IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 14 April 2023, accepted 8 May 2023, date of publication 31 May 2023, date of current version 20 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3281685

RESEARCH ARTICLE

Boosting With Prior for Accurate Classification

MUBASHER BAIG[1], TAHIR EJAZ[1], KHAWAJA M. FAHAD[1], SYED ASIF MEHMOOD GILANI[1], MIAN M. AWAIS[2], (Senior Member, IEEE), AND SANA SAEED[1]

[1]Department of Computer Science, National University of Computer and Emerging Sciences (NUCES), Lahore 54770, Pakistan
[2]Department of Computer Science, Lahore University of Management Sciences (LUMS), Lahore 54792, Pakistan

Corresponding author: Mubasher Baig (mubasher.baig@nu.edu.pk)

This work was supported by the Lahore University of Management Sciences.

ABSTRACT Adaptive Boosting (AdaBoost) based meta learning algorithms generate an accurate classifier ensemble using a learning algorithm with only moderate accuracy guarantees. These algorithms have been designed to work in typical supervised learning settings and hence use only labeled training data along with a base learning algorithm to form an ensemble. However, significant knowledge about the solution space might be available along with training data. The accuracy and convergence rate of AdaBoost might be improved using such knowledge. An effective way to incorporate such knowledge into boosting based ensemble learning algorithms is presented in this paper. Using several synthetic and real datasets, empirical evidence is reported to show the effectiveness of proposed method.Significant improvements have been obtained by applying the proposed method for detecting roads in aerial images.

INDEX TERMS AdaBoost, ensemble learning, prior/domain knowledge.

I. INTRODUCTION

Boosting based meta learning algorithms form an accurate classifier ensemble using a weighted combination of Freund and Schapireseveral simple classifiers selected using a base learning algorithm. The AdaBoost algorithm by [1], is one of the most well studied boosting algorithm. It maintains a distribution, D, over training examples that is used to select a classifier during successive iterations by minimizing the training error w.r.t D. The distribution is modified during the iterations to make the errors of individual classifiers independent. This idea of accuracy boosting was developed by Schapire [2] in a theoretical setting assuming the PAC learning model [3]. The idea culminated into the first practical accuracy boosting algorithm, AdaBoost by Freund and Schapire [1] that has been extended to use real valued classifiers as base learners and to use it for handling multi-category classification problems [4], [5], [6]. AdaBoost works in a typical supervised learning settings assuming the availability of labeled data $\{(\bar{x}_i, y_i)\ i = 1 \ldots N\}$ along with a learning algorithm to form an ensemble. Each point, $\{(\bar{x}_i, y_i)$, in the training data consists of a vector $\bar{x}_i$ of raw or high level measurements of the object of interest and the corresponding class label y_i. For example, in a voice activity detection (VAD) problem the feature vector might be either some high level measurements like signal energy, zero crossing rate etc or low level representations like amplitude samples. The features used to represent an object of interest are not arbitrary and the feature space representation reveals significant information about the solution. For example, it is highly unlikely that an audio frame with very low total energy contains any voice activity. Many learning tasks exhibit a similar structure and have apriori knowledge about the actual solution besides training data.

In general, feature space representations carry significant information about the actual solution and can be used to guide the learning algorithms while creating a solution. Either a human expert can provide such knowledge or an autonomous method might be created to extract such knowledge from the instances provided in the training data. As reported by Schapire et al. [7], the domain knowledge can be used to mitigate the necessity of large amounts of training data and can also be used to improve the overall classification accuracy and convergence properties of the learning algorithm. However, a generic solution for effectively incorporating the domain or prior knowledge into ensemble learning algorithms is not available. Only Schapire et al. [7] has presented a method for using prior along with AdaBoost to form the ensemble. Their method compensates for the shortage of training data and has been

The associate editor coordinating the review of this manuscript and approving it for publication was Nuno M. Garcia.

applied to various problems related to call classification [8] and classification of spoken dialogue [9]. The method outlined in [7] is primarily applicable to binary classification problems; though an extension to multi-class problems has also been presented. The methodology presented by Schapire et al. [7] uses the prior knowledge to generate a larger training set and uses AdaBoost-L Collins et al. [10] to build the final classifier ensemble. The prior was required to be in the form of a probability density $\pi_+(\bar{x})$ giving the conditional probability that a given instance $\bar{x}$ belongs to the positive class. Results presented in [7] suggest that the method does not present any significant advantage when sufficient training data is available. That means the domain knowledge is not contributing towards a faster convergence or better accuracy for learning problems as the number of instances in training data increases. Ideally one would expect a reduction in training time and enhanced accuracy if domain knowledge is used during the ensemble learning even if training data is available in abundance.

