A Weighted Ensemble based Active Learning model to label Microarray Data

Report submitted in partial fulfilment of the requirement for the award of the degree of

Bachelor of Computer Science and Engineering
In the Faculty of Engineering and Technology

Jadavpur University

By

Rajonya De, Roll No. 001610501031

Anuran Chakraborty, Roll No. 001610501020

Under the guidance of

Dr. Ram Sarkar

Department of Computer Science and Engineering

Jadavpur University
Kolkata – 700 032
2020

ACKNOWLEDGEMENT

We are indebted to our project guide, **Dr. Ram Sarkar** for his constant support, prompt help, support and unwavering encouragement which helped us a lot in completing the project. Under his guidance, we have been able to learn many new concepts regarding the topic at hand.

We are also thankful to **Prof. Mahantapas Kundu**, Head of the Department of Computer Science and Engineering, Jadavpur University for allowing us to carry out research in the department.

Rajonya De

Anuran Chakraborty

ABSTRACT

Classification of cancerous genes from microarray data is an important research area in bioinformatics. Even though, large amount of microarray data is available, it is very costly to label them. This paper proposes an Active Learning model, a semi-supervised classification approach, to label the microarray data using which predictions can be made even with lesser amount of labeled data, thereby reducing the cost of labeling data. Initially, a pool of unlabeled instances is given from which some instances are randomly chosen for labeling. Successive selection of instances to be labeled from unlabeled pool is determined by selection algorithms. The proposed method is devised following an ensemble approach to combine the decisions of three classifiers in order to arrive at a consensus which provides a more accurate prediction of the class label to ensure that each individual classifier learns in an uncorrelated manner. Our method combines the heuristic techniques used by an Active Learning algorithm to choose training samples with the multiple learning paradigm attained by an ensemble to optimize the search space by choosing efficiently from an already sparse learning pool. On evaluating the proposed method on 6 microarray datasets, we achieve performance which is comparable to state-of-the-art methods.

Keywords: *Active Learning, classifier ensemble, Gene Expression, Cancer classification.*

____

Dr. Ram Sarkar
Project Guide

1. Introduction

Machine learning algorithms can be broadly classified into two categories namely supervised learning algorithms and unsupervised learning algorithms. Supervised learning algorithms involve classifying samples based on their class labels. The success of any supervised learning model depends on the availability of large amount of labeled data. In traditional supervised learning algorithm, the system receives all the training samples and develops a model using the same. The motivation behind developing an Active Learning framework seeds from this very ground. In the modern world, cheap, unlabeled data are plenty but obtaining their labels is generally costly. An active learner, unlike its passive counterpart, not only trains its model but also interacts with the unlabeled data to choose the most optimal samples which make the model more robust in terms of classification ability, updating the model with every such interaction. An active learner does not deal with a fixed size training data, increasing the same at every iteration of the algorithm. It is assumed that this freedom reduces the learner's need for large quantities of labeled data.

Active Learning can be broadly classified into two groups: stream-based learning [1][2] and pool-based learning [3][4]. The former involves scanning each sample sequentially, whereas in the latter, the entire pool is ranked, followed by selection of candidate samples. The ranking of the samples in the pool plays an important role, as incorrect ranking could make the model prone to erroneous outcome. Serial query based pool learning [5] is another approach where the model is updated post every query. Single-classifier [6] as well as ensemble of classifiers [7] has been tuned to perform Active Learning. In the past, Active Learning has been applied in a wide range of domains ranging from image classification and retrieval [8], text classification [9], drug discovery [10] to cancer diagnosis [11] to name a few.

To that avail, traditional machine learning algorithms have been potently used in bioinformatics and medical imaging [12]. An estimate of 1.7 million new cancer cases have been diagnosed in 2018 in USA itself. A continuous mutation in the normal cell causes damage to the Deoxyribonucleic acid (DNA). The impairment of cell replication is also one of the main causes of forming malignant tumor cells [13]. Microarray is one of the breakthroughs in experimental molecular biology that allows us to monitor the expression levels of tens of thousands of genes simultaneously. This, in turn, allows researchers to investigate issues in cancer cell identification [14][15] that were once thought to be intractable.

DNA microarray data are formed using a glass slide on which a large number of genes are arranged. On activating a single gene, the cellular machinery begins to capture a little segment of the gene known as messenger RNA (mRNA) [16]. This mRNA is complementary in nature and so, binds to the original part of the DNA from which it was copied. In order to find out which genes are turned on and off in a given cell, the mRNA molecules present in the cell are collected. Subsequent labeling of each mRNA molecule by transcriptase enzyme produces a complementary cDNA and mRNA. Fluorescent nucleotides are adhered with the cDNA under this process and the normal and tumor samples are marked with the help of distinct fluorescent dyes. The marked cDNAs are then placed onto a DNA microarray slide and the fluorescent intensity for each spot on the slide is measured. If a specific gene is effective, it generates many mRNAs. Thus, more labeled cDNAs, which hybridized to the DNA on the slide, produce a glittering fluorescent area. Genes which are not so active produce fewer mRNAs with lesser labeled cDNAs that cause dull fluorescent spot. The absence of fluorescence shows that the gene is inactive as none of the mRNA has been hybridized to the DNA. A red spot signifies that the gene is more expressed in cancer than in normal, while a green spot signifies that the gene is more expressed in normal tissue. Yellow spot indicates that the gene is equally expressed in both. Microarray technology has thus empowered researchers to extensively explore the genetic causes of anomalies in functioning of the human body and as a result classification of microarray data [17] and feature selection [18] have been incessantly explored. But these methods entail a cost as these kinds of data are not only rare in number due to the sheer sparsity of samples, but also very expensive to label. As a result, all machine learning algorithms have a large-scale dependency on human intrinsic labor.

Keeping the above facts in mind, it can be said that microarray data classification is one such field where the application of Active Learning could be deemed successful. Liu et al. [11] have developed a Support Vector Machine (SVM) based Active Learning model for cancer classification, but this approach suffers from the overhead of choosing an optimal SVM kernel as well as the tuning of the hyper-parameters.

Our approach aims to introduce an ensemble of classifiers as a novel approach in Active Learning and develop a weighted probabilistic model to enable garnering more importance towards classifiers which performs well on the training set. Conventional ensemble techniques assign equal importance to every classifier used in an ensemble technique, thereby classifiers which under-fit during classification are also carried forward with equal impetus. To overcome this issue, we assign weighted parameters as discussed later to classifiers that perform well to direct the classification towards with certainty. As a result, the quality of training is enhanced and the ensemble becomes well-furnished towards classification. Our model has been evaluated on microarray data.

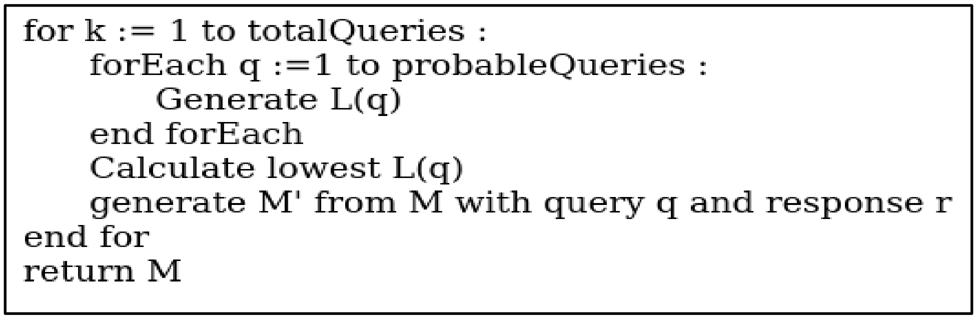
2 Preliminaries

As mentioned previously, Active Learning differs from traditional machine learning on the ground that the former asks queries and receives responses. We define a model M and its corresponding quality by its model loss with $L(M)$**.** With the following parameters, we aim to minimize the model loss with every query at each iteration. With a new query q**,** we evaluate the model M', which is the model M with query q and response r**.** Not knowing the true response to a query q, an aggregated distribution of responses can be computed, which is then utilized to generate the expected model loss, corresponding to a query. The expectation is taken over the viable responses to the query generating the equation which focuses on choosing queries which minimizes the expected model loss given by equation 1.

$$L(q) = E_r L(M') \qquad (1)$$

Where $L(q)$ is the generated loss against a query q, E_r is the expectation taken over the responses and $L(M')$ is the model quality of M'.

In a nutshell, given a model and its quality, we aim to select queries which give the lowest feasible model loss. Figure 1 describes a general Active Learning algorithm.



```
for k := 1 to totalQueries :
    forEach q :=1 to probableQueries :
        Generate L(q)
    end forEach
    Calculate lowest L(q)
    generate M' from M with query q and response r
end for
return M
```

Fig. 1 A general Active Learning algorithm

In a pool-based Active Learning setup, the process is initialized with a small training dataset D and a large unlabeled pool U of data. On every iteration, the learner selects samples from the unlabeled pool of data for labeling. One or more data are then selected for labeling by an annotator. The unlabeled samples, chosen here to add to the model, are quantified based on some information which is an important criterion of any Active Learning model. This information is assessed by some utility measures such as the margin sampling and entropy sampling. The now-labeled samples are added to the training set, after which the process is iteratively continued. After every iteration, the model is re-trained aiming to upgrade the model quality. The final performance evaluation is done on a separate test set which is independent of the trained set.

2.1 Selection Procedures

Traditionally, various methods are deployed in Active Learning to select the most confusing samples to be labeled by experts. In this section, we describe some such measures. This paper uses the method of marginal selection described in Section 2.1.3

2.1.1 Random Selection

This is the simplest and the most computationally inexpensive method where a random number of samples are queried for labeling. As is apparent, this method does not involve computing any objective function for a given sample, and hence can never be assured of providing the classifier with a better update.

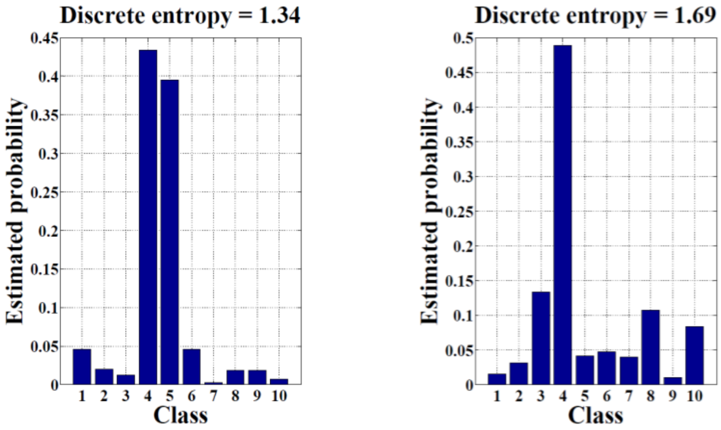
2.1.2 Entropy Measure

The measure of uncertainty of a random variable is defined as entropy. If a sample has high entropy, it goes on to show that the classifier is confused about its class membership. Samples with the highest value of entropy are queried at every iteration, labeled manually, incorporated into the training set and the model is retrained with the updated training set. Equation 2 gives a mathematical description of entropy H for a range of class labels y_i :

$$H = -\sum_i P(\, y_i) \log P\,(y_i) \qquad (2)$$

2.1.3 Marginal Selection

A major drawback of entropy based Active Learning is that it is heavily affected by the probability values of classes having a lower probability. Figure 2 illustrates this for two instances of a 10-class problem. The instance on the left has a smaller entropy than the one on right. However, the classifier is more confused about the former due to the close probability values for the two classes 4 and 5. For the instance in Fig.2(b), the small probability values of classes having a lower probability result in a higher entropy although the classifier is relatively more confident that the sample belongs to class 4. This problem becomes even more acute as the number of classes increases.



(a) (b)

Fig. 2 An illustration of why entropy can be a poor estimate of classification uncertainty. The plots show estimated probability distributions for two unlabeled instances of a 10-class pattern classification problem. [19]

To overcome this problem, in this paper, we employ a greedy approach. We define the measure of uncertainty to be the difference between the probability values of the two classes having the highest estimated probability value. Since it is a comparison between the best guess and the second-best guess, it is also known as the Best-versus-Second-Best (BvSB) approach. This is a more direct measure of estimating confusion about class membership from a classification perspective. If the difference between the highest and second highest probability is small, then the classifier becomes confused about which of these two classes to output as the final label. This implies that classifier becomes indecisive to make a correct guess while a significant difference infers that the classifier is confident and most likely correct about the final class label. Using this approach, the example of **Fig. 2(a)** is selected as the more confusing one.

2.2 Classifier Ensemble

Traditionally, in order to solve a particular pattern recognition problem, different classifiers are tested and the one which yields the best performance is chosen as the final solution to the problem. However, it has been observed that the sets of patterns misclassified by the different classifiers do not necessarily overlap. This has been suggested that different classifiers potentially offered complementary information [18] about the pattern which could be utilized to give a better performance. These observations have triggered the interest of the researchers to combine different information obtained from the classifiers instead of relying on a single one. The individual opinions are then used to arrive at a consensus decision. Over the years, various classifier combination schemes have been devised by the researchers and it has been experimentally demonstrated that some of them consistently outperform a single best classifier [18].

