Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 492387, 14 pages
http://dx.doi.org/10.1155/2014/492387

Hindawi

Research Article

Robust Framework to Combine Diverse Classifiers Assigning Distributed Confidence to Individual Classifiers at Class Level

Shehzad Khalid,[1] Sannia Arshad,[1] Sohail Jabbar,[2,3] and Seungmin Rho[4]

[1] *Department of Computer Engineering, Bahria University, Islamabad 44000, Pakistan*
[2] *Department of Computer Science, COMSATS Institute of Information Technology, Sahiwal, Pakistan*
[3] *Department of Computer Science, Bahria University, Islamabad 44000, Pakistan*
[4] *Department of Multimedia, Sungkyul University, Anyang-si, Republic of Korea*

Correspondence should be addressed to Shehzad Khalid; shehzad_khalid@hotmail.com
and Seungmin Rho; korea.smrho@gmail.com

Received 7 July 2014; Accepted 12 August 2014; Published 8 September 2014

Academic Editor: Changhoon Lee

We have presented a classification framework that combines multiple heterogeneous classifiers in the presence of class label noise. An extension of *m*-Mediods based modeling is presented that generates model of various classes whilst identifying and filtering noisy training data. This noise free data is further used to learn model for other classifiers such as GMM and SVM. A weight learning method is then introduced to learn weights on each class for different classifiers to construct an ensemble. For this purpose, we applied genetic algorithm to search for an optimal weight vector on which classifier ensemble is expected to give the best accuracy. The proposed approach is evaluated on variety of real life datasets. It is also compared with existing standard ensemble techniques such as Adaboost, Bagging, and Random Subspace Methods. Experimental results show the superiority of proposed ensemble method as compared to its competitors, especially in the presence of class label noise and imbalance classes.

1. Introduction

In recent years, there has been a growth of research interest in the development of sophisticated data classification techniques as it has many practical applications in variety of fields such as object recognition, surveillance, medical imaging, and financial data analysis. These techniques have been widely used in industry in the form of tools and commercial applications giving so many benefits to them.

Classification of unseen data requires building model of normality of commonly followed patterns that exist in a given domain. Once these models for all the known patterns are learnt, they are then used to classify unseen samples to one of the modelled patterns. A variety of machine learning approaches have been proposed that model the normal pattern and classify newly coming data using the generated models of normality. Statistical approaches for classification [1–3] model a pattern by approximating the density of the training samples belonging to the given pattern. Khalid and Naftel [3] model the pattern by approximating a single multivariate Gaussian for each class. Classification of unseen samples is then performed using Mahalanobis classifier. Classification approaches based on Gaussian Mixture Model (GMM) have also been proposed [4–7]. Various techniques [8–11] based on support vector machines (SVM) have also been presented. The underlying logic behind SVM-based approaches is to identify an optimal hyperplane separating training data belonging to various patterns and then classifying new data based on these identified decision boundaries. One class classifier based on SVM (OCC-SVM) has also been employed for classification and anomaly detection [12, 13]. Neural network based classifiers have also been reported in literature [14, 15]. Owens and Hunter [14] employ self-organizing maps to learn model of normality of given set of patterns. Various *m*-Mediods based approaches have been proposed for data classification and anomaly detection [16–19].

There are many scenarios of distribution of samples belonging to different classes with respect to each other. There may be a high level of overlap in the distributions of samples belonging to different classes. In other cases, samples belonging to various classes may be exhibiting complex shape nonoverlapping distributions with tight decision boundaries amongst them. Similarly, we may have classes that exhibit multivariate distributions of samples within individual patterns. Another commonly occurring phenomenon is that different classes in a dataset have widely varying number of training samples for different classes which is normally referred to as class imbalance problem. This phenomenon is frequent in medical world where the sensitive, dangerous, and later stage of diseases is not very frequent, thus resulting in lesser amount of data available for training our classifiers. On the other hand, nonmalignant forms of these diseases are often commonly occurring and hence the large number of samples is available for training our machine learning algorithms. As general classifiers normally optimize their results by having overall higher classification accuracy, they focus more attention on correctly classifying samples from heavily populated classes as compared to classes which may be sensitive but having lesser number of membership counts. This is in total contrast to our actual expectations. Misclassification of insensitive diseases may be ignored, but doing the same with serious disease classes may result in loss of lives.

The above-mentioned and variety of other scenarios impose challenges for the classifiers. Individual classifiers may be more expert to handle some of these discussed problems, but there is a possibility that it may give poor performance in the presence of others. For instance, GMM is known to perform good in the presence of overlapping distribution of samples belonging to different classes but does not yield good results in the presence of nonoverlapping distributions with tight and complex decision boundaries. On the other hand, SVM performs well in the presence of complex decision surfaces between nonoverlapping distribution, but its performance degrades in the presence of class imbalance problem. Similarly, m-Mediods [19] can handle multivariate distribution of samples within a pattern and can handle overlapping distributions belonging to different classes to a reasonable extent.

To overcome this problem, combining classifiers in an ensemble has gained interest quite recently [20–43] in literature. The underlying logic behind classifier ensemble is that many classifiers are combined together in a certain framework to make a final strong classifier whose decision is expected to be more precise and effective as compared to its individual components. The ensemble-based classification approaches can be broadly categorized into two types: homogeneous and heterogeneous classifier ensembles. The homogeneous approaches combine classification algorithms of the same type. Contrary to this, heterogeneous ensemble combines classification algorithms of different types. The most popular and standard ensemble methods include Bagging [30], Boosting (Adaboost) [31], and Random Subspace Methods (RSM) [44] (generalization of Random Forest Method [32]). These are the dominant methods for diversifying and combining classification results.

Most of the existing classification techniques assume that the training data is free of problems such as class label noise and class imbalance problems. There exist some ensemble methods [45–49] that handle class imbalance problem in datasets. There also exist very few methods that cater for the problem of class label noise in the datasets. In this paper, we present a novel classification framework that can handle problems such as class label noise and imbalance classes. An extension of m-Mediods based classification approach is proposed that learns the model of normality of classes whilst identifying and filtering training samples representing class label noise. m-Mediods classifier is further combined with other well-known classifiers using a proposed framework. We selected GMM and one class classifier based on SVM (OCC-SVM) which are carefully selected based on their capabilities to handle different types of class distributions. The learned models using selected classifiers are then combined by introducing genetic algorithm based weight learning method to learn weights at class level individually for all heterogeneous classifiers. The probabilistic output of each classifier is then combined in a weighted combination at class level to achieve better classification performance. The proposed framework is robust to the presence of class imbalance problem, class label noise, and the presence of multivariate distribution of samples within classes.

The remaining paper is organized as follows. In Section 2, we present a brief review of existing classifier ensemble techniques. Section 3 presents an overview of the learning of proposed ensemble framework and classification of unseen data using the learned ensemble. Section 4 presents different classifiers that are used for the construction of proposed ensemble. In Section 5, an extension of m-Mediods based approach to filter class label noise is presented. The detailed description of proposed ensemble framework is given in Section 6. Experiments have been conducted to show the effectiveness of proposed approach as compared to the competitors These experiments are reported in Section 7. The last section summarizes the paper.

2. Background and Related Work

Classifier ensembles are known to be very useful methods for improving the classification accuracy as well as diversity. They combine multiple classifiers together to get a single stronger one whose performance is more precise and accurate as compared to its individual members. A variety of factors have been considered in literature to improve the accuracy of the ensemble. These include classifier selection [20, 21, 28, 50], feature selection [21, 27, 32], diversity creation in ensemble of classifiers [34, 39, 40], combination methods [20–23, 25, 30–33, 39], and combining more than one ensemble [24, 41, 42]. Certain ensemble approaches integrate some statistical procedures with them such as Bagging with Principal Component Analysis (PCA) [21], weighting classifier dynamically based on cross validation [22], Dempster-Shafer theory of evidence [25], and supervised projection [26].

Earlier work has put a lot of concentration towards assigning weights to instances as well as classifiers to construct

an ensemble by applying weight assignment methods [51] and voting algorithms [29–32, 35, 52, 53]. Bagging [30] and Boosting [31] are one of the most well-known voting and weighting based standard ensemble methods. Breiman [30] introduced the Bagging algorithm which is an independent ensembling method as the output produced by one classifier does not depend on the output of previous classifiers. Bootstrap samples are generated from training set which are obtained by sampling with replacement. A classifier is then learned with different training set in each iteration. It follows the voting approach in order to combine the predictions of classifiers. C.-X. Zhang and J. Zhang [21] combined the concept of Bagging with Principal Component Analysis (PCA) to construct an ensemble. Bauer and Kohavi [52] provided the variant of Bagging, namely, wagging, which makes stochastic assignment of weights to each instance.

Random Forest Method proposed by Breiman [32] uses unpruned decision trees. It consists of a collection of trees like structured classifiers and uses a number of input variables to find the decision at a node of the tree. To classify a new pattern, this algorithm collects votes from every tree in the forest and then uses majority voting to finalize the class label. The generalization of Random Forest approach, referred to as Random Subspace Method (RSM), is also presented [44]. Instead of working only with decision trees, as in the case of Random Forest, RSM can take into account any classifier such as nearest neighbor classifier and support vector machine. A subspace from the feature space representation is identified by randomly selecting a subset of features.

Freund and Schapire proposed Boosting [31] algorithm which enhances the performance of a weak learner by iteratively running it on training data. García-Pedrajas and García-Osorio [26] combine the concept of Boosting with supervised projection method. The technique focuses on misclassified instances as they are used to find supervised projection of data. After getting the projections, the next classifier is trained on it. This approach does not employ majority voting or weighting scheme to combine the classifiers. Merler et al. [53] improve the simple Boosting algorithm by an iterative process which focuses on the inaccuracies produced by the previous classifiers. It entirely concentrates on those samples which are hard to classify. It is a homogeneous and dependent ensemble method. It takes the whole training set in each of its iterations and does not create the bootstrap of samples. Equal weights are initially assigned to every sample in a training dataset. After every iteration, weights of misclassified instances are increased while decreasing the weights of correctly classified ones. It further assigns weight to individual classifier to measure its overall accuracy. The higher weights are given to those classifiers performing accurately. New samples are classified using these weights.

Approaches using dynamic ensemble learning have also been proposed in literature [22, 27]. Zhu et al. [22] provided a new ensemble model named dynamic weighting ensemble classifier based on cross validation (DWEC-CV). In this method, different classifiers are used for different samples. To train and construct a classifier, Random Subspace Method is used. The feature subspaces from the original feature set are selected randomly. The number of feature subspaces selected determines the number of classifiers to be produced. A weight adjusted voting algorithm [29] is introduced which uses a weight vector to weight instances as well as the classifiers. The weight vector, for instance, gives higher weights to those instances which are difficult to classify. On the other side, the classifiers weight vector gives the highest weights to only those classifiers giving better performance on these difficult instances. These classifier weights are identified by those samples having higher weights. DECORATE [54] is a diversity creation method to construct a classifiers ensemble. It creates the additional artificial training data in order to create the diverse classifier ensemble. The methods for dynamic ensemble selection [28] using majority voting rule to combine the classifiers have also been introduced in literature. The methods in [24, 41, 42] combine more than one ensemble in order to achieve improved accuracy and diversity of classifiers. Al-Ani and Deriche [25] combine classifiers using Dempster-Shafer theory.

The problem with majority of exiting ensembles is that they do not cater for the presence of noise in the training data including feature space noise and class label noise, although we expect to have noise related problems in real life datasets. Development of approaches that are robust to various problems such as feature space noise and class label noise has received scant attention. Dietterich [55] performed comparative analysis of standard ensembling techniques in the presence of class label noise. It has been shown that accuracies of these ensembling techniques such as Bagging, Adaboost, and Random Subspace Method degrade in the presence of noise. We will also show through our experimental evaluation that Adaboost, Bagging, and RSM are also not performing well on imbalanced and overlapping classes.

There exist few approaches which cater for the presence of noise in the training data. One of them is a method based on Boosting with supervised projection [26] which shows the noise tolerance of its proposed ensemble method. It compares only one standard ensemble method such as Adaboost with their proposed ensemble using different base learners to prove the sensitivity of Adaboost toward different levels of class label noise. In literature, there exist some researches which show that Boosting methods such as Adaboost degrade their performance when some level of noise is present in the dataset [56]. Arjun and Arora introduced the TRandom [45] Adaboost which performs better than the simple Adaboost in the presence of low and high noise data. In literature there exist classifier methods [46] constructed from the imbalanced datasets. A dataset is said to be imbalanced if the number of instances for all the classes is not equally represented. In literature, ensemble methods [47–49, 57, 58] have been proposed to improve their performance on imbalanced datasets. Zhou et al. [57] provided the relationship between ensemble and its Neural Network components for regression and classification. An EasyEnsemble method and BalanceCascade method have been introduced [47] to handle the class imbalance problem. EasyEnsemble method produces subsets from the majority class and the base learner is trained using each of these subsets and at the end, output of those learners is combined. Ryan and Nitesh [48] gave an extension of Random Subspace

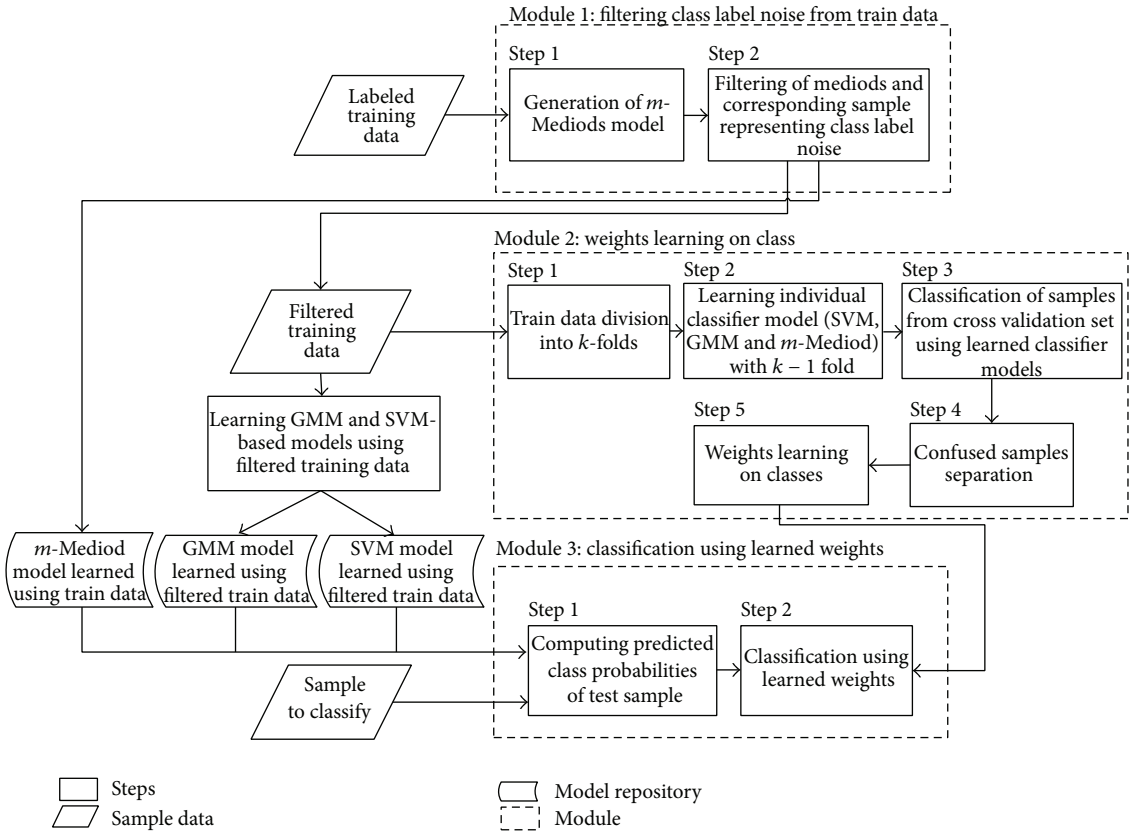

Figure 1: Overview of our proposed framework of classifier ensembles.