This paper presents and effective method of incorporating prior into AdaBoost based ensemble learning algorithms. The proposed method works for classifiers that output a conditional density over possible classes. Most of the learning algorithms including decision trees and classifier that output real valued outputs [4] can be modified readily to output the probability of an instance belonging to a certain class instead of a classification decision. The presented work suggests a criterion for selecting the base classifier so that the prior knowledge plays an effective role in the overall classification decision. Empirical evidence is provided to show the improved overall performance both in convergence and accuracy as obtained by using the proposed method for various datasets. It has been observed that the proposed method also compensates for the lack of training data.Therefore, it has obvious advantage over the method of incorporating prior into boosting presented by Schapire et al. [7]. The proposed method can be used to incorporate the prior in an entire class of boosting algorithms without significant modifications.It works extremely well for problems both when the prior is relatively precise and also when it is relatively vague. The method has been used to incorporate prior for several learning tasks of varying complexity using synthetic as well as real datasets from the UCI machine learning repository. In all cases the prior has been extracted from the training data and decision stumps have been used as base learners in the experiments. Significantly improved performance was observed in case of incorporating prior than the case when no prior was used during ensemble learning.

The proposed method has also been used on a challenging practical problem of automatic detection of road in aerial images proposed by Mnih [11]. The obtained results show a significantly improved performance for this real world problem.

The remaining paper is organized as follows: Section II provides detailed discussion of our method for incorporating prior knowledge into boosting. Section III presents the experimental settings and implementation details while the results on various datasets and our conclusion is presented in section IV.

II. COMBINING PRIOR AND BOOSTING

A detailed account of proposed method for prior incorporation is presented in this section after a brief review of AdaBoost.

Algorithm 1 presents pseudocode of the AdaBoost algorithm [1]. It takes labeled training data consisting of N examples $\{(\bar{x_i}, y_i)\ i = 1 \ldots N\}$ and uses it to form an ensemble consisting of a linear combination of T selected classifier instances. Each instance h_t of the classifier is selected using the weights D_t of training examples. The weight distribution is modified during every iteration so that the incorrectly classified training examples have larger weights in successive iterations. Finally, a linear combination of the selected classifier instances is formed to create the final ensemble $H(\bar{x}) = sign(\sum_{t=1}^{T} \alpha_t.h_t(\bar{x}))$.The weights α_t of each classifiers are computed using the error, ϵ_t, of h_t $w.r.t.$ the distribution D_t.

Algorithm 1 AdaBoost [1]

Require: Training Data $(\bar{x_1}, y_1) \ldots (\bar{x_n}, y_n)$
consisting of training instances $\bar{x_i}$ and corresponding labels $y_i \in \{-1, +1\}$ and
T = a parameter to specify total classifiers in the ensemble

1: Initialize the distribution $D_1(i) = \frac{1}{n} for i = 1 \ldots N$
2: **for** each t in range 1 to T **do**
3: Use weights D_t to select a classifier h_t
4: Compute the error $\epsilon_t = Pr[h_t(\bar{x_i}) \neq y_i]\ w.r.t\ D_t$
5: Compute $\alpha_t = \frac{1}{2} log(\frac{1-\epsilon_t}{\epsilon_t})$
6: Set $D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i.h_t(\bar{x_i}))}{Z_t}$
where Z_t is the normalization factor
7: **end for**
8: Final classifier $H(\bar{x}) = sign(\sum_{t=1}^{T} \alpha_t.h_t(\bar{x}))$

Most variants of AdaBoost [1], [4], [10], [12], [13] do not allow the direct use of prior knowledge during ensemble creation. Schapire et al. [7] proposed a way of prior incorporation into boosting that generates additional training examples using the prior. This creation of additional examples is shown to be equivalent to assigning additional weights to both the positive and negative versions of the example. The method was used to mitigate the necessity of large number of training examples and has been used to classify spoken dialogue, call classification, and for categorizing text [7], [8], [9].

A. INCORPORATING PRIOR INTO BOOSTING

To incorporate prior knowledge into any learning algorithm in general and AdaBoost in particular an approach similar

to that of Coryn A.L [14] is used. This approach is initially presented for binary classification problems where the label of a training example belong to either the $+ve$ or $-ve$ class i.e. each label $\in \{+1, -1\}$. Latter a straight forward extension is given to handle multiclass learning problems where label of each example comes from a larger set $\{y_1, y_2, \ldots y_k\}$ of possible labels. For the binary classification problems the prior $\pi_+(\bar{x})$ is assumed to be in the form of a conditional density such that $\pi_+(\bar{x}) = P(y_+|\bar{x})$ and hence denotes the probability of the actual class being $+ve$ for the given instance $\bar{x}$. It is also assumed that the ensemble $H(\bar{x})$ independently computes a conditional density $P(y_+|\bar{x})$. To combine the prior π_+ with the output of the learning algorithm, it is observed that the value of observation $\bar{x}$ must affect the classification decision via the probability estimates of the prior π_+ and that of the classifier H. As the classifier H and the prior π_+ are generated through independent processes therefore the conditional independence of H and π_+ given $\bar{x}$ can be assumed and hence the overall classification process that incorporates the prior into learning can be modeled using a simple belief network as shown in Figure 1.

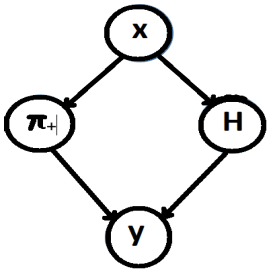


FIGURE 1. Belief network.

Therefore, the multiplicative approach of Coryn A.L [14] is used to combine these probabilities to form an estimate of the final probability using

$$P(y_+|\bar{x}) \propto \pi_+(\bar{x}).H(\bar{x}).P(y_+) \tag{1}$$

Assuming that $P(y_+)$ is constant and using a proportionality constant β the above equation can be written as

$$P(y_+|\bar{x}) = \beta\pi_+(\bar{x}).H(\bar{x}) \tag{2}$$

Equation 2 dictates a general multiplicative method of combining prior knowledge with the output of any learning algorithm that yield class conditional density estimates. M-Boost by Baig and Awais [12] is an example of such a boosted classifier. As many learning algorithms including Support Vector Machines, Decision Trees, Neural Networks etc, generate real valued outputs, therefore, the final output of such algorithms can be readily modified to output a conditional density estimates by using the sigmoid function. Hence the proposed prior incorporation method can be used with all such algorithms

Since AdaBoost forms the final ensemble as a linear or convex sum of base learner instances hence the final form of the ensemble is

$$H(\bar{x}) = P(y_+|\bar{x}) = \sum_{t=1}^{T} \alpha_t \dot{h}_t(\bar{x}) \tag{3}$$

Substituting 3 into 2

$$P(y_+|\bar{x}) = \delta.\pi_+(\bar{x}).\sum_{t=1}^{T} \alpha_t.h_t(\bar{x}) \tag{4}$$

with the normalization constant being δ that ensure that the output is a probability of the instance being in the positive class. Equation 4 summarizes our method of prior incorporation into AdaBoost based ensemble learning when the base classifier gives a probabilistic output. Hence the version of AdaBoost that uses prior during learning is similar to the original AdaBoost algorithm with the final classifier output computed using

$$H(\bar{x}) = sign\left(\pi_+(\bar{x}).\sum_{t=1}^{T} \alpha_t.h_t(\bar{x}) - \gamma\right) \tag{5}$$

with γ being the threshold used to assign the final classification label. This method can be used with all probabilistic base learners that generate an estimate of class density for the given input. For such classifiers, Equation 4 can be expressed as

$$P(y_+|\bar{x}) = \sum_{t=1}^{T} \alpha_t.\pi_+(\bar{x}).h_t(\bar{x}) \tag{6}$$

which defines the proposed method of prior incorporation into ensemble learning. By using equation 6 any boosting algorithm that uses probabilistic base learner can be readily modified to incorporate the prior $\pi_+(\bar{x})$ into boosting by considering prior combined base classifiers $p_t(\bar{x}) = \pi_+(\bar{x}).h_t(\bar{x})$.

A modification in the base learning algorithm would be required if such a prior incorporated classifier $p_t(\bar{x})$ is directly selected from first principle every time. This is due to the fact that the selection of base learning instance will not only be dependent on the weight distribