol{x}_1, t_1), (\boldsymbol{x}_2, t_2), \dots, (\boldsymbol{x}_{|\Gamma|}, t_{|\Gamma|})\}$, KW variance for each layer l is computed as

$$KW = \frac{1}{|\Gamma| \times L} \sum_{j=1}^{|\Gamma|} D_l(\boldsymbol{x}_j) \times (L - D_l(\boldsymbol{x}_j)) \tag{8}$$

where L is the number of layers, and D_l is set as

$$D_l(\boldsymbol{x}_j) = \begin{cases} 1 & if\ \boldsymbol{x}_j\ is\ classified\ correctly \\ 0 & if\ \boldsymbol{x}_j\ is\ classified\ incorrectly \end{cases} \tag{9}$$

The classification results of clustered ensemble [19], bagging [28], and boosting (AdaBoostM1) are compared with the proposed approach. The results of bagging and boosting are obtained using WEKA [44] with neural network (Multi Layer Perceptron) as the base classifier. A total of ten MLPs were used for both bagging and boosting. The results on the proposed ensemble classifier and clustered ensemble [19] were obtained using MATLAB 7.5.0. The data set is partitioned into one to ten clusters and the best performing number of clusters on the training set is used in the clustered ensemble method. The same implementation of backpropagation learning algorithm was used in both MATLAB and WEKA. The same partitions (i.e. folds) of the data were used in the MATLAB and WEKA executions.

V. RESULTS AND DISCUSSION

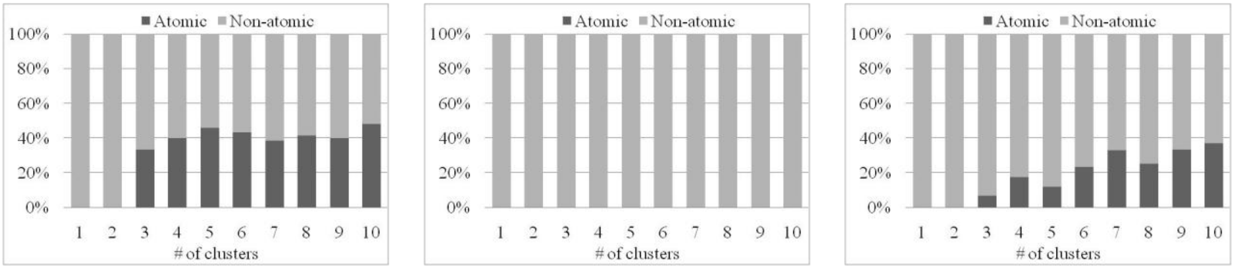
The discussions with graphical representations in Section A, Section B, and Section C are confined to a subset of data sets in Table II. A discussion on accuracies and comparative analysis on all the data sets is presented in Section D.

A. Impact of Clustering on Ensemble Classifier Learning

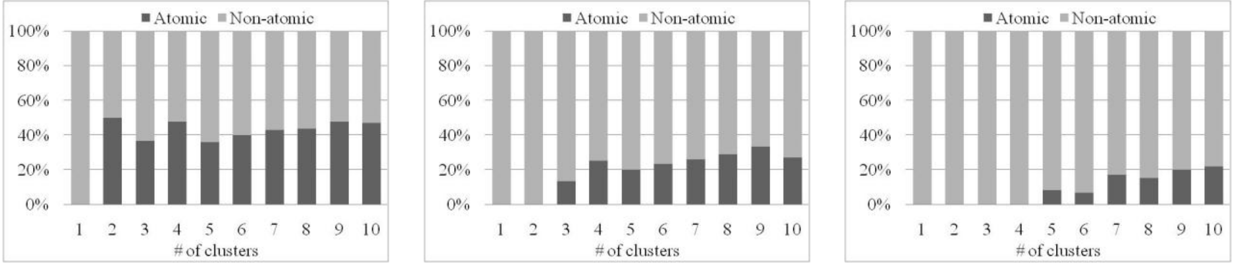
Fig 5 represents the content of each cluster as the datasets are partitioned into one to ten clusters. No atomic clusters are produced for *Diabetes* and *Vowel* data set. This indicates the presence of strong overlapping clusters in these data sets. Relatively higher number of atomic clusters is observed for *Breast Cancer*, *Parkinsons* and *Wine* data set. The remaining data sets fall in between these two extremes. Only the class label needs to be remembered for atomic clusters and a higher number of atomic clusters in a dataset imply less learning complexity.

Fig 6 represents the classification accuracy achieved on the data sets as they are partitioned into one to ten clusters. We considered only one cluster layer ($N_{layers} = 1$) for this experiment. The graphs in Fig 6 demonstrate that the classification accuracies change significantly with the data set being partitioned at different number of clusters. In general it is beneficial to have higher number of clusters due to the fact that it identifies non–atomic clusters containing highly overlapping data points from different clusters. This provides the overlapping regions in the data set and the corresponding base classifier learns the patterns efficiently leading to higher classification accuracy. It can be observed that the classification accuracy degrades sometimes at higher number of clusters. One of the main reasons for performance degradation at some higher number of clusters is data imbalance. When the data set is partitioned, some clusters contain only one or two examples of a class and significant number of patterns from other classes. This sometimes leads to poor classification performance at higher number of clusters. At extremely high number of clusters there is insufficient data in each cluster for efficient classifier learning. As the number of clusters equals the number of training patterns, all the clusters become atomic. This results in memorization leading to poor generalization and classification accuracy.

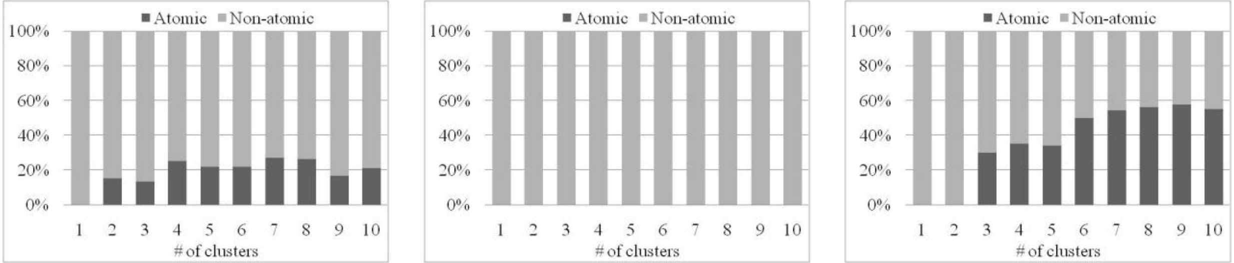
The impact on the change in accuracy with respect to clusters is more significant for some datasets like *Vowel* (4.93) and *Ionosphere* (1.80) than others. The impact is relatively smaller on some data sets including *Wine* (0.51), *Diabetes* (0.87), and *Breast Cancer* (0.23). In majority of the data sets the best classification accuracies are achieved when data is partitioned into more than one cluster. Note that in a clustered ensemble a pattern can belong to one cluster only and thus one classifier is trained on a pattern. Similarly only one decision is available on a test pattern. The objective of obtaining multiple decisions on a pattern is not achieved in clustered ensembles and thus leaves space for improvement.



Breast Cancer *Diabetes* *Ionosphere*

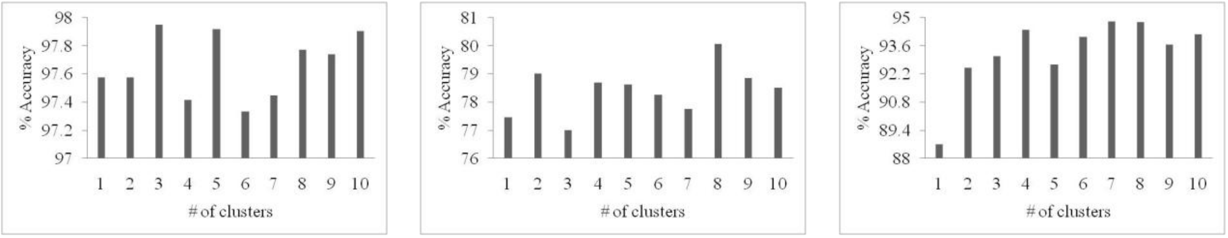


Parkinsons *Segment* *Sonar*

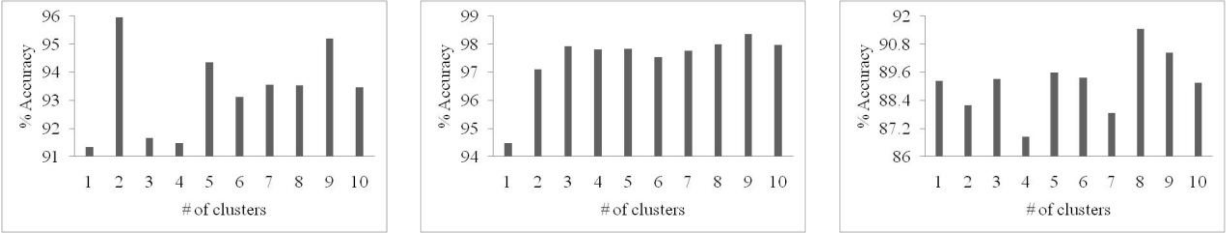


Spam *Vowel* *Wine*

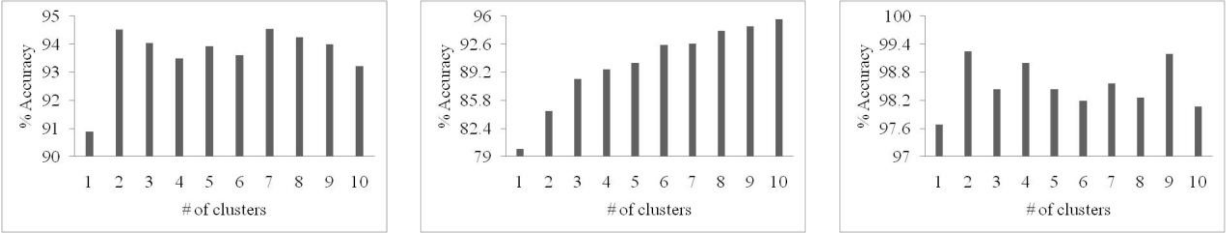
Fig 5: Atomic and non–atomic clusters as the data sets are partitioned into one to ten clusters achieved using the proposed ensemble classifier with one cluster layer. The numbers are averaged over the ten folds for each data set.



Breast Cancer *Diabetes* *Ionosphere*



Parkinsons *Segment* *Sonar*



Spam *Vowel* *Wine*

Fig 6: Classification accuracies at different number of clusters achieved using the proposed ensemble classifier with one cluster layer.

B. *Impact of Layers on Ensemble Classifier Learning*

Fig 7 represents the class–cluster co-occurrence matrices obtained at different layers for some data sets namely *Breast Cancer* and *Ionosphere*. Note that the content of the clusters changes in all cases as the clustering parameters are initialized randomly at different layers. Identical patterns belonging to different clusters indicate dissimilar learning of the base classifiers making their errors non–correlated thus achieving diversity and inclusion of higher number of layers achieves higher diversity. This is evidenced from Fig 8 where change of KW variance is represented with respect to the change of number of layers for different data sets. Note that the trend lines show that addition of layers increases diversity in general. As a pattern belongs to different cluster and thus learned by different base classifier at different layer, the errors made are uncorrelated. Addition of layers thus produces more non–correlated base classifiers leading to higher diversity.

Fig 9 represents the classification accuracies achieved as the data sets are partitioned at one to ten layers. The trend lines in majority of the graphs in Fig 9 show increasing classification accuracy at higher number of layers. A set of diverse base classifiers make dissimilar errors on identical patterns and increase the chance of obtaining the correct classification. This relationship between diversity and accuracy is evidenced from the trend of change of diversity and accuracy in Fig 8 and Fig 9 respectively. It can be observed that increasing diversity implies increasing accuracy for majority of the data sets. The proposed ensemble classifier provides the provision to change diversity and thus accuracy with the number of layers.

The variation of accuracy in Fig 9 with respect to the number of layers is high for *Sonar* (1.44), *Ionosphere* (1.06) and *Vowel* (1.49) data sets. Note that the variation was high too with respect to number of clusters for *Ionosphere*, and *Vowel* data sets. Overlapping clusters at different layers lead to better learning for these data sets. The *Breast Cancer* (0.06) and *Wine* (0.04) data sets are least affected by the change in number of layers.

Cluster

Class	1	2	3	4	5
1	166	3	5	226	0
2	0	51	80	13	71

Layer 1

Cluster

Class	1	2	3	4	5
1	1	7	0	159	233
2	80	77	44	14	0

Layer 2

Cluster

Class	1	2	3	4	5
1	5	3	159	233	0
2	79	50	15	0	71

Layer 3

(a) *Breast Cancer*

Cluster

Class	1	2	3	4	5
1	21	0	42	0	139
2	47	24	2	9	31

Layer 1

Cluster

Class	1	2	3	4	5
1	114	22	26	8	32
2	24	7	3	77	2

Layer 2

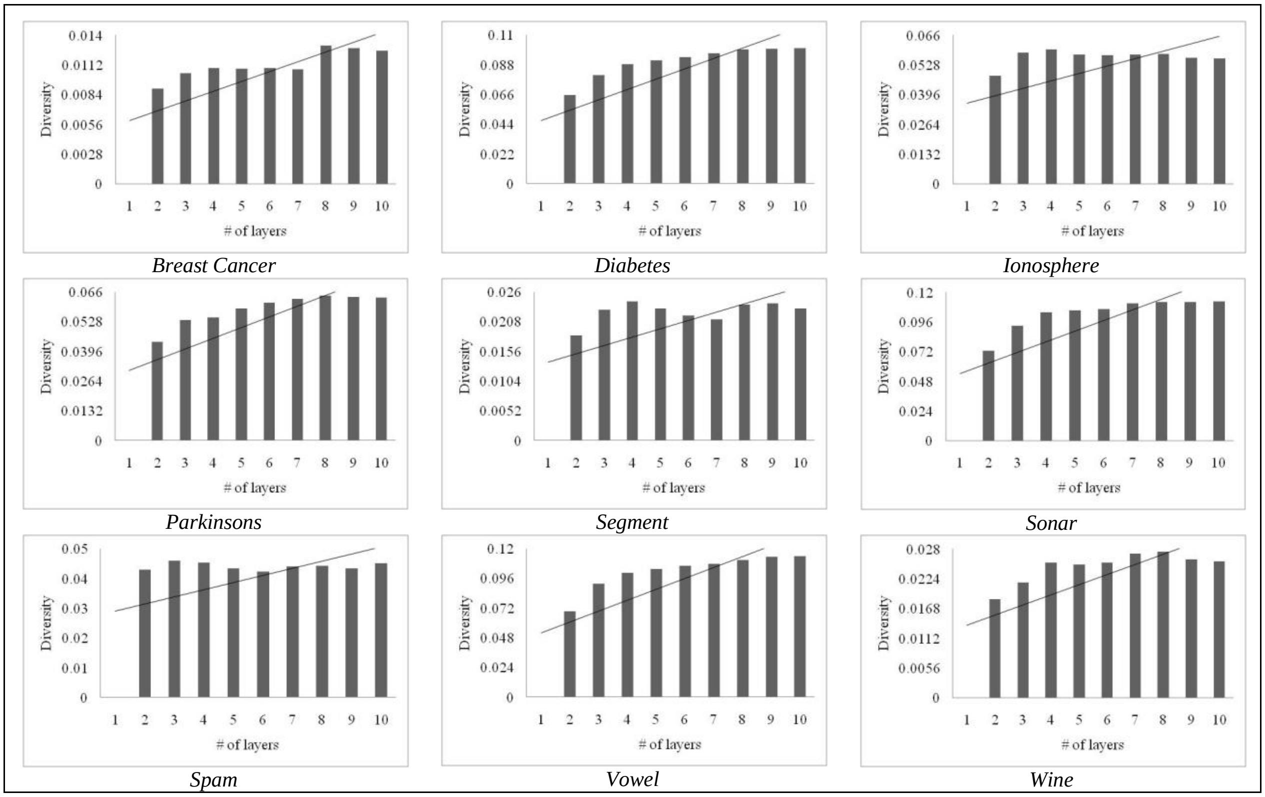
Cluster

Class	1	2	3	4	5
1	123	8	20	32	19
2	29	76	4	2	2

Layer 3

(b) *Ionosphere*

Fig 7: Impact of layers on clustering with the proposed ensemble classifier.

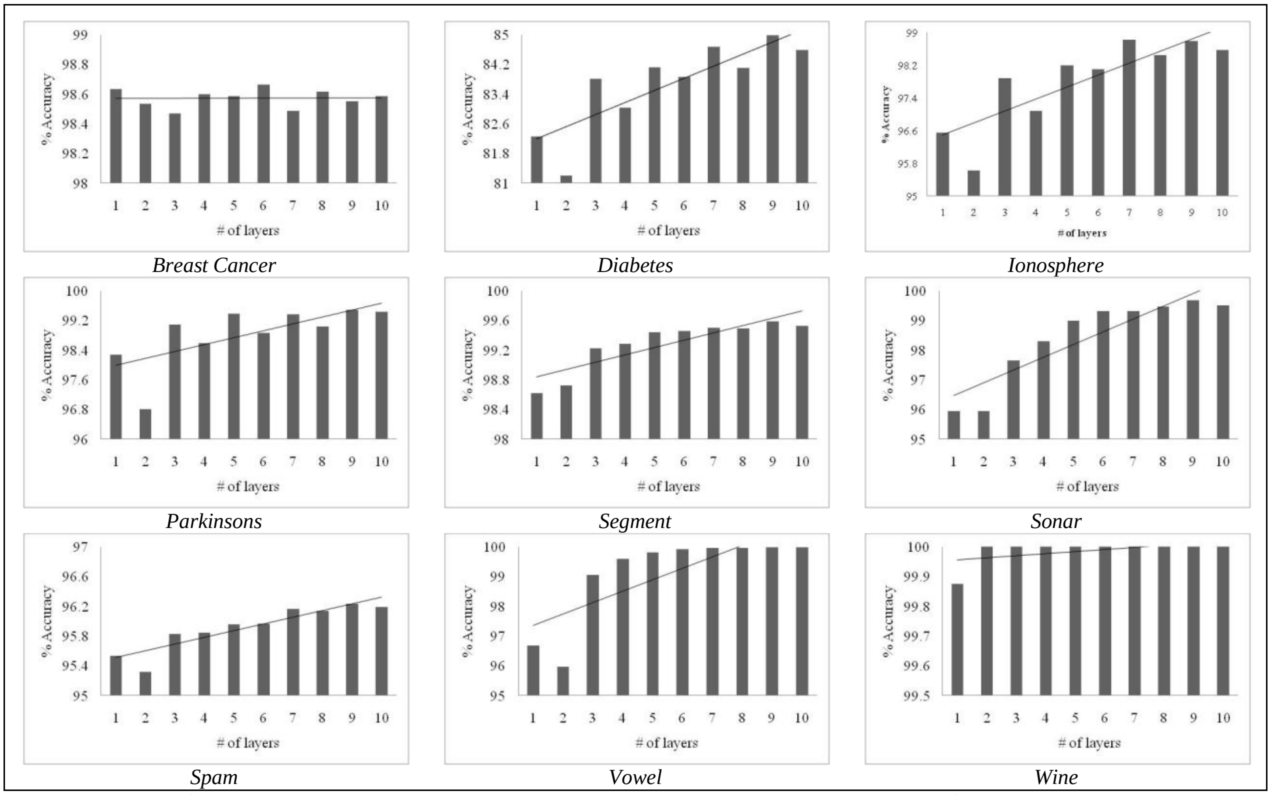


Breast Cancer *Diabetes* *Ionosphere*

Parkinsons *Segment* *Sonar*

Spam *Vowel* *Wine*

Fig 8: A graph showing change in diversity (KW variance) as the number of layer changes for the different data sets in Table II.



Breast Cancer *Diabetes* *Ionosphere*

Parkinsons *Segment* *Sonar*

Spam *Vowel* *Wine*

Fig 9: Classification accuracies on training sets when clustered at one to ten layers and their decisions fused using majority voting.

C. *Optimal Parameter Settings*

The performance of the proposed approach is constrained by two parameters – (i) number of clusters, and (ii) number of layers. The best parameter settings for the data sets are obtained based on the best training set accuracy of each fold in the data set. The optimal number of clusters and layers for each fold in the data sets and corresponding average training and test set accuracies are presented in Table III. In majority of the folds best training accuracy is obtained at more than one layer. This can be attributed to the fact that the inclusion of more cluster layers implies the learning of identical patterns by multiple classifiers and at the same time training sets for the base classifiers are significantly different. This leads to diversified base classifiers and better classification accuracy as shown in the previous section. In data sets like *Satellite*, *Segment*, *Pendigits*, *Vehicle*, and *Vowel*, the best accuracies are achieved at higher number of layers. Addition of layers do not improve the scenario that much for some data sets like *Wine* as majority folds achieve best training accuracy at one layer. This is evidenced from the fact that the impact of changing clusters and layers on *Wine* data set is 0.51 and 0.04 respectively. This implies that the patterns from different classes in these data sets are relatively well separated.

D. *Comparative Performance Analysis*

A comparison of the classification accuracies between the proposed approach and three commonly related ensemble classifier generation methods namely *clustered ensemble* [19], *bagging* [28], and *boosting* (AdaBoostM1) is provided in Table IV. The proposed approach performs better than *clustered ensemble* in nineteen out of twenty cases. The proposed approach obtains multiple decisions from a set of diverse base classifiers whereas the clustered ensemble [19] relies on the decision of an individual classifier. This accounts for the better performance of the proposed approach.

The proposed approach outperforms boosting in sixteen out of twenty cases and outperforms bagging in thirteen out of twenty cases. The proposed approach identifies difficult–to–classify overlapping patterns by clustering and can improve diversity by increasing the number of layers as evidenced in Fig 8. This is where the proposed approach takes the lead. Overall the proposed approach performs 4.27% better than clustered ensemble, 1.08% better than bagging and 1.73% better than boosting. The combination of clustering and layering leads to better performance and puts the proposed approach ahead of others. We justify this claim by conducting a two tailed Wilcoxon Signed Rank test [45][46] as presented in Table V. Note that the null hypothesis is rejected in all cases either at 0.01, 0.20, or 0.05 significance level indicating the fact that the proposed approach performs significantly better than the related ensemble classifiers.

VI. CONCLUSION

In this paper, we have proposed a novel approach for generating ensemble of classifiers and evaluated it on benchmark datasets from UCI machine learning repository. The proposed approach partitions the data set into multiple clusters at different layers and trains a set of base classifiers on the patterns within a cluster. The classification decision of a test pattern is achieved by finding the decision of the corresponding base classifier at each layer and fusing their decisions using majority voting.

The experiments on twenty benchmark datasets have been conducted. The proposed approach has significantly improved the overall classification accuracy. Based on evidence from experiments, we draw the following conclusions (1) the classification accuracy has increased with the increase in number of layers. The impact of number of layers on classification accuracy is substantial; (2) the impact of number of clusters is also noteworthy as the best classification accuracy is achieved when data sets are segmented into two or more clusters in general; and (3) the proposed approach

Table III: Optimal parameters based on the best performance of the training folds for the data sets in Table II.