Method [44] that joins together SMOTE [46] to overcome the class imbalance problem. In literature, there is a cost sensitive ensemble method for class imbalance dataset [49]. It divides the instances of majority class into several subsets on the basis of imbalanced samples proportions. It then trains the subclassifiers using Adaboost method. Ensemble-based wrapper approach [58] for feature selection from the dataset has also been proposed. This method selects the feature from the data having highly imbalanced class distribution. It creates multiple balanced datasets from the original imbalanced one by doing sampling and then evaluates the feature subsets using ensemble classifiers each trained on balanced dataset.

The contribution of this paper is to present a robust classifier ensemble that combines existing state-of-the-art classifiers such as GMM and OCC-SVM with our proposed m-Mediods classifier in a weighted ensemble. A genetic algorithm based weight learning approach to optimize the accuracy of proposed classifier ensemble is presented. The proposed ensemble can handle real life issues of labelled datasets such as class label noise and class imbalance problems.

3. Overview of Proposed Ensemble Framework

Classification in the presence of class label noise and imbalance in the distribution of samples across different classes is a challenging task. Figure 1 presents an overview of our proposed framework of combining classifiers in an ensemble to handle this challenge. The proposed framework is composed of three main modules: (1) filtering of class label noise from train data to mitigate its effects on model learning process, (2) learning of weights on classes with respect to different classifiers, and (3) classification by combining their probabilistic output using learned weights. The module for filtering class label noise is based on the extension of our previously proposed m-Mediods classifier and is composed of two steps. In step 1, we generate the mediods based model to represent different classes. In step 2, we prune those mediods that are isolated or surrounded by mediods from different classes and representing fewer numbers of samples. This module generates the m-Mediod based model while filtering the samples representing class label noise. The filtered training data is then further utilized to generate model of normalities of other classifiers. The filtered training data is also used by second module to learn weights on the probabilistic output of classifiers with respect to different classes. We employed k-fold cross validation to learn the weights. The models of normality of various classifiers are learned on $k - 1$ fold whereas the left-out fold is used for cross validation. We speed up the learning process by identifying those samples from cross validation set for which there is a confusion; that is, all the classifiers do not predict same class for a given sample. Weight learning using genetic algorithm is then carried out by using only the confused samples. Learning weights at the class level with respect to

different classifiers will enable our proposed framework to extract appropriate benefit from the classifier speciality at the class level. A classifier may perform very good while predicting a subset of classes due to existence of a particular type of distribution among those subsets of classes. The same classifier may perform bad for other classes with samples exhibiting different types of distributions. For instance, a classifier performing good in highly overlapping distributions may not work well for nonoverlapping but complex and tight boundaries between them. The learned models of normality of classifiers using filtered training data and weights learned in second module are used by third module for classifying test data. The probabilistic output of each classifier with respect to different classes is combined using learned weights to yield improved classification performance.

4. Classifiers

In this section, we describe different classifiers that are used for the construction of proposed classifier ensemble. Given the feature vector representation of training samples from any domain, we propose to generate model of normality using state-of-the-art classifiers including Gaussian Mixture Model (GMM), one class classifier based on support vector machine (OCC-SVM), and an extension of multivariate m-Mediods based classifier, as proposed in [19], to incorporate the capacity of handling class label noise. We have selected these classifiers intentionally as they have abilities to handle different types of class distributions. GMM [4] is good in handling overlapping classes and diverse distribution, but it gives poor results with nonoverlapping classes having complex and tight boundaries. On the other hand, SVM [8, 12] handles the problem of complex and tight decision boundaries in a very good manner. It does this by looking for an optimal hyperdimensional decision surfaces which should separate the samples belonging to different classes. However, the effectiveness of OCC-SVM decreases with the increase in the amount of overlap amongst the classes. Another disadvantage of OCC-SVM is that it overlooks those classes having smaller membership counts to correctly classify samples belonging to classes having large membership counts. Multivariate m-Mediod [19] classifier has the capacity to cater for the presence of multivariate distribution of samples within a modeled class. It has the strength of modeling complex patterns without imposing any limitation on the shape of distribution of samples within a given pattern. Once the multivariate m-Mediods models for all the classes have been learnt, the classification of test samples is achieved using a soft classification technique that can handle for multimodal and overlapping distributions of samples among different patterns within a dataset. It also caters for patterns having small membership count. Combining classifiers with different skills and specialties will enable the ensemble to cover all possible aspects and problems of classification.

One of the major drawbacks of all these classifiers is that their effectiveness degrades significantly with the increasing amount of class label noise in the training data. In the next section, we are presenting an extension of our previously proposed m-Mediods based classifier to mitigate the effect of class label noise in training data.

5. Modified m-Mediods Based Classifier to Filter Class Label Noise

In this section, we are presenting an extension of m-Mediods based classifier, as presented in [19], to filter class label noise whilst generating model of normality of different classes. The proposed approach is comprised of three major steps: (1) modelling of known classes using m-Mediods model, (2) filtering training samples representing class label noise using m-Mediods based model for all the classes, and (3) classification of unseen samples using proposed classifier.

5.1. Multivariate m-Mediods Based Modeling. Given a labelled training data, we propose to generate m-Mediods model that represents the normal class distribution of known classes containing N samples using a model composed of m-Mediods. Let $\mathbf{S} = \{S_1, S_2, \ldots, S_N\}$ be our training dataset containing N samples. A sample S_i in $\mathbf{S}$ is represented by a t-dimensional feature vector $S_i = [f_1, f_2, \ldots, f_t]$. Let $\mathbf{S}^{(j)}$ represent the samples belonging to class j; an algorithm to generate modified m-Mediods based model, whilst filtering samples representing class label noise, is comprised of the following steps.

(1) If the number of samples in training data is less than a threshold κ, go to step 8. The value of κ is set to a value large enough for which the application of agglomerative merging is feasible with respect to time. We assumed $\kappa = 1000$.

(2) Initialize the semifuzzy self-organizing map (SFSOM) with a number of output nodes m_{init} which is much greater than m. Setting higher values of m_{init} results in high computational complexity whereas lower values fail to mitigate the problem of local minima associated with quantization. We assumed $m_{\text{init}} = 3 * m$ based on empirical evaluation. Setting much higher values results in increasing computational complexity without having any considerable impact on modeling quality. On the other hand, setting much lower values fails to mitigate the issue of local minima typically affiliated with approaches based on quantization.

(3) Initialize weight vector representation of output nodes, referred to as W_i (where $1 \leq i \leq m_{\text{init}}$), using the probability density function $N(\mu, \Sigma)$ approximated from training samples in $\mathbf{S}^{(j)}$.

(4) Determine k nearest neighbors $\mathbf{P}$ of input training sample S_i from set of weight vectors representation of output nodes ($\mathbf{W}$) as follows:

$$\mathbf{P} = \{\mathbf{P} \in \mathbf{W} \mid \forall Q \in \mathbf{P}, R \in \mathbf{W} - \mathbf{P}, \left\|S_i, Q\right\| \leq \left\|S_i, R\right\| \wedge |\mathbf{P}| = k\}, \quad (1)$$

where $\|\cdot, \cdot\|$ is the Euclidean distance function and $|\cdot|$ is the membership count function and $k = \delta(t)$, where

$\delta(t)$ is the neighborhood size function with respect to current training iteration t.

(5) Train the proposed SFSOM network by updating weights in $\mathbf{P}$ using

$$W_c(t+1) = W_c(t) + \alpha(t)\,\zeta(j)\,(S_i - W_c(t)) \quad \forall W_c \in \mathbf{P}, \tag{2}$$

where W_c is the weight vector associated with output neuron c, j is the order of closeness of W_c to S_i ($1 \leq j \leq k$), $\alpha(t)$ is the learning rate of SFSOM with respect to current training cycle t, and $\zeta(j,k) = \exp(-(j-1)^2/2k^2)$ is a membership function which has an initial value of 1 and decreases with increasing values of j.

(6) Decrease neighborhood size $\delta(t)$ and the learning rate $\alpha(t)$ with time as follows:

$$\delta(t) = \left\lceil \delta_{\text{init}} \left(1 - e^{2(t-t_{\max})/t_{\max}}\right) \right\rceil, \qquad \alpha(t) = 1 - e^{2(t-t_{\max})/t_{\max}}, \tag{3}$$

where δ_{init} is the number of neighbors to be affected when $t = 1$ and $t_{\max}$ is the maximum number of training iterations.