The combination of classifiers is particularly useful if they are different [20]. This combination can be done by using different feature sets [21], [22], by using different training sets, randomly selected, [23] or based on a cluster analysis [24]. Alternatively, we could use completely different classifiers and integrate their outputs. An interesting topic for research concerning classifier ensembles is the way the classifiers are combined. If only labels are available (abstract-level output), a method called majority voting [25], [26] is used. Sometimes, we can use a label ranking [27], [22]. Continuous outputs such as posteriori probabilities, on the other hand, let us use some linear combinations like average, product [28],[21], [29], [30]. Some of these techniques are briefly described below.

Consider that we have T number of classifiers and C is the number of classes representing the dataset under consideration. The decision of the t^{th} classifier for the j^{th} class is defined as $d_{t,j}$ where $t = 1, \cdots, T \ and \ j = 1, \cdots, C$. For abstract-level output, $d_{t,j} \in \{0,1\}$. If t^{th} classifier chooses class j, then $d_{t,j} = 1, else\ 0$. For continuous outputs, $d_{t,j} \in [0,1]$. Such outputs are usually normalized so that they add up to 1. This may be interpreted as the normalized support given to class j by the classifier t, or as an estimate of the posterior probability.

The final ensemble decision is the class j that receives the largest support $\mu_j(x)$ after the algebraic expression is applied to individual supports obtained by each class. Specifically,

$h_{final}(x) = argmax(\mu_j(x))$, where the final class supports are computed as follows:

Average rule: $\boldsymbol{\mu_j(x) = 1/T \sum_{t=1}^{T} d_{t,j}(x)}$

Sum rule: $\boldsymbol{\mu_j(x) = \sum_{t=1}^{T} d_{t,j}(x)}$ (provides identical final decision as the mean rule)

Product rule: $\boldsymbol{\mu_j(x) = \prod_{t=1}^{T} d_{t,j}(x)}$

Max rule: $\boldsymbol{\mu_j(x) = max\{d_{t,j}(x)\}; t = 1, 2, \dots, T}$

Majority voting: $\boldsymbol{\sum_{t=1}^{T} d_{t,J}(x) = max\{\sum_{t=1}^{T} d_{t,j}(x)\}; j = 1, 2, \dots, C}$
The class J that gets the highest total vote is chosen.

There are several more sophisticated rules such as Borda Count, which considers rankings of the class supports, as well as Dempster-Shafer theory based technique which calculates the plausibility based belief measures for each class [26], [31], [32].

In Active Learning, we need the final probability measure or membership degree for each class in order to select the most confusing samples to be labeled for the next iteration. Hence, we use an ensemble of different classifiers that output continuous confidence scores for each class, which are then normalized to give a probability estimate, as discussed earlier.

2.3 Classifiers Used

As mentioned before, the ensemble of classifiers is particularly effective if they provide complementary information about the pattern classes to be classified. Each classifier may perform well on different sets of instances for a pattern. In addition to that, different classifiers draw different conclusions about the features fed into the system based on the underlying principle of the classifier. In other words, one classifier may be able to capture certain patterns in the data which the other might not be able to. For this reason, we have chosen a diverse set of classifiers that work on completely different principles and their computation is entirely independent of one another. The final ensemble takes into account the decisions of each individual classifier and generally gives better performance. Here we have provided a brief description of the classifiers that have been used in this paper.

2.3.1 Logistic Regression

The logistic regression model takes real-valued inputs, applies the predefined hypothesis function and then applies a sigmoid function on the result to get the final output in the range of [0, 1]. This is interpreted as the probability of the input belonging to the predefined default class. The gradient descent optimization function is generally used for parameter estimation.

2.3.2 Support Vector Machine

SVM looks for the hyper-plane which best separates the data points belonging to different classes. Maximizing the distances between the hyper-plane and the nearest data points can help achieve the result. The closest points which identify the plane are referred to as support vectors and the region they define around the plane is called margin. The kernel trick is often used to project the data to a higher dimension where there is a clear dividing margin between the classes.

2.3.3 Random Forest

Random Forest creates a forest of decision trees and predicts the label by aggregating the predictions of all the trees. It randomly picks a subset of samples from the training set with replacement and use this to grow a particular tree. A subset of features is also chosen at random and the best split among them is used to split the node. Each tree is grown to the largest extent possible and there is no pruning.

3. Proposed Method

This paper looks at pool-based Active Learning, where a large pool of unlabeled samples (***Unlabeled Pool***) is available to the learner, and the queries may be done only from this pool. Initially, a small number of samples is selected randomly and labeled. This forms the ***Seed Set*** which is used for initial training of the classifiers. We assume that we have finite resources and can manually label at most ***max_queried*** number of samples. In each iteration of Active Learning, we select the ***k*** most confusing samples using the method of Marginal Selection, described in section 2.1.3. The value of ***k*** is predetermined. These are manually labeled by experts and added to the ***Training Set*** which consists of all samples that have been labeled so far.

This querying process is iterative and the updated Training Set is used to retrain the classifiers during the next iteration. Finally, performance evaluation is done on a separate ***Test Set*** different from the ***Seed Set*** and the ***Training Set***.

We have used an ensemble of three classifiers described earlier and computed a weighted average of the outputs of the three classifiers, taking into account the history of each classifier's performance until now. For ease of understanding, we have divided the proposed method into different disjoint phases, explained below and Fig. 2 displays a flowchart of the proposed method.

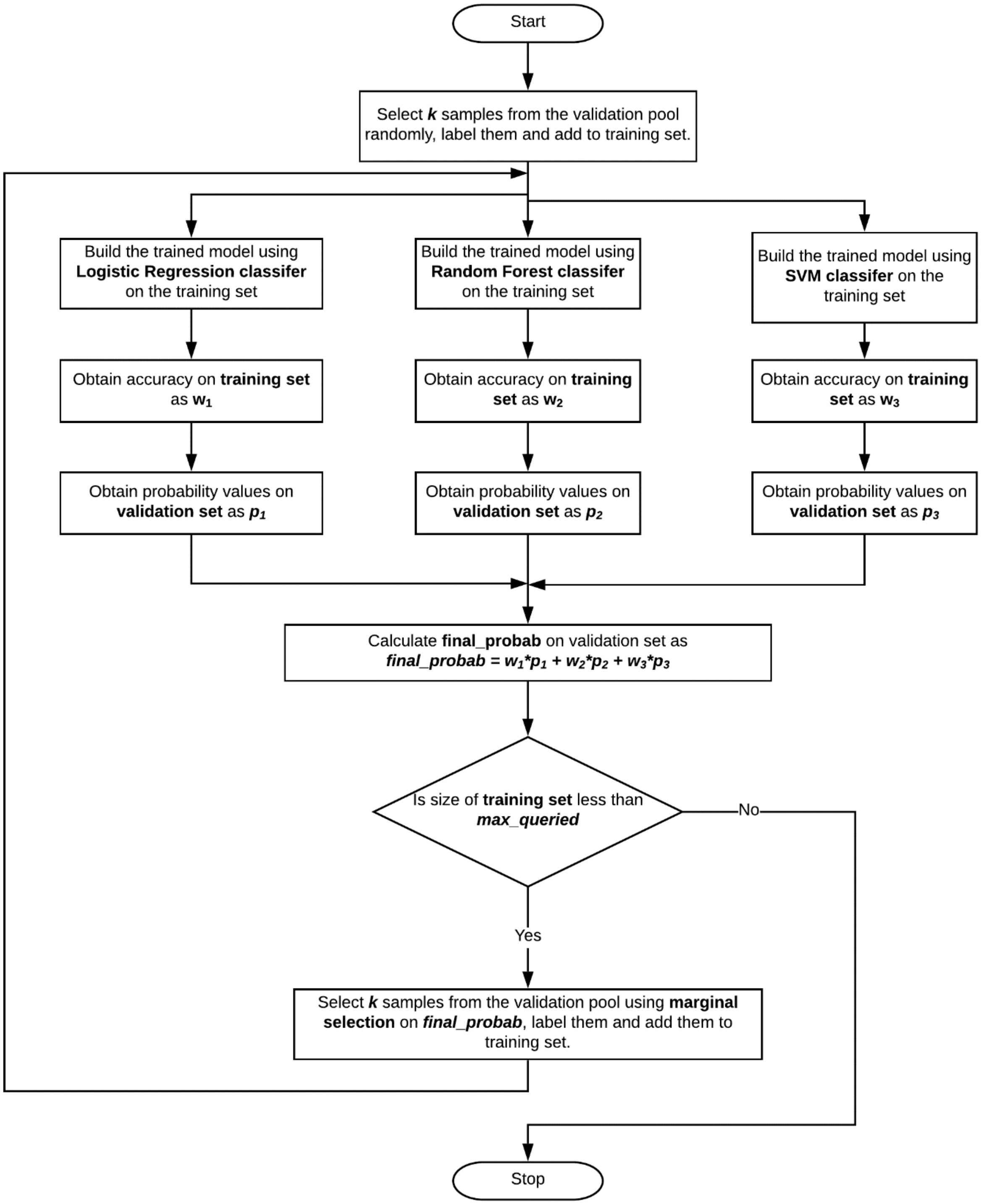

Start
Select k samples from the validation pool randomly, label them and add to training set.
Build the trained model using Logistic Regression classifer on the training set
Build the trained model using Random Forest classifer on the training set
Build the trained model using SVM classifer on the training set
Obtain accuracy on training set as w1
Obtain accuracy on training set as w2
Obtain accuracy on training set as w3
Obtain probability values on validation set as p1
Obtain probability values on validation set as p2
Obtain probability values on validation set as p3
Calculate final_probab on validation set as final_probab = w1*p1 + w2*p2 + w3*p3
Is size of training set less than max_queried
No
Yes
Select k samples from the validation pool using marginal selection on final_probab, label them and add them to training set.
Stop

Fig. 3 Flowchart of the proposed ensemble based Active Learning framework

3.1 Training Phase

As mentioned before, the combination of classifiers will be ideal if each individual classifier provides complementary information about the pattern classes under consideration. Thus, instead of using classifiers of the same nature such as a set of SVMs with different architectures or several K-NN classifiers with different values of $\boldsymbol{k}$, we use 3 classifiers which function on completely different principles. This gives a greater likelihood for the classifier of learning different characteristics from the data. The classifiers used, namely Logistic Regression, Random Forest, and SVM, have been elucidated in Section 2.3. Other classification techniques could also potentially be employed keeping the above discussion in mind.

In each iteration, the ***Training Set*** is used to train the three classifiers independently, and these trained classifiers are used in the next phases.

3.2 Validation Phase

Once the classifiers have been trained, it is necessary to evaluate the performance of the individual classifiers. Traditionally, $\boldsymbol{k}$-fold cross-validation is used to estimate the skill of a machine learning model. This requires an additional set of labeled samples (i.e. *Validation Set*), which cannot be used for training. However, scarcity of labeled samples in microarray gene expression data, limits the use of an extra validation set. Thus, we use the ***Training Set*** itself for validating the performance of the classifiers. Empirically, this does lead to significant degradation in the performance of the model when used for the microarray data. Compared to the cost of labeling the data samples and considering the fact that Active Learning tries to train the model using minimum possible labeled samples, this works as a good trade-off as we can utilize each labeled sample to train the classifiers.

The accuracies of the three classifiers are stored as weights w_1, w_2, w_3 to be used later for the ensemble.

3.3. Prediction Phase

The three classifiers are used to predict the class label for each sample in the ***Unlabeled Pool***. For each sample, the classifier i gives a vector $\mathbf{P_i} = [\,\mathrm{P_{i1}}, \mathrm{P_{i2}}, \dots, \mathrm{P_{in}}\,]$ (considering n-class classification problem). P_{ik} denotes the probabilty of the sample belonging to class $\boldsymbol{k}$, that is output by the classifier i.

Outputs of all the classifiers are combined to give the final ***Class Membership Degree(M)*** which gives the final probability of the sample to belong to a certain class. M_i denotes the probability of the sample belongs to class i. It is computed using the eqn. 3:

$$\boldsymbol{M_i} = \boldsymbol{w_1} * \boldsymbol{P_{1i}} + \boldsymbol{w_2} * \boldsymbol{P_{2i}} + \boldsymbol{w_3} * \boldsymbol{P_{3i}} \qquad \dots \mathbf{(3)}$$

Using vector representation, the eqn. can be written in more compact form:

$$\boldsymbol{M} = \boldsymbol{w_1} * \boldsymbol{P_1} + \boldsymbol{w_2} * \boldsymbol{P_2} + \boldsymbol{w_3} * \boldsymbol{P_3} \qquad \dots \mathbf{(4)}$$

where $\boldsymbol{M} = [\,M_1, M_2, M_3, \dots]$.

3.4 Selection Phase

Since in our work, we deal with a multi-class classification problem, the entropy measure may sometimes lead to unpredictable behavior. So, we use the method of ***Marginal Selection***, described in Section 2.1.3, to select the most confusing samples from the ***Unlabeled Pool***. These are sent for labeling by experts and simultaneously removed from the unlabeled set.

3.5 Testing Phase

For the testing phase, a separate test set which is independent of the training set has been used. Each of the test samples is then fed into the trained classifiers and the required ***Class Membership Degree*** is obtained for every class, which are then used in order to ensemble those by the proposed ***weighted average procedure*** and a final *Class Membership Degree* ***M*** is obtained. The final class is then found out as follows,

$$\boldsymbol{Class\ label} = \boldsymbol{argmax}(\boldsymbol{M}) \qquad ..(5)$$

These obtained class labels are compared against the actual class labels to compute the accuracy of the model.

Algorithm: Ensemble based Active Learning Model

Input: Preprocessed dataset

Output: Trained model

Initial State

Number of labeled samples (***Training Set***) = 0

Number of unlabeled samples (***Unlabeled Pool***) = N

1. Select K samples randomly from the ***Validation Pool*** and label them manually using the help of experts.

2. Add these labeled samples to the ***Training Set*** and remove them from the ***Validation Pool***.

3. ***Training Phase:***

3.1. Train a **Logistic Regression** classifier using the current ***Training Set***.

3.2. Train a **Random Forest** classifier using the current ***Training Set***.

3.3. Train an **SVM** classifier using the current ***Training Set***.

4. ***Validation Phase***:

4.1. Compute the accuracy of the trained models on the ***Training Set*** itself.

4.2. Store the obtained accuracies of the models of 3.1, 3.2 and 3.3 as $\boldsymbol{w_1}, \boldsymbol{w_2}, \boldsymbol{w_3}$ respectively.

5. ***Prediction Phase***:

5.1. Compute probability of each sample of the ***Unlabeled Pool*** through the models in 3.1, 3.2, 3.3 and obtain the probability vectors $\boldsymbol{p_1}, \boldsymbol{p_2}, \boldsymbol{p_3}$ respectively.

5.2. Compute the vector ***M*** for ***Class Membership Degree*** of each sample using Eq. 4.

6. ***Selection Phase***:

6.1. Using *Marginal Selection* on the set of vectors ***M***, select the ***k*** most confusing samples.

6.2. Label these samples by experts.

7. Repeat steps 2 to 6 as long as size of the ***Training Set*** is less than ***max_queried***.

4. Experimental Results and Evaluation

This section describes the datasets used and the results obtained on these datasets by applying the proposed method.

4.1 Datasets Used

The experiments have been performed on eight microarray datasets [33], [34] namely **AML, Bladder, DLBCL, Leukemia, MLL, Prostate**. The datasets consist of the gene expression values and the class labels associated with each data. It is to be noted that only a few samples have been used to train the classifiers. A brief discussion about the datasets is given below and **Table 1** gives a summary of the datasets used.

AML dataset consists of data from patients with acute myeloid leukemia (AML) according to their prognosis after treatment - remission or relapse with resistant disease. Out of 54 samples, 28 samples are from remission patients and 26 samples are from relapse patients.

Bladder consists of microarray gene expression data from 40 patients affected with bladder carcinoma. 10 samples are from patients in stage T2+, 19 in stage Ta, and 11 in stage T1.

DLBCL Follicular Lymphoma and diffuse large B-cell lymphomas are the two B-cell lineage malignancies. The subtypes are DLBCL (58 samples) and Follicular Lymphomas (FL) (19 samples). Among the 58 DLBCL patient samples, 32 are from cured patients, while 26 are from patients with fatal diseases. The dataset has 77 samples and 7070 genes.

Leukemia This dataset consists of gene expression samples from human AML and acute lymphoblastic leukemia (ALL) from 72 patients out of which 47 are ALL patients and 25 are from AML.

MLL This dataset consists of gene expression samples from Mixed-lineage Leukemia (MLL), which is a subset of ALL with chromosomal translocation involving the MLL gene, from 72 patients out of which 24 are cases of ALL, 20 cases of MLL and 28 cases of AML.

Prostate Prostate is an Affymetrix Human Genome 95Av2 (HG U95Av2) array set. The dataset contains 102 samples and 12533 genes, out of which 52 are prostate-tumor samples and 50 are non-tumor prostate samples.

Table 1. Summary of the six microarray datasets used for the experiments

Dataset	**Number of samples**	**Number of Genes**	**Number of classes**
AML	54	12616	2
Bladder	40	5724	3
DLBCL	77	7070	2
Leukemia	72	5147	2
MLL	72	12533	3
Prostate	102	12533	2

4.2 Results and Discussion

The experiments have been performed on the data (10 times), and the average accuracy and the standard deviation have been noted for each of the datasets. As the number of genes in a microarray dataset is very high compared to the number of samples, it becomes very difficult to train a classifier with sufficient accuracy. Hence, feature selection plays an important role before classifier is used for microarray data classification. Therefore, for each of the datasets, the proposed method has been applied by selecting 10, 50 and 100 features respectively. The feature selection technique used here is *Mutual Information* [35]. **Table 2 s**hows the accuracies obtained for the respective datasets using 10, 50 and 100 features, k=5 and labeling at most ***max_queried***=30 number of samples. In all of the experiments 40% of the dataset are used for testing. Though accuracy was not the key interest here, still the proposed method shows reasonably satisfactory results with a relatively smaller number of training samples. Also, the value of k has been chosen to be as minimum as possible so that initially lesser number of samples needs to be labeled at the same time making sure that all possible classes are selected by random selection.

Table 2. Results of the proposed weighted ensemble-based method on the microarray datasets

Dataset	**Number of Samples**	**Number of selected Features**	**Accuracy (mean) in %**	**Standard Deviation**
AML	54	10	70.74	3.68
		50	72.59	4.68
		100	83.33	3.14
Bladder	40	10	87.5	7.16
		50	90	0
		100	95	2.35
DLBCL	77	10	96.40	1.32
		50	96.12	1.24
		100	95.89	4.40
Leukemia	72	10	99.72	0.87
		50	98.05	1.34
		100	96.94	2.05
MLL	72	10	95.55	1.43
		50	100	0
		100	97.22	0
Prostate	102	10	95.67	3.78

		50	92.34	1.44
		100	93.91	2.34

In order to compare the proposed method with other traditional classifier ensemble methods, the experiments are also performed by changing the ensemble techniques to *sum rule*, *product rule* and *max rule*. **Table 3** shows a comparison among the aforementioned ensemble methods when applied to Active Learning with the same parameters as used in the previous experiments. Note that the rank of the method in terms of its accuracy has been provided within parenthesis.

Table 3. Comparison of the proposed method with other ensemble techniques

Dataset	**Number of Samples**	**Number of Selected Features**	**Accuracy with Sum Rule (in %)**	**Accuracy with Product Rule (in %)**	**Accuracy with Max Rule (in %)**	**Accuracy with Proposed Method (in %)**
AML	54	10	**75.04** (1)	59.25 (4)	62.21 (3)	70.74 (2)
		50	69.08 (3)	72.22 (2)	67.03 (4)	**72.59** (1)
		100	81.36 (3)	**88.88** (1)	74.44 (4)	83.33 (2)
Bladder	40	10	85.62 (2)	83.50 (3)	83.50 (3)	**87.50** (1)
		50	**94.37** (1)	88.50 (3)	80.00 (4)	90.00 (2)
		100	**100.00** (1)	92.00 (3)	91.00 (4)	95.00 (2)
DLBCL	77	10	92.89 (3)	95.89 (2)	89.98 (4)	**96.40** (1)
		50	94.86 (1)	94.86 (2)	88.45 (4)	**96.12** (1)
		100	92.25 (4)	**96.91** (1)	96.41 (2)	95.89 (3)
Leukemia	72	10	99.31 (3)	**100.00** (1)	89.16 (4)	99.72 (2)
		50	94.48 (3)	96.08 (2)	93.88 (4)	**98.05** (1)
		100	**98.27** (1)	96.36 (4)	97.22 (2)	96.94 (3)
MLL	72	10	89.30 (4)	91.30 (3)	94.40 (2)	**95.55** (1)
		50	100.00 (1)	100.00 (1)	100.00 (1)	**100.00** (1)
		100	93.10 (3)	97.20 (2)	97.20 (2)	**97.22** (1)
prostate	102	10	92.43 (3)	87.98 (4)	**96.46** (1)	95.67 (2)
		50	**99.02** (1)	91.10 (3)	89.36 (4)	92.34 (2)
		100	**97.80** (1)	91.69 (4)	97.63 (2)	93.91 (3)

Table 3 shows that the proposed method performed better than the others 8 times out of a total of 18 times, whereas the other methods like the sum has given best results in 6 cases, product rule in 3 cases and max rule in only 1 case. Thus, the proposed method is more robust compared to the other ensemble methods although it does not give the best accuracy in all cases. **Fig. 4** provides a bar graph summarizing the data in **Table 3**.

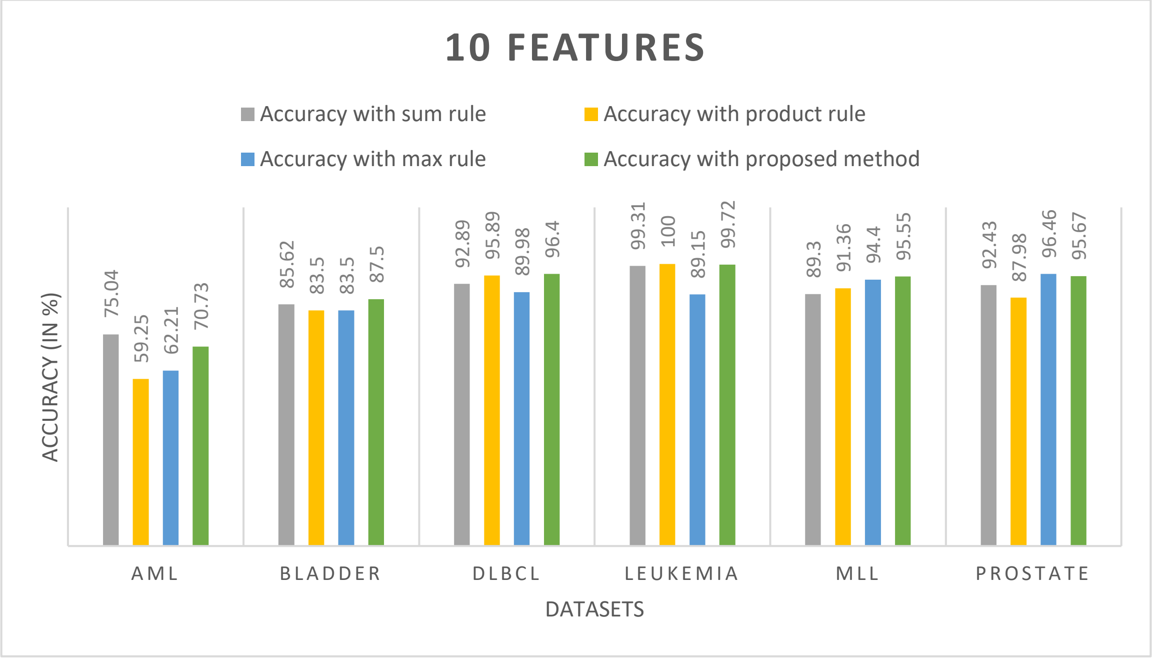


Fig 4. Graph showing accuracy for each dataset for 10 selected features with different ensemble procedures.

Table 4 shows a comparison among the performance of the various selection methods used in Active Learning on the said datasets. From **Table 4** it can be seen clearly that the proposed method using Marginal Selection outperforms the other Active Learning selection methods in most of the cases.

Table 4: Performance comparison of the various selection methods used in Active Learning on the six microarray datasets used here.

Dataset	**Number of Samples**	**Number of Selected Features**	**Accuracy with Random Selection (in %)**	**Accuracy with Entropy Selection (in %)**	**Accuracy with Proposed Method (in %)**
AML	54	10	**82.72**	78.17	70.74
		50	80.90	**81.81**	72.59
		100	73.63	74.54	**83.33**
Bladder	40	10	77.50	77.50	**87.50**
		50	83.75	85.00	**90.00**
		100	83.75	**95.00**	**95.00**
DLBCL	77	10	90.96	96.12	**96.40**
		50	88.38	**98.70**	96.12
		100	92.25	89.02	**95.89**

Leukemia	72	10	96.55	93.79	**99.72**
		50	93.79	97.24	**98.05**
		100	99.31	96.55	**96.94**
MLL	72	10	94.48	94.48	**95.55**
		50	96.55	96.68	**100.00**
		100	95.86	95.17	**97.22**
prostate	102	10	93.65	91.70	**95.67**
		50	**95.12**	90.24	92.34
		100	**94.13**	92.68	93.91

Not only does the proposed ensemble outperforms the other ensembles in most of the cases, but also it shows a considerable amount of consistency in performance in different scenarios. For the cases where it does not come up with the best performance, it is mostly ranked second. Thus, its performance never dips significantly, thereby showing certain reliability in its predictions.r Classifiers which fit poorly on the data could mislead the final classification decision and thus degrade the performance. The weighted ensemble, on the other hand, computes the weighted average of the decisions of the classifiers, thus giving less significance to the decision of the classifiers having a history of poor performance. For example, for the DLBCL dataset, SVM performs well and Random Forest gives a poor performance while the opposite is observed for the MLL dataset. The weighted ensemble is able to capture these variations in performance while evaluating the final result.

As mentioned previously, a separate validation set could be used to calculate the performance of the classifiers thereby computing their respective weights but this would require an additional set of labeled samples which would weaken the purpose of Active Learning to minimize the number of labeled data instances. Instead, the accuracy has been calculated on the training set itself after each iteration, the weights being proportional to the final accuracy obtained. However, from the experiments, it is observed that this is rarely the case. This is probably because Active Learning ensures that the number of training samples is minimum and eliminates any repeating or redundant samples. Moreover, new and confusing samples are added after each iteration further reducing the chances of overfitting of the classifier. Additionally, limitations of one classifier are compensated by the other classifiers of the ensemble which are often able to capture complementary information thus improving the generalization on the data. Thus, it can safely be said that the training set is a decent yardstick to measure the performance of the classifiers.

Fig. 5 shows the variation of accuracy on the two datasets Leukemia and MLL as the number of samples in the training pool increases.

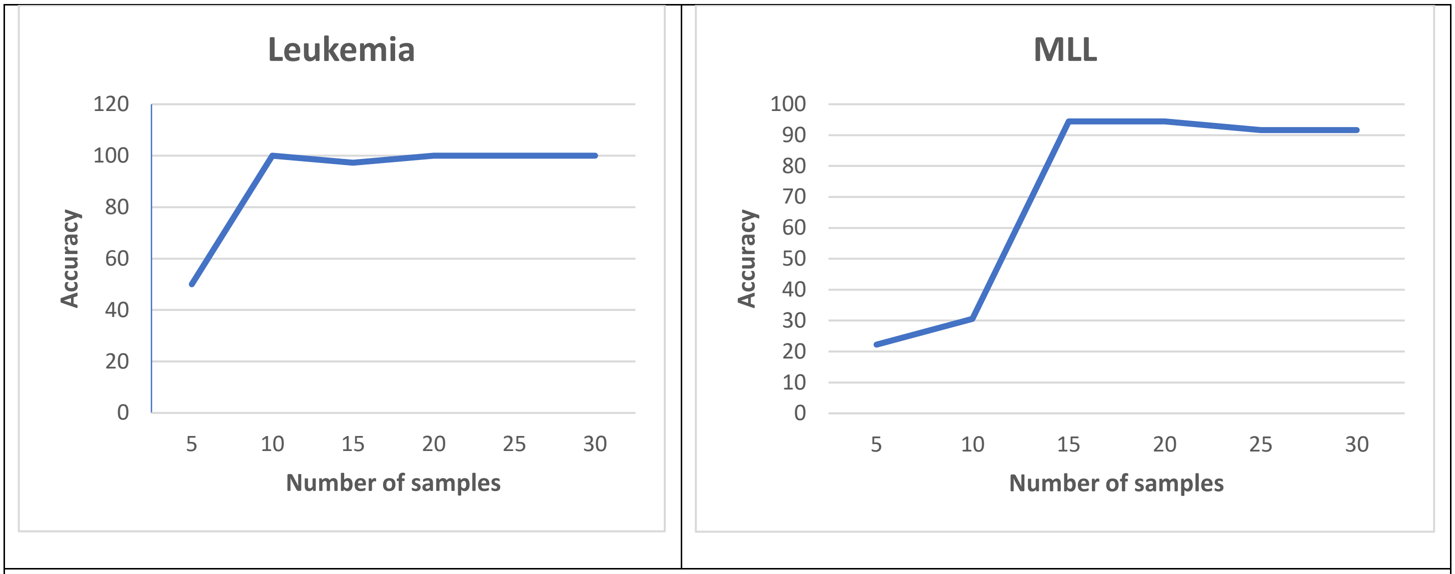


Fig 5. Variation of accuracy with increasing number of samples

It is to be noted that at the beginning, the classifiers perform poorly and its performance increases gradually as newer samples are added for training. Finally there comes a point after which there is no significant change in performance with increasing number of samples. Initially as new samples are added for training, the classifier is able to capture newer patterns of a dataset under consideration for the classification. As more patterns are recognized, the classification performance improves. Finally an optimum number is reached after which the newer samples carry no further information which could help in classification. Therefore, even if there may be numerous samples in the dataset, only a few of them are sufficient to encompass all information needed for classification and it is enough to label only these samples. Active Learning helps to select these samples for labeling while leaving out the rest, thereby greatly reducing the number of labeled samples used for training a classification model.

5. Conclusion

Due to the scarcity of labeled gene expression data for cancer classification traditional supervised machine learning algorithms may face difficulties during its training phase. As a solution to this problem, we propose an ensemble based Active Learning model in order to obtain better results with lesser amount of labeled data by making use of very few labeled samples. Using an ensemble of classifiers when the number of samples is less becomes important it diminishes the probability of choosing less informative data as it's constituents are trained in a non-identical way. The aim of the method is to maintain the percentage accuracy, with the sparse amount of data that is available. We vary the number of features selected, *k*, to test on the range of optimal features present within the given sample. In this paper, we apply pool-based Active Learning, where a large pool of unlabeled data samples is available to the classifier, and the queries can be done only from this pool. We use an ensemble of three classifiers (Random Forest, SVM and Logistic Regression) and estimated a weighted average of the outcomes of the three classifiers. Our method has been experimented on 6 microarray datasets namely AML, MLL, Bladder, Prostate, DLBCL, and Leukemia, and its recognition accuracy is comparable to state-of-the-art methods. The use of Active Learning in microarray datasets guarantees that each iteration of training, the most efficacious data sample is picked, whereas the ensemble minimizes the variance present in the samples that is inherent in data of this nature. In future, the method may be further improved to give better results by improvising on the ensemble procedure. One trivial extension would be to make use of other classifiers which differ in the classification principle. Furthermore, techniques such as Dempster-Shafer theory could be explored to better understand the degree of cohesion between the classifiers to be used in making the consensus. More such works could be explored in the near future to bridge the gap between domain discrepancies and help the community at large.

References

[1] S. Dasgupta, D. J. Hsu, and C. Monteleoni, “A general agnostic active learning algorithm,” in *Advances in Neural Information Processing Systems 20*, J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, Eds. Curran Associates, Inc., 2008, pp. 353–360.

[2] V. Krishnamurthy, “Algorithms for optimal scheduling and management of hidden Markov model sensors,” *IEEE Trans. Signal Process.*, vol. 50, no. 6, pp. 1382–1397, Jun. 2002.