Data Set	**Layer/Cluster no.**										**Average Training Accuracy**	**Average Test Accuracy**
	F1	**F2**	**F3**	**F4**	**F5**	**F6**	**F7**	**F8**	**F9**	**F10**		
Breast Cancer	9/4	4/5	2/2	2/8	1/3	1/5	10/5	1/5	4/6	5/8	98.89±0.21	97.37±1.67
Diabetes	9/8	7/9	3/10	7/9	3/8	9/10	9/10	10/8	9/9	7/8	85.29±0.99	73.57±3.73
Ecoli	9/10	10/8	10/9	6/10	9/7	10/9	10/9	7/9	9/10	10/9	96.32±0.63	90.25±4.00
German	3/7	1/8	5/9	3/7	9/8	9/7	1/9	9/4	7/7	7/8	83.03±1.21	72.90±3.00
Glass	6/3	4/2	3/4	6/1	1/1	6/1	10/3	4/4	5/3	9/1	100.0±0.00	96.82±2.20
Ionosphere	10/7	5/6	9/2	6/2	5/1	7/4	7/2	9/1	7/3	7/1	98.92±0.55	90.83±4.73
Iris	4/9	2/9	2/4	4/7	3/2	4/3	1/1	6/10	8/3	4/5	100.0±0.00	98.00±3.22
Liver	10/5	3/10	10/10	8/4	8/9	8/6	6/8	6/5	10/7	6/6	86.44±1.35	67.62±6.78
Parkinsons	7/10	7/10	9/9	7/9	9/8	5/10	10/6	10/5	5/8	9/7	99.60±0.38	96.06±2.72
Pendigits	10/9	7/10	10/9	10/10	10/10	9/9	9/10	8/10	10/8	10/8	99.82±0.03	99.28±0.20
Satellite	9/9	9/8	9/10	3/10	8/10	10/9	10/10	9/10	7/10	9/10	94.21±0.26	89.74±0.73
Segment	9/10	9/9	6/6	10/10	9/10	7/9	9/10	9/10	9/10	9/9	99.60±0.11	97.88±1.20
Sonar	9/4	9/7	7/8	7/8	10/3	9/10	8/8	9/10	4/6	8/4	99.73±0.68	90.85±4.76
Spam	9/10	8/8	10/10	5/9	7/9	9/7	9/6	3/10	9/9	7/9	96.29±0.15	93.20±1.42
Spect	10/6	9/10	9/10	10/9	6/7	6/9	1/6	7/8	6/10	2/9	90.64±0.92	78.98±6.24
Thyroid	5/4	1/7	3/2	3/1	4/1	9/8	5/3	3/6	3/8	3/2	100.0±0.00	99.55±1.44
Transfusion	10/10	2/8	8/8	9/8	3/9	8/7	6/8	2/8	3/9	4/8	82.37±0.49	79.70±1.60
Vehicle	9/6	10/5	10/10	10/9	9/9	10/9	9/9	9/9	8/6	10/9	97.46±0.53	81.88±4.34
Vowel	7/7	9/8	10/10	10/8	9/6	8/10	8/10	8/10	9/6	7/9	100.0±0.00	97.98±1.51
Wine	1/7	1/2	2/1	1/4	3/5	1/3	3/2	1/4	1/7	1/2	100.0±0.00	100.0±0.00

Table IV: Comparative classification accuracy (%) between the proposed and existing ensemble classifiers with majority win fusion. The best accuracy for each data set is marked bold.

Dataset	Clustered Ensemble	Bagging	Boosting	Proposed Approach
Breast Cancer	96.65	97.09	96.94	**97.37**
Diabetes	73.17	**76.02**	74.62	73.57
Ecoli	85.97	88.67	88.90	**90.25**
German	69.40	**74.90**	71.60	72.90
Glass	89.41	95.91	95.45	**96.82**
Ionosphere	88.80	**91.30**	90.00	90.83
Iris	94.67	96.67	97.33	**98.00**
Liver	62.57	70.24	**70.71**	67.62
Parkinsons	89.90	91.82	90.20	**96.06**
Pendigits	98.61	95.02	94.71	**99.28**
Satellite	87.69	**91.06**	89.51	89.74
Segment	95.97	96.80	95.72	**97.88**
Sonar	78.37	85.51	82.66	**90.85**
Spam	92.26	92.85	89.37	**93.20**
Spect	80.93	80.09	**81.25**	78.98
Thyroid	95.00	96.82	96.82	**99.55**
Transfusion	77.83	78.59	78.47	**79.70**
Vehicle	76.71	**84.24**	80.96	81.88
Vowel	89.09	89.29	**98.08**	97.98
Wine	98.33	97.78	97.22	**100.0**

Table V: Pair–wise classification performance comparison between the proposed ensemble classifier and the related ensemble classifiers using two–tailed *Wilcoxon Signed Rank test.*

Classifier Pair	Hypothesis Test
Proposed approach vs. clustered ensemble	*Null Hypothesis* H_0: The proposed approach is equivalent to clustered ensemble *Alternative Hypothesis* H_1: The proposed approach is significantly better than clustered ensemble *Wilcoxon Signed Rank Test*: $R^+ = 202$, $R^- = 8$, and $T = 8$. Null hypothesis H_0 rejected at significance level $\alpha = 0.01$.
Proposed approach vs. bagging	*Null Hypothesis* H_0: The proposed approach is equivalent to bagging *Alternative Hypothesis* H_1: The proposed approach is significantly better than bagging *Wilcoxon Signed Rank Test*: $R^+ = 139.5$, $R^- = 70.5$, and $T = 70.5$. Null hypothesis H_0 rejected at significance level $\alpha = 0.20$.
Proposed approach vs. boosting	*Null Hypothesis* H_0: The proposed approach is equivalent to boosting *Alternative Hypothesis* H_1: The proposed approach is significantly better than boosting *Wilcoxon Signed Rank Test*: $R^+ = 173$, $R^- = 37$, and $T = 37$. Null hypothesis H_0 rejected at significance level $\alpha = 0.05$.

performs 4.27%, 1.08% and 1.73% better than clustered ensemble, bagging and boosting. The improvement is significant in terms of classification accuracies as evidenced from the two–tailed *Wilcoxon Signed Rank test*. In our future research, we aim to investigate the optimality issues of the number of clusters and layers considering a broader range, the impact of variable clustering at different layers, and the impact of data imbalance at the clusters.

REFERENCES

[1] T. Windeatt, "Accuracy/Diversity and ensemble MLP classifier design," IEEE Trans. on Neural Networks, vol. 17, no. 5, pp. 1194–1211, 2006.

[2] H. Zouari, L. Heutte, and Y. Lecourtier, "Controlling the diversity in classifier ensembles through a measure of agreement," Pattern Recognition, vol. 38, pp. 2195–2199, 2005.

[3] R. Polikar, "Ensemble based systems in decision making," IEEE Circuits and Systems Magazine, vol. 6, no. 3, pp. 21–45, 2006.

[4] T. Windeatt, "Diversity/accuracy and ensemble classifier design," Int. Conf. on Pattern Recognition, vol. 3, pp. 454–457, 2004.

[5] 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.

[6] L. I. Kuncheva, J. C. Bezdek, and R. Duin, "Decision templates for multiple classifier fusion: an experimental comparison," Pattern Recognition, vol. 34, no. 2, pp. 299–314, 2001.

[7] A. H. R. Ko, R. Sabourin, A. de S. Britto, and L. Oliveira, "Pairwise fusion matrix for combining classifiers," Pattern Recognition, vol. 40, pp. 2198–2210, 2007.

[8] N. M. Wanas, R. A. Dara, and M. S. Kamel, "Adaptive fusion and co-operative training for classifier ensembles," Pattern Recognition, vol. 39, pp. 1781–1794, 2006.

[9] O. R. Terrades, E. Valveny, and S. Tabbone, "Optimal classifier fusion in a non-Bayesian probabilistic framework," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 31, no. 9, pp. 1630–1644, 2009.

[10] T. G. Dietterich, "Ensemble methods in machine learning," Int. Workshop on Multiple Classifier Systems, pp. 1–15, 2000.

[11] R. Maclin and J. W. Shavlik, "Combining the predictions of multiple classifiers: using competitive learning to initialize neural networks," Int. Joint Conference on Artificial Intelligence, pp. 524–531, 1995.

[12] T. Yamaguchi, K. J. Mackin , E. Nunohiro, J. G. Park, K. Hara, K. Matsushita, M. Ohshiro, and K. Yamasaki, "Artificial neural network ensemble-based land-cover classifiers using MODIS data," Artificial Life and Robotics, vol. 13, no. 2, pp. 570–574, 2009.

[13] K. J. Cherkauer, "Human expert–level performance on a scientific image analysis task by a system using combined artificial neural networks," Working notes of the AAAI workshop on Integrating Multiple Learned Models, pp. 15–21. 1996.

[14] T. K. Ho, "The random subspace method for constructing decision forests," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 20, no. 8, pp. 832–844, 1998.

[15] A. Bertoni, R. Folgieri, and G. Valentini, "Bio-molecular cancer prediction with random subspace ensembles of Support Vector Machines," Neurocomputing, vol. 63, pp. 535–539, 2005.

[16] L. I. Kuncheva, J. J. Rodriguez, C. O. Plumpton, D. E. Linden, and S. J. Johnston, "Random subspace ensembles for FMRI classification," IEEE Trans. on Medical Imaging, vol 29, no. 2, pp. 531–542, 2010.

[17] G. Martínez–Muñoz, A. Sánchez–Martínez, D. Hernández–Lobato, and A. Suarez, "Class-switching neural network ensembles," Neurocomputing, vol. 7, pp. 2521–2528, 2008.

[18] T. G. Dietterich and G. Bakiri, "Solving multiclass learning problems via error–correcting output codes," Journal of Artificial Intelligence Research, vol. 2, pp. 263–286, 1995.

[19] L. Rokach, O. Maimon, and I. Lavi, "Space decomposition In data mining: a clustering approach", Int. Symp. On Methodologies For Intelligent Systems, pp. 24–31, 2003.

[20] J. Xiuping and J. A. Richards, "Cluster-space classification: a fast k-nearest neighbor classification for remote sensing hyperspectral data," IEEE Workshop on Advances in Techniques for Analysis of Remotely Sensed Data, pp. 407–410, 2003.

[21] L. I. Kuncheva, "Cluster-and-selection method for classifier combination," Int. Conf. on Knowledge-Based Intelligent Engineering Systems & Allied Technologies (KES), 185–188, 2000.

[22] L. I. Kuncheva, "Switching between selection and fusion in combining classifiers: an experiment," IEEE Trans. on Systems, Man and Cybernetics, vol. 32, no. 2, pp, 146–156, 2002.

[23] B. Tang, M. I. Heywood, and M. Shepherd, "Input partitioning to mixture of experts," Int. Joint Conf. on Neural Networks, pp. 227–232, 2002.

[24] G. Nasierding, G. Tsoumakas, A. Z. Kouzani, "Clustering based multi-label classification for image annotation and retrieval," IEEE Int. Conf. on Systems, Man and Cybernetics, pp. 4514–4519, 2009.

[25] H. Cevikalp and R. Polikar, "Local classifier weighting by quadratic programming," IEEE Transactions on Neural Networks, vol. 19(10), pp. 1832–1838, 2008.

[26] L. Nanni and A. Lumini, "Evolved feature weighting for random subspace classifier", IEEE Transactions on Neural Networks, vol. 19(2), pp. 363–366, 2008.

[27] M. J. Jordan and R. A. Jacobs, "Hierarchical mixtures of experts and the EM algorithm," Neural Computation, vol. 6, no. 2, pp. 181–214, 1994.

[28] L. Breiman, "Bagging predictors," Machine Learning, vol. 24, no. 2, pp. 123–140, 1996.

[29] L. Breiman, “Random forests,” Machine Learning, vol. 45, no. 1, pp. 5–32, Oct. 2001.

[30] L. Breiman, “Pasting small votes for classification in large databases and on-line,” Machine Learning, vol. 36, pp. 85–103, 1999.

[31] G. Martínez-Muñoz and D. Hernández-Lobato, and A. Suarez, “An analysis of ensemble pruning techniques based on ordered aggregation,” IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 31, no. 2, pp. 245–259, 2009.

[32] L. Chen and M. S. Kamel, “A generalized adaptive ensemble generation and aggregation approach for multiple classifiers systems,” Pattern Recognition, vol. 42, pp. 629–644, 2009.

[33] L. Nanni and A. Lumini, “Fuzzy bagging: a novel ensemble of classifiers,” Pattern Recognition, vol. 39, pp. 488–490, 2006.

[34] S. Eschrich and L. O. Hall, “Soft partitions lead to better learned ensembles,” Proc. Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS), pp. 406–411, 2002.

[35] R. E. Schapire, “The strength of weak learnability,” Machine Learning, vol. 5, no. 2, pp. 197–227, 1990.

[36] H. Drucker, C. Cortes, L. D. Jackel, Y. LeCun, and V. Vapnik, “Boosting and other ensemble methods,” Neural Computation, vol. 6, no. 6, pp. 1289–1301, 1994.

[37] Y. Freund and R. E. Schapire, “Decision-theoretic generalization of on-line learning and an application to boosting,” Journal of Computer and System Sciences, vol. 55, no. 1, pp. 119–139, 1997.

[38] N. García–Pedrajas, “Constructing ensembles of classifiers by means of weighted instance selection,” IEEE Trans. on Neural Networks, vol. 20, no. 2, pp. 258–277, 2009.

[39] J. J. Rodriguez and J. Maudes, “Boosting recombined weak classifiers,” Pattern Recognition Letters, vol. 29, pp. 1049–1059, 2008.

[40] D. Parikh and R. Polikar, “Ensemble based incremental learning approach to data fusion,” IEEE Trans. on Systems, Man and Cybernetics, vol. 37, no. 2, pp. 437–450, 2007.

[41] M. D. Muhlbaier, A. Topalis, and R. Polikar, “Learn++.NC: Combining ensemble of classifiers with dynamically weighted consult-and-vote for efficient incremental learning of new classes,” IEEE Trans. on Neural Networks, vol. 20, no. 1, pp. 152–168, 2009.

[42] UCI Machine Learning Database, http://archive.ics.uci.edu/ml/, accessed on 6th October 2009.

[43] R. Kohavi and D. H. Wolpert, “Bias plus variance decomposition for zero-one loss functions,” Proc. Int. Conf. on Machine Learning, pp. 275–283, 1996.

[44] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The WEKA data mining software: an update,” SIGKDD Explorations, vol. 11, no. 1, 2009.

[45] J. Demsar, “Statistical comparisons of classifiers over multiple data sets,” Journal of Machine Learning Research, vol. 7, pp. 1–30, 2006.

[46] D. J. Sheskin, “Handbook of parametric and nonparametric statistical procedures,” Chapman & Hall/CRC, 2000

[47] H. Parvin, H. Alizadeh, and B. Minaei–Bidgoli, “Using clustering for generating diversity in classifier ensemble,” Int. Journal of Digital Content Technology and its Applications, vol. 3, no. 1, pp. 51–57, 2009.

[48] R. Neumayer, “Clustering based ensemble classification for spam filtering,” Proc. Workshop on Data Analysis (WDA), pp. 11–22, 2006.