(7) Iterate through steps from 4 to 6 for all the training iterations.

(8) Compute the membership of training samples by assigning them to the nearest output nodes.

(9) Identify and remove output nodes with zero membership count.

(10) Merge the closest pair of weight vectors, indexed as (a, b), using the following equation:

$$W_{ab} = \frac{|W_a| \times W_a + |W_b| \times W_b}{|W_a| + |W_b|}, \tag{4}$$

where

$$(a,b) = \arg\min_{(i,j)} \left\|W_i, W_j\right\| \times \sqrt{|W_i| + |W_j|} \quad \forall i, j \wedge i \neq j. \tag{5}$$

(11) Iterate through step 10 till the number of output nodes is equal to m. Append weight vector $\mathbf{W}$ to the list of mediods $\mathbf{M}^{(j)}$ modeling the pattern j.

(12) Approximate the density of the local distribution around each mediod by computing the mean of the distance of the mediod from its k nearest mediods. Append the average distance to $\mathbf{D}^{(j)}$ in correspondence with a given mediod in the mediods list $\mathbf{M}^{(j)}$.

5.2. Filtering of Class Label Noise. Once the m-Mediods based models of all the classes have been learnt, we apply a filtering process on the complete set of mediods to identify a subset of mediods tentatively representing samples with class label noise. This filtration process is based on the observation that the sample with class label noise will not generally be surrounded by samples belonging to the same class. Consequently, the mediod representing such samples will be surrounded by mediods representing other classes with little or no presence of mediods from the same class. The filtration algorithm to remove samples representing class label noise is composed of the following steps.

(1) Merge sets of mediods $\mathbf{M}^{(j)}$ modeling different classes j into a superset $\mathbf{M}$ as follows:

$$\mathbf{M} = \left\{\mathbf{M}^{(1)} \bigcup \mathbf{M}^{(2)} \bigcup \cdots \bigcup \mathbf{M}^{(\#_{\text{classes}})}\right\}, \tag{6}$$

where $\#_{\text{classes}}$ is the total number of classes in a given dataset.

(2) Sequentially select mediod M_i from $(\mathbf{M})$. Identify the subset of mediods from $\mathbf{M}$, referred to as $\mathbf{P}_i$ that are member of k nearest neighbors of M_i specified as follows:

$$\mathbf{P}_i = \{\mathbf{P}_i \in \mathbf{M} \mid \forall Q \in \mathbf{P}, R \in \mathbf{M} - \mathbf{P}_i, \left\|M_i, Q\right\| \leq \left\|M_i, R\right\| \wedge \left|\mathbf{P}_i\right| = k\}. \tag{7}$$

(3) Identify subset of mediods $\widetilde{\mathbf{P}}_i$ from $\mathbf{P}$ that belongs to the same class as M_i, specified as follows:

$$\widetilde{\mathbf{P}}_i = \left\{\widetilde{\mathbf{P}}_i \in \mathbf{P}_i \mid \forall Q \in \widetilde{\mathbf{P}}_i, R \in \mathbf{P}_i - \widetilde{\mathbf{P}}_i, \mathbf{\Gamma}(M_i) = \mathbf{\Gamma}(Q) \wedge \mathbf{\Gamma}(M_i) \neq \mathbf{\Gamma}(R)\right\}, \tag{8}$$

where $\mathbf{\Gamma}(\cdot)$ is the function that returns the label of a given sample or mediod.

(4) Prune mediod M_i if there are no mediods in $\widetilde{\mathbf{P}}_i$ specified as follows:

$$\mathbf{M}_{\text{noise}} = \left\{M_i \in \mathbf{M} \mid \widetilde{\mathbf{P}}_i = \{\}\right\} \quad \forall i. \tag{9}$$

(5) Filter mediods representing samples with class label noise using

$$\widehat{\mathbf{M}} = \left\{\mathbf{M} - \mathbf{M}_{\text{noise}}\right\}. \tag{10}$$

(6) Filter samples representing class label noise from training data using

$$\widehat{\mathbf{S}} = \left\{S_j \in \mathbf{S} \mid S_j \in M_i \wedge M_i \in \widehat{\mathbf{M}}\right\} \quad \forall j \tag{11}$$

$$\mathbf{P}^{(j)} = \left\{\mathbf{P}^{(j)} \in \mathbf{M}^{(j)} \mid \forall R \in \mathbf{P}^{(j)}, S \in \mathbf{M}^{(j)} - \mathbf{P}^{(j)}, \|Q, R\| \leq \|Q, S\| \wedge \left|\mathbf{P}^{(j)}\right| = k\right\} \quad \forall j. \tag{12}$$

5.3. Multivariate m-Mediods Based Classification. Once we have generated the multivariate m-Mediods based model of normal classes after mitigating the effect of class label noise, the classification of unseen samples is done by checking the closeness of feature vector representation of unseen sample from the m-Mediods models of different classes. The sample is then classified to the class with minimum distance. The proposed algorithm for multivariate m-Mediods based classification of unseen samples using learned m-Mediods model is composed of the following steps.

(1) Identify k nearest neighbors of test sample Q from m-Mediods model $\mathbf{M}^{(j)}$ separately for each class j as specified in (12).

(2) Compute the membership $\mathfrak{I}\{j\}$ of unseen lesion sample with respect to class j as follows:

$$\mathfrak{I}\{j\} = 1 - \frac{\sum_{j=1}^{k} \left\| Q, \mathbf{P}_j^{(j)} \right\| / k}{\overline{\mathbf{D}^{(j)}}}, \tag{13}$$

where $\overline{\mathbf{D}^{(j)}}$ is the average of mean distances corresponding to mediods in $\mathbf{P}^{(j)}$ as identified in (12). The mean distance corresponding to a given mediod is precomputed and stored in $\mathbf{D}^{(j)}$ as specified in step 12 of the modeling algorithm, presented in Section 5.1. The test sample is classified to the class with the highest probability ($\mathfrak{I}$).

6. Proposed Classifier Ensemble

In this section, we present a framework for learning and classification using the proposed ensemble of classifiers. Although we intend to use three classifiers as specified in Section 4, we present a generic ensemble framework that can accommodate any number of classifiers. Once the model of different classifiers has been generated, we proposed a genetic algorithm based approach to learn weights on each class for different classifiers using set of identified confused samples. An instance in the population is a weight vector W that contains weights to scale the probability/confidence of each classifier on prediction of different classes. The weight vector W can be represented as follows:

$$W = \begin{bmatrix} w_{C_1}^1, \ldots, w_{C_1}^j, \ldots, w_{C_1}^{\#_{\text{classes}}}, \\ \vdots \\ w_{C_a}^1, \ldots, w_{C_a}^j, \ldots, w_{C_a}^{\#_{\text{classes}}}, \\ \vdots \\ w_{C_\Psi}^1, \ldots, w_{C_\Psi}^2, \ldots, w_{C_\Psi}^{\#_{\text{classes}}} \end{bmatrix}, \tag{14}$$

where Ψ is the number of classifiers integrated in the proposed framework and $w_{C_a}^j$ is the weight associated with the probabilistic output of classifier C_a regarding class j. The algorithm for learning optimal weight values on the classes to improve ensemble accuracy using $\widehat{\mathbf{S}}$ comprises the following steps.

(1) Initialize the population for genetic algorithm, represented as $\mathbf{W}$, by randomly generating $\#_W$ weight vectors. Pad $\mathbf{W}$ with some predefined weights where a particular classifier gives maximum confidence to one class and no confidence for other classes.

(2) Normalize the weight vectors so that the sum of weights in each vector is equivalent to 1.

(3) Set $\text{acc}_W = 0\ \forall W \in \mathbf{W}$.

(4) Divide noise free dataset $\widehat{\mathbf{S}}$ into k-folds. Let CF_l represent the lth fold of $\widehat{\mathbf{S}}$ where $l = 1, \ldots, k$.

(5) Select the lth fold and treat it as cross validation set $\widehat{\mathbf{S}}_{\text{CV}}^l = CF_l$. Initialize training set $\widehat{\mathbf{S}}_{\text{train}}^l = \{\}$. The training set to learn classifier model is then obtained as follows:

$$\widehat{\mathbf{S}}_{\text{train}}^l = \left\{ \widehat{\mathbf{S}}_{\text{train}}^l \bigcup CF_j \right\} \quad \forall j \wedge j \neq l. \tag{15}$$

(6) Learn the individual classifier models such as multivariate m-Mediods, GMM, and OCC-SVM using $\widehat{\mathbf{S}}_{\text{train}}^l$.

(7) Select a sample S_i from $\widehat{\mathbf{S}}_{\text{CV}}^l$ and compute its probability to be classified to different classes using various classifiers as follows:

$$\text{class}_{S_i}^{C_a} = \arg\max_{\forall j} \text{Prob}_{C_a}\left(y = \text{class}_j \mid S_i\right), \tag{16}$$

where $\text{Prob}_{C_a}(y = \text{class}_j \mid S_i)$ is the probability of a given sample S_i to be classified to class j according to classifier C_a.

(8) Identify those samples for which at least two classifiers give different class predictions. This is achieved by pruning those samples from $\widehat{\mathbf{S}}_{\text{CV}}^l$ for which all the concerned classifiers predict the same class as these will not contribute in weight learning process. Removing such samples will have a positive impact of significantly speeding up the weight learning process on classes. More formally, let pruned cross validation set, containing confused samples, be referred to as $\widetilde{S_{\text{CV}}^l}$. Set $\widetilde{S_{\text{CV}}^i} = \{\}$. The filtered cross validation set containing confused samples corresponding to lth fold is obtained as specified in

$$\widetilde{\mathbf{S}_{\text{CV}}^l} = \widetilde{\mathbf{S}_{\text{CV}}^l} \bigcup \left\{ S \in \widehat{\mathbf{S}}_{\text{CV}}^l \mid \text{class}_S^{C_p} \neq \text{class}_S^{C_q}\ \forall p, q \wedge p \neq q \right\} \quad \forall S \in \widehat{\mathbf{S}}_{\text{CV}}^l. \tag{17}$$

(9) Repeat steps 5–8 for all the folds.

(10) Select a sample S from the set of confused samples corresponding to ith fold ($\widetilde{S_{\text{CV}}^i}$) and compute its probabilities to belong to various classes using different

classifiers learnt using $\widehat{S^i}_{\text{train}}$. Combine these probabilities given by different classifiers in an ensemble using a weight vector $W \in \mathbf{W}$ as follows:

$$\text{Prob}^j_{\text{ens}} = \sum_{\forall l} \omega_{C_l,j} * \text{Prob}_{C_l}\left(y = \text{class}_j \mid S\right) \quad \forall j, \tag{18}$$

where $\omega_{C_l,j} \in W$ is the weight assigned to the probabilistic output of sample S to belong to class j according to classifier C_l and $\text{Prob}^j_{\text{ens}}$ is the probability of sample S to belong to class j according to classifier ensemble created using weight vector W. Classify the sample using

$$\text{class}_S^{C_{\text{ens}}} = \arg\max_{\forall j} \text{Prob}_{C_{\text{ens}}}\left(y = \text{class}_j \mid S\right). \tag{19}$$

(11) Increment acc_W by 1 if predicted class $\text{class}_S^{C_{\text{ens}}}$ is equal to the true label of sample S.

(12) Iterate through steps 10-11 for all the confused samples in $\widetilde{\mathbf{S}^i_{\text{CV}}}$.

(13) Iterate steps 10–12 for all the folds in the dataset.

(14) Repeat steps 10–13 $\forall W \in \mathbf{W}$ and compute their corresponding accuracies acc_W. The classification accuracies of ensemble computed using different weight vectors in $\mathbf{W}$ are the objective function that we want to optimize using genetic algorithm.

(15) Generate new sets of weight vectors $\mathbf{W}_{\text{new}}$ by selecting 10 best weight vectors from $\mathbf{W}$ with respect to their objective function values and applying the genetic operators, that is, crossover and mutation. The objective function using $\mathbf{W}_{\text{new}}$ is computed as specified in steps 10–14. The evolved population of best 20 weight vectors is obtained by selecting the top weight vectors from $\{\mathbf{W} \bigcup \mathbf{W}_{\text{new}}\}$. This step prevents us from losing track of any weight vector that gives us the optimal results during the weight learning procedure whilst filtering the newly evolved but poor population of weight vectors.

(16) Iterate through steps 10–15 till there is no improvement in the optimal classification accuracy of the ensemble for 5 consecutive iterations or the number of iterations over the genetic algorithm exceeds a certain threshold. Select the weight vector W_{opt} that yields best accuracy for the learned ensemble.

Once the weight vector to combine classifiers in an ensemble is learned, the classification of test sample Q using proposed classifier ensemble is performed as follows:

$$\text{class}_Q^{C_{\text{ens}}} = \arg\max_{\forall \text{class}_j} \sum_{\forall a} \omega_{C_a,j} * \text{Prob}_{C_a}\left(y = \text{class}_j \mid Q\right) \quad \forall j, \tag{20}$$

where $\omega_{C_a,j} \in W_{\text{opt}}$.

7. Experiments

In this section, we present different experiments that are performed to show the effectiveness of proposed approach as compared to the competitors.

7.1. Datasets. To evaluate the performance of our proposed classifier ensemble methodology, four real life datasets have been used including IRIS, Satimage, DiaretDB, German, Pima Indian Diabetes, and Heart datasets. A brief overview of these datasets is given in Table 1. The distribution of instances in different classes within a dataset is presented in Table 2 to highlight the variation in distribution of samples among the classes.

7.2. Experiment 1: Evaluation of Proposed Classifier Ensemble Approach. The purpose of this experiment is to assess the effectiveness of our proposed ensemble approach as compared to individual classifiers in the ensemble. The experiment is conducted on DiaretDB dataset. We have extracted 70% of the instances separately from each class and treated it as a training data. The remaining 30% of the samples from each class are treated as test data. We employ k-fold cross validation with $k = 7$ to learn the classifier models individually in an ensemble as specified in Section 6. We managed to extract only 159 confused samples out of 2156 samples present in training DiaretDB dataset. Selection of only confused samples from the train dataset speeds up the weight learning process on all individual classes present in the dataset. Learning of optimal weights on classes for the proposed ensemble is done as specified in Section 6. The classification accuracies of classifier ensemble approach and its individual classifier members, using test dataset, are presented in Table 3. It is observed that hybrid classifier yields the best accuracy, that is, 99.675%, as compared to its individual member classifiers which shows the effectiveness of our proposed classifier ensemble as compared to individual classifiers.

7.3. Experiment 2: Comparison of Proposed Classifier Ensemble Approach with Competitors. This experiment is conducted to compare the performance of proposed ensemble method as compared to the competitors including Adaboost, Bagging, and Random Subspace Method (RSM). We have performed the experiment using DiaretDB, IRIS, Satellite Image, Heart, Pima Indian Diabetes, and German datasets. The experimental setup is similar to the one specified in Experiment 1. The competitive approaches require the manual specification of number of iterations used for training the ensemble. We use different number of iterations, that is, 50, 100, and 150 iterations, to learn the competitive learning approaches. The classification accuracies of competitors for variety of real life datasets, using 50, 100, and 150 iterations, are reported in Table 4. We can observe from the table that the performance of competitors is affected by changing the number of iterations. Adaboost shows more sensitivity toward the selection of the number of iterations followed by Bagging and RSM. We select the best classification accuracies of

Table 1: Overview of datasets used for experimental evaluation.

Dataset	Description	Number of samples	Number of features	Number of classes
IRIS	A flower dataset [59] used as a test bed for comparison of classification techniques.	150	4	3
DiaretDB	A lesion image dataset [51] used for diabetic retinopathy detection methods. These images are processed by [60, 61] to extract feature vector representation of lesions.	3080	15	4
Satimage	A publicly available Satellite Images dataset [59] generated from Landsat scanner image data.	6435	36	6
Heart	A good dataset [59] to test the ML algorithms. In this dataset, each patient is classified as normal and abnormal.	267	23	2
Pima Indian diabetes	It is provided by National Institute of Diabetes and Digestive and Kidney diseases [59]. It is a dataset of all female patients, at least 21 years old.	768	8	2
German	This is a German credit dataset [59].	1000	20	2

Table 2: Description of number of instances present in each class of datasets.

Dataset	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6
IRIS	50	50	50			
DiaretDB	1028	725	1262	65		
Satimage	1533	703	1358	626	707	1508
Heart	55	212				
Pima Indian Diabetes	500	268				
German	300	700				

Table 3: Classification accuracies of proposed classifier ensemble as compared to individual classifiers using DiaretDB dataset.

Method	% accuracy
Proposed classifier ensemble	**99.675**
Modified m-Mediods	97.62
GMM	98.59
OCC-SVM	93.72

different competitors with respect to different number of iterations (represented in bold in Table 4) and used them to compare it with the classification accuracies obtained using our proposed ensembling approach. The results are presented in Table 5. As obvious from Table 5, the proposed approach gives the best classification accuracy results as compared to the competitors. Our approach is also independent of the manual specification of number of iterations.

7.4. Experiment 3: Comparing Proposed Ensemble Approach with Competitors in the Presence of Noise. The purpose of this experiment is to study the sensitivity of proposed ensemble approach and its competitors in the presence of class label noise. Class label noise is induced by randomly changing the label information of certain number of samples in training data with wrong labels. To simulate different noise level, we simulated noise in 5% and 10% of samples in training data. The remaining setup of the experiment is the same as specified in Experiment 2. The comparison of proposed approach with competitors using DiaretDB, IRIS, Satimage, Pima Indian Diabetes, and German datasets with class label noise is presented in Table 6. Based on these results and their comparison with performance of various approaches without the presence of class label noise, as presented in Table 5, we can see that the performance of Adaboost is affected by the presence of class label noise specially for Satimage and German data. With DiaretDB, the performance of Adaboost goes down for 10% noise level. Bagging is relatively less sensitive to noise than Adaboost. RSM also shows sensitivity to various noise levels using different dataset. However, it can be observed from Table 6 that our proposed ensemble approach shows the least sensitivity toward class label noise as compared to its competitors. Filtering class label noise using proposed extension of m-Mediods based modeling approach has significantly mitigated the effect of class label noise. Resultantly, the proposed approach gives more accurate results as compared to competitors in the presence of class label noise.

7.5. Experiment 4: Analyzing Sensitivity of Ensembling Approaches to Class Imbalance in Datasets. The performance of proposed ensemble approach in the presence of class imbalance problem is evaluated and compared in this experiment. The experimental setup is similar to the one specified in Experiment 2. The confusion matrices of competitors and proposed approach using DiaretDB, Satimage, Heart,

Table 4: Classification accuracies of competitors using different number of training iterations.

Dataset	Adaboost			Bagging			RSM		
	Iter = 50	Iter = 100	Iter = 150	Iter = 50	Iter = 100	Iter = 150	Iter = 50	Iter = 100	Iter = 150
IRIS	**100**	97.8	97.8	**100**	97.8	100	**100**	**100**	**100**
DiaretDB	91.92	**92.35**	**92.35**	97.68	**97.9**	97.68	**97.24**	97.1	97
Satimage	76.1	77.1	**78.1**	91.25	91.5	**91.7**	**90.99**	90	90
Heart	**67.5**	**67.5**	66.3	62.5	62.5	**63.8**	**68.8**	**68.8**	**68.8**
Pima Indian Diabetes	75.37	74.08	**76.53**	77.48	77.51	**78.01**	76.34	**77.6**	76.34
German	72	**72.33**	71.1	**73.1**	72.4	72.67	**72.7**	72.3	72

Table 5: Classification accuracies of competitors using different number of training iterations.

Dataset	Adaboost	Bagging	RSM	Proposed approach
IRIS	**100**	**100**	**100**	**100**
DiaretDB	92.35	97.9	97.24	**99.675**
Satimage	78.1	91.7	90.99	**94.05**
Heart	67.5	63.8	68.8	**70.27**
Pima Indian Diabetes	76.53	78.01	77.6	**78.26**
German	72.33	73.1	72.7	**74.7**

Table 6: Comparison based on classification accuracies of proposed approach as compared to competitors in the presence of different levels of class label noise.

Dataset	Adaboost		Bagging		RSM		Proposed approach	
	5% noise	10% noise	5% noise	10% noise	5% noise	10% noise	5% noise	10% noise
IRIS	**98**	**97.8**	95.6	95.6	97.6	97.6	**98**	**97.8**
DiaretDB	90.95	88.6	97.2	96.1	95.9	94.1	**97.9**	**97.2**
Satimage	75.5	72.2	91.2	90.8	90.5	90.3	**93.4**	**92.6**
Pima Indian Diabetes	75.4	74.9	77.3	76.1	76.7	74.5	**78**	**77.7**
German	71.58	69.9	72.6	71.3	69.9	68.4	**73.6**	**72.8**

Pima Indian Diabetes, and German datasets are presented in Tables 7, 8, 9, 10, and 11, respectively. Looking at the confusion matrices obtained using different ensembling approaches across variety of datasets, it is obvious that Adaboost is performing poorly on class imbalance problem. Adaboost attempts to increase overall classification accuracy by focusing on classes with large membership count and ignoring those with lesser memberships. This is obvious from results for class 3 and class 4 for DiaretDB dataset, as presented in Table 7(a). Similarly, we can observe the same phenomena with classes 3, 4, and 6 of Satimage dataset as presented in Table 8(a). Classes 3 and 6 have large number of samples and hence they have overshadowed class 4 which has resulted in misclassification of many class 4 samples to class 3 and class 6. Similarly, the same situation is observed with Pima Indian Diabetes and German datasets as all of these binary class datasets have imbalanced number of instances in both classes. It is observed that there is a large difference between both classes of each of these two datasets. From Table 10(a) it can be observed that Adaboost performs well on the class having large membership count such as class 1, but its performance goes down on the class comprising small number of instances. It also shows poor performance on class 1 of German dataset having very small membership count as presented in Table 11(a). On the other hand, Bagging and RSM perform better than Adaboost but are still affected by the presence of class imbalance issue. The reason behind this behavior of competitors is that they focus on improving their overall classification accuracy without considering individual classes irrespective of their membership counts. As compared to competitors, proposed ensemble approach based on weight learning on classes is performing well on DiaretDB, Satimage, Heart, Pima Indian Diabetes, and German datasets as shown in Tables 7(d), 8(d), 9(d), 10(d), and 11(d), respectively.

7.6. Discussion. In this section, we presented a variety of experiments to demonstrate the superiority of proposed approach as compared to competitors in the presence of real world problems including presence of different level of class label noise class imbalance problem. The experimental results on different datasets obtained from variety of domains are presented. It has been observed that the proposed approach performs better than the existing state-of-the-art approaches such as Adaboost, RSM, and Bagging. The results presented in Table 4 demonstrated that the proposed approach gives the best classification accuracy results as compared to the competitors without imposing any requirement of manual

Table 7: Confusion matrices of (a) Adaboost, (b) Bagging, (c) RSM, and (d) proposed approach using DiaretDB dataset.

(a)

	$C1$	$C2$	$C3$	$C4$
$C1$	284	14	10	0
$C2$	13	203	2	0
$C3$	2	10	367	0
$C4$	0	2	17	0

(b)

	$C1$	$C2$	$C3$	$C4$
$C1$	307	0	1	0
$C2$	2	215	1	0
$C3$	2	3	374	0
$C4$	0	2	8	9

(c)

	$C1$	$C2$	$C3$	$C4$
$C1$	305	0	3	0
$C2$	5	212	1	0
$C3$	1	0	378	0
$C4$	0	2	13	4

(d)

	$C1$	$C2$	$C3$	$C4$
$C1$	306	1	1	0
$C2$	2	217	1	0
$C3$	0	0	379	0
$C4$	0	0	0	19

Table 8: Confusion matrices of (a) Adaboost, (b) Bagging, (c) RSM, and (d) proposed approach using Satimage dataset.

(a)

	$C1$	$C2$	$C3$	$C4$	$C5$	$C6$
$C1$	363	0	22	2	4	69
$C2$	13	177	0	2	13	6
$C3$	2	0	394	5	0	6
$C4$	4	0	60	14	0	110
$C5$	30	2	1	3	137	39
$C6$	1	0	17	11	2	421

(b)

	$C1$	$C2$	$C3$	$C4$	$C5$	$C6$
$C1$	446	1	9	0	4	0
$C2$	0	206	0	3	1	1
$C3$	2	2	391	10	0	2
$C4$	2	1	34	120	1	30
$C5$	9	2	0	0	188	13
$C6$	0	2	6	21	5	418

(c)

	$C1$	$C2$	$C3$	$C4$	$C5$	$C6$
$C1$	450	1	9	0	0	0
$C2$	0	255	0	2	3	1
$C3$	1	3	390	8	0	5
$C4$	1	1	34	120	1	30
$C5$	13	2	0	1	177	19
$C6$	0	1	7	27	4	413

(d)

	$C1$	$C2$	$C3$	$C4$	$C5$	$C6$
$C1$	448	2	10	0	0	0
$C2$	1	204	0	1	5	0
$C3$	1	0	393	8	4	1
$C4$	1	0	12	153	14	8
$C5$	1	0	1	6	202	2
$C6$	0	1	3	11	23	414

specification of number of iterations. The proposed framework for combining classifiers is also robust to the presence of different level of class label noises. The proposed approach is least affected by the presence of noise as highlighted in Table 6. Robustness of proposed approach to the presence of class imbalance problem is highlighted in Experiment 4. This superiority of proposed approach, as compared to competitors, is attributed to the selection of classifiers with different specialties to handle various types of distribution that is expected within a dataset. Learning weights on these classifiers at the class level will result in having major contribution of decision making from a classifier that suits the localized probability distribution. This results in significant reduction of false positives and false negatives as compared to the competitors.

8. Conclusion

In this paper, we have discussed the issue of combining classifiers in an ensemble to enhance the overall classification accuracies. A novel classifier ensemble framework is presented that combines heterogeneous classifiers while handling the problems of class label noise and class imbalance problem. An extension of m-Mediods based approach is presented to handle and filter samples in training data that are expected to have incorrect labels. The filtered data is then used to learn the remaining classifier models. These individual models are further combined by introducing weight learning method to learn weights on classes for each individual classifier to construct an ensemble. A genetic algorithm based approach is also presented that quickly searches for an optimal set of weights at class level for different classifiers. The weighted classifier ensemble is then used to classify unseen samples to one of the known classes.

Experiments are conducted to compare the performance of proposed classifier ensemble framework with existing state-of-the-art approaches such as Adaboost, Bagging, and RSM. It has been shown that the proposed classifier ensemble gives better classification accuracies, as compared to the competitors, on variety of datasets selected from different

Table 9: Confusion matrices of (a) Adaboost, (b) Bagging, (c) RSM, and (d) proposed approach using Heart dataset.

(a)

	$C1$	$C2$
$C1$	16	17
$C2$	9	38

(b)

	$C1$	$C2$
$C1$	12	21
$C2$	8	39

(c)

	$C1$	$C2$
$C1$	13	20
$C2$	5	42

(d)

	$C1$	$C2$
$C1$	10	23
$C2$	1	46

Table 10: Confusion matrices of (a) Adaboost, (b) Bagging, (c) RSM, and (d) proposed approach using Pima Indian Diabetes dataset.

(a)

	$C1$	$C2$
$C1$	128	22
$C2$	51	29

(b)

	$C1$	$C2$
$C1$	135	15
$C2$	39	41

(c)

	$C1$	$C2$
$C1$	131	19
$C2$	38	42

(d)

	$C1$	$C2$
$C1$	138	12
$C2$	27	53

domains. Experimental results to show the robustness of proposed approach to class label noise are also presented in Section 7.4. The proposed approach gives better classification results than competitors in the presence of various levels of class label noise as obvious from Table 6. The sensitivity of proposed approach, as compared to competitors, in the presence of class imbalance problem is also analyzed.

Table 11: Confusion matrices of (a) Adaboost, (b) Bagging, (c) RSM, and (d) proposed approach using German dataset.

(a)

	$C1$	$C2$
$C1$	34	56
$C2$	27	183

(b)

	$C1$	$C2$
$C1$	28	62
$C2$	5	205

(c)

	$C1$	$C2$
$C1$	28	62
$C2$	8	202

(d)

	$C1$	$C2$
$C1$	39	51
$C2$	3	207

The proposed approach shows the least sensitiveness to class imbalance issue as demonstrated in Tables 7–9, respectively.

Conflict of Interests

The authors do not have a direct financial relation that might lead to a conflict of interests for any of the authors.

Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2013R1A1A2061978).

References

[1] W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan, and S. Maybank, "A system for learning statistical motion patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 9, pp. 1450–1464, 2006.

[2] W. Hu, D. Xie, Z. Fu, W. Zeng, and S. Maybank, "Semantic-based surveillance video retrieval," *IEEE Transactions on Image Processing*, vol. 16, no. 4, pp. 1168–1181, 2007.

[3] S. Khalid and A. Naftel, "Classifying spatiotemporal object trajectories using unsupervised learning in the coefficient feature space," *Multimedia Systems*, vol. 12, no. 3, pp. 227–238, 2006.

[4] S. Bertrand, *Gaussian Mixture Model Classifiers*, 2007.

[5] T. Brotherton, T. Johnson, and G. Chadderdon, "Classification and novelty detection using linear models and a class dependent-elliptical basis function neural network," in *Proceedings of the IEEE International Joint Conference on Neural Networks*, pp. 876–879, Anchorage, Alaska, USA, May 1998.

[6] S. Roberts and L. Tarassenko, "A probabilistic resource allocating network for novelty detection," *Neural Computation*, vol. 6, no. 2, pp. 270–284, 1994.

[7] D. Y. Yeung and C. Chow, "Parzen window network intrusion detectors," in *Proceedings of the 16th IEEE International Conference on Pattern Recognition*, pp. 385–388, Alberta, Canada, 2002.

[8] M. D. Hanif and P. S. Yashwant, "Cybercrime detection te chniques based on support vector machines," *Journal of Artificial Intelligence Research*, vol. 2, no. 1, pp. 1–11, 2013.

[9] H. Zhang, A. C. Berg, M. Maire, and J. Malik, "SVM-KNN: discriminative nearest neighbor classification for visual category recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '06)*, pp. 2126–2136, June 2006.

[10] G. Rätsch, S. Mika, B. Schölkopf, and K.-R. Müller, "Constructing boosting algorithms from SVMs: an application to one-class classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 9, pp. 1184–1199, 2002.

[11] C. P. Diehl and J. B. Hampshire II, "Real-time object classification and novelty detection for collaborative video surveillance," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '02)*, vol. 3, pp. 2620–2625, Honolulu, Hawaii, USA, May 2002.

[12] D. M. J. Tax, *One-class classification [Ph.D. thesis]*, Delft University of Technology, 2001.

[13] S. Khalid and S. Razzaq, "Frameworks for multivariate *m*-mediods based modeling and classification in Euclidean and general feature spaces," *Pattern Recognition*, vol. 45, no. 3, pp. 1092–1103, 2012.

[14] J. Owens and A. Hunter, "Application of the self-organising map for trajectory classification," in *Proceedings of 3rd IEEE International Workshop on Visual Surveillance*, Dublin, Ireland, 2000.

[15] S. Marsland, J. Shapiro, and U. Nehmzow, "A self-organising network that grows when required," *Neural Networks*, vol. 15, no. 8-9, pp. 1041–1058, 2002.

[16] S. Khalid, "Motion-based behaviour learning, profiling and classification in the presence of anomalies," *Pattern Recognition*, vol. 43, no. 1, pp. 173–186, 2010.

[17] S. Khalid, "Activity classification and anomaly detection using m-mediods based modelling of motion patterns," *Pattern Recognition*, vol. 43, no. 10, pp. 3636–3647, 2010.

[18] M. Usman Akram, S. Khalid, A. Tariq, and M. Younus Javed, "Detection of neovascularization in retinal images using multivariate *m*-Mediods based classifier," *Computerized Medical Imaging and Graphics*, vol. 37, no. 5, pp. 346–357, 2013.