[3] A. McCallum and K. Nigam, “Employing EM and Pool-Based Active Learning for Text Classification,” in *Proceedings of the Fifteenth International Conference on Machine Learning*, 1998, pp. 350–358.

[4] B. Settles and M. Craven, “An Analysis of Active Learning Strategies for Sequence Labeling Tasks,” in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2008, pp. 1070–1079.

[5] A. Holub, P. Perona, and M. C. Burl, “Entropy-based active learning for object recognition,” in *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2008, pp. 1–8.

[6] P. Mitra, C. A. Murthy, and S. K. Pal, “A Probabilistic Active Support Vector Learning Algorithm,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 3, pp. 413–418, Mar. 2004.

[7] Y. Freund, H. S. Seung, E. Shamir, and N. Tishby, “Selective Sampling Using the Query by Committee Algorithm,” *Mach. Learn.*, vol. 28, no. 2–3, pp. 133–168, Sep. 1997.

[8] C. Zhang and T. Chen, “An active learning framework for content-based information retrieval,” *IEEE Trans. Multimed.*, vol. 4, pp. 260–268, 2002.

[9] S. C. H. Hoi, R. Jin, and M. R. Lyu, “Large-scale Text Categorization by Batch Mode Active Learning,” in *Proceedings of the 15th International Conference on World Wide Web*, 2006, pp. 633–642.

[10] M. K. Warmuth, J. Liao, G. Rätsch, M. Mathieson, S. Putta, and C. Lemmen, “Active Learning with Support Vector Machines in the Drug Discovery Process,” *J. Chem. Inf. Comput. Sci.*, vol. 43, no. 2, pp. 667–673, 2003.

[11] Y. Liu, “Active Learning with Support Vector Machine Applied to Gene Expression Data

for Cancer Classification," *J. Chem. Inf. Comput. Sci.*, vol. 44, no. 6, pp. 1936–1941, Nov. 2004.

[12] S. C. H. Hoi, R. Jin, J. Zhu, and M. R. Lyu, "Batch Mode Active Learning and Its Application to Medical Image Classification," in *Proceedings of the 23rd International Conference on Machine Learning*, 2006, pp. 417–424.

[13] H. J. Ruskin, "Computational Modeling and Analysis of Microarray Data: New Horizons," *Microarrays (Basel, Switzerland)*, vol. 5, no. 4, p. 26, Oct. 2016.

[14] C. B. Epstein and R. A. Butow, "Microarray technology - enhanced versatility, persistent challenge," *Curr. Opin. Biotechnol.*, vol. 11, no. 1, p. 36—41, Feb. 2000.

[15] J. Fan and Y. Ren, "Statistical analysis of DNA microarray data in cancer research.," *Clin. Cancer Res.*, vol. 12, no. 15, pp. 4469–4473, Aug. 2006.

[16] K. A. Schalper *et al.*, "In situ tumor PD-L1 mRNA expression is associated with increased TILs and better outcome in breast carcinomas.," *Clin. Cancer Res.*, vol. 20, no. 10, pp. 2773–2782, May 2014.

[17] P. Xu, G. N. Brock, and R. S. Parrish, "Modified linear discriminant analysis approaches for classification of high-dimensional microarray data," *Comput. Stat. Data Anal.*, vol. 53, no. 5, pp. 1674–1687, 2009.

[18] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 3, pp. 226–239, Mar. 1998.

[19] A. J. Joshi, F. Porikli, and N. Papanikolopoulos, "Multi-class active learning for image classification," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 2372–2379.

[20] K. Ali, "On the Link between Error Correlation and Error Reduction in Decision Tree Ensembles," 1995.

[21] L. Xu, A. Krzyzak, and C. Y. Suen, "Methods of combining multiple classifiers and their applications to handwriting recognition," *IEEE Trans. Syst. Man. Cybern.*, vol. 22, no. 3, pp. 418–435, May 1992.

[22] T. K. Ho, J. J. Hull, and S. N. Srihari, "Decision combination in multiple classifier systems," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 1, pp. 66–75, Jan. 1994.

[23] D. H. Wolpert, "Stacked Generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–260, 2011.

[24] J. Cao, M. Ahmadi, and M. Shridhar, "Recognition of handwritten numerals with multiple feature and multistage classifier," *Pattern Recognit.*, vol. 28, no. 2, pp. 153–160, 1995.

[25] F. Kimura and M. Shridhar, "Handwritten numerical recognition based on multiple algorithms," *Pattern Recognit.*, vol. 24, no. 10, pp. 969–983, 1991.

[26] J. Franke and E. Mandler, "A comparison of two approaches for combining the votes of cooperating classifiers," in *Proceedings., 11th IAPR International Conference on Pattern Recognition. Vol.II. Conference B: Pattern Recognition Methodology and Systems*, 1992,

pp. 611–614.

[27] S. C. Bagui and N. R. Pal, “A multistage generalization of the rank nearest neighbor classification rule,” *Pattern Recognit. Lett.*, vol. 16, no. 6, pp. 601–614, 1995.

[28] S. Hashem and B. Schmeiser, “Improving model accuracy using optimal linear combinations of trained neural networks,” *IEEE Trans. Neural Networks*, vol. 6, no. 3, pp. 792–794, May 1995.

[29] J. Kittler, M. Hater, and R. P. W. Duin, “Combining classifiers,” in *Proceedings of 13th International Conference on Pattern Recognition*, 1996, vol. 2, pp. 897–901 vol.2.

[30] and T. W. J. Kittler, A. Hojjatoleslami, “Weighting Factors in Multiple Expert Fusion,” in *Proc. British Machine Vision Conf., Colchester, England*, 1997, pp. 41–50.

[31] G. Rogova, “Combining the results of several neural network classifiers,” *Neural Networks*, vol. 7, no. 5, pp. 777–781, 1994.

[32] V. Tresp and M. Taniguchi, “Combining Estimators Using Non-Constant Weighting Functions,” in *Advances in Neural Information Processing Systems 7*, G. Tesauro, D. S. Touretzky, and T. K. Leen, Eds. MIT Press, 1995, pp. 419–426.

[33] M. Ghosh, S. Begum, R. Sarkar, D. Chakraborty, and U. Maulik, “Recursive Memetic Algorithm for gene selection in microarray data,” *Expert Syst. Appl.*, vol. 116, pp. 172–185, 2019.

[34] M. Ghosh, S. Adhikary, K. K. Ghosh, A. Sardar, S. Begum, and R. Sarkar, “Genetic algorithm based cancerous gene identification from microarray data using ensemble of filter methods.,” *Med. Biol. Eng. Comput.*, vol. 57, no. 1, pp. 159–176, Jan. 2019.

[35] H. Peng, F. Long, and C. Ding, “Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.