[19] S. Khalid and S. Razzaq, "Frameworks for multivariate m-mediods based modeling and classification in Euclidean and general feature spaces," *Pattern Recognition*, vol. 45, no. 3, pp. 1092–1103, 2012.

[20] N. García-Pedrajas, C. García-Osorio, and C. Fyfe, "Nonlinear boosting projections for ensemble construction," *The Journal of Machine Learning Research*, vol. 8, pp. 1–33, 2007.

[21] C.-X. Zhang and J. Zhang, "A novel method for constructing ensemble classifiers," *Journal of Statistics and Computing*, vol. 19, no. 3, pp. 317–327, 2009.

[22] Y. Zhu, J. Ou, G. Chen, and H. Yu, "An approach for dynamic weighting ensemble classifiers based on cross-validation," *Journal of Computational Information Systems*, vol. 6, no. 1, pp. 297–305, 2010.

[23] T. K. C. Neo and D. Ventura, "A direct boosting algorithm for the k-nearest neighbor classifier via local warping of the distance metric," *Pattern Recognition Letters*, vol. 33, no. 1, pp. 92–102, 2012.

[24] S. Kotsiantis, "Combining bagging, boosting, rotation forest and random subspace methods," *Journal of Artificial Intelligence Review*, vol. 35, no. 3, pp. 223–240, 2011.

[25] A. Al-Ani and M. Deriche, "A new technique for combining multiple classifiers using the Dempster-Shafer theory of evidence," *Journal of Artificial Intelligence Research*, vol. 17, pp. 333–361, 2002.

[26] N. García-Pedrajas and C. García-Osorio, "Constructing ensembles of classifiers using supervised projection methods based on misclassified instances," *Expert Systems with Applications*, vol. 38, no. 1, pp. 343–359, 2011.

[27] Z. Zhang and P. Yang, "An ensemble of classifiers with genetic algorithm based feature selection," *The IEEE Intelligent Informatics Bulletin*, vol. 9, no. 1, pp. 18–24, 2008.

[28] A. H. R. Ko, R. Sabourin, and A. S. Britto, Jr., "From dynamic classifier selection to dynamic ensemble selection," *Pattern Recognition*, vol. 41, no. 5, pp. 1735–1748, 2008.

[29] H. Kim, H. Moon, and H. Ahn, "A weight-adjusted voting algorithm for ensembles of classifiers," *Journal of the Korean Statistical Society*, vol. 40, no. 4, pp. 437–449, 2011.

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

[31] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," *Proceedings of the 13th International Machine Learning*, pp. 325–332, 1996.

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

[33] L. Xu, A. Krzyzak, and C. Y. Suen, "Methods of combining multiple classifiers and their applications to handwriting recognition," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 22, no. 3, pp. 418–435, 1992.

[34] L. I. Kuncheva, M. Skurichina, and R. P. W. Duin, "An experimental study on diversity for bagging and boosting with linear classifiers," *Information Fusion*, vol. 3, no. 4, pp. 245–258, 2002.

[35] C.-X. Zhang and J.-S. Zhang, "A local boosting algorithm for solving classification problems," *Computational Statistics & Data Analysis*, vol. 52, no. 4, pp. 1928–1941, 2008.

[36] K. Shehzad and A. Sannia, "Framework for constructing hybrid classifier using weight learning to combine heterogeneous classifiers," in *Proceedings of the IEEE 5th International Conference on Computational Intelligence*, Seol, South Korea, September 2013.

[37] K. Shehzad and A. Sannia, "A robust ensemble based approach to combine heterogeneous classifiers in the presence of class label noise," in *Proceedings of the IEEE 5th International Conference on Computati onal Intelligence, Modeling and Simulation*, Seol , South Korea, September 2013.

[38] N. M. Elham and S. Hedieh, "Increasing classifier ensemble efficiency using KSBC algorithm," *International Journal of Computer Applications*, vol. 70, no. 13, pp. 43–50, 2013.

[39] N. Hatami and R. Ebrahimpour, "Combining multiple classifiers: diversify with boosting and combining by stacking," *International Journal of Computer Science and Network Security*, vol. 7, no. 1, pp. 127–131, 2007.

[40] 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.

[41] P. Pance and D. Saso, "Combining bagging and random subspaces to create better ensembles," *Proceedings of the 7th International Intelligent data analysis*, pp. 118–129, 2007.

[42] S. B. Kotsiantis and P. E. Pintelas, "Combining bagging and boosting," *International Journal of Computational Intelligence*, vol. 1, no. 4, 2004.

[43] H. Parvin, B. Minaei-Bidgoli, and A. Beigi, "A new classifier ensembles framework," in *Knowledge-Based and Intelligent Information and Engineering Systems*, vol. 6881 of *Lecture Notes in Computer Science*, pp. 110–119, Springer, Berlin, Germany, 2011.

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

[45] A. Arjun and R. P. Arora, "Injecting non-uniformity into Adaboost for intensive production on noisy data," *International Journal of Engineering Research and Technology*, vol. 2, no. 10, pp. 419–422, 2013.

[46] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[47] X.-Y. Liu, J. Wu, and Z.-H. Zhou, "Exploratory undersampling for class-imbalance learning," *IEEE Transactions on Systems, Man, and Cybernetics B: Cybernetics*, vol. 39, no. 2, pp. 539–550, 2009.

[48] T. H. Ryan and V. C. Nitesh, "Generating diverse ens emble to counter the problem of class imbalance," in *Proceeding of the 14th Paci fic-Asia Conference on Advances in Knowledge Discovery and Data Mining*, vol. 2, pp. 488–499, 2010.

[49] Y. Zhang and D. Wang, "A cost-sensitive ensemble method for class-imbalanced datasets," *Abstract and Applied Analysis*, vol. 2013, Article ID 196256, 6 pages, 2013.

[50] H. Parvin, B. Minaei-Bidgoli, and H. Shahpar, "Classifier selection by clustering," in *Proceedings of the 3rd Mexican Conference on Pattern Recognition*, pp. 60–66, 2011.

[51] L. Rokach, "Ensemble-based classifiers," *Artificial Intelligence Review*, vol. 33, no. 1-2, pp. 1–39, 2010.

[52] E. Bauer and R. Kohavi, "Empirical comparison of voting classification algorithms: bagging, boosting, and variants," *Journal of Machine Learning*, vol. 36, no. 1, pp. 105–139, 1999.

[53] S. Merler, B. Caprile, and C. Furlanello, "Parallelizing AdaBoost by weights dynamics," *Computational Statistics & Data Analysis*, vol. 51, no. 5, pp. 2487–2498, 2007.

[54] M. Prem, S. Nishit, M. Lilyana, and R. J. Mooney, "Experiments on ensembles with missing and noisy data," in *Proceedings of the 5th International Workshop on Multiple Classifier Systems*, vol. 3077 of *Lecture Notes in Computer Science*, pp. 293–302, Cagliari, Italy, 2004.

[55] T. G. Dietterich, "Experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization," *Machine Learning*, vol. 40, no. 2, pp. 139–157, 2000.

[56] J. R. Quinlan, "Bagging, boosting, and C4.5," in *Proceedings of the 13th National Conference on Artificial Intelligence*, pp. 725–730, Portland, Ore, USA, August 1996.

[57] Z. Zhou, J. Wu, and W. Tang, "Ensembling neural networks: many could be better than all," *Artificial Intelligence*, vol. 137, no. 1-2, pp. 239–263, 2002.

[58] Y. Pengyi, L. Wei, Z. B. Bing, C. Sanjay, and Y. Z. Albert, "Ensemble based Wrapper methods for feature selection and class imbalance learning," *Advances in Knowledge Discovery and Data Mining*, vol. 7818, pp. 544–555, 2013.

[59] UCI Machine Learning Data Repository, http://archive.ics.uci.edu/ml/datasets.

[60] U. Akram, S. Khalid, A. Tariq, and S. Khan, "Detection and classification of retinal lesions for grading of diabetic retinopathy," *Computers in Biology and Medicine*, vol. 45, pp. 161–171, 2014.

[61] M. U. Akram, S. Khalid, and S. A. Khan, "Identification and classification of microaneurysms for early detection of diabetic retinopathy," *Pattern Recognition*, vol. 46, no. 1, pp. 107–116, 2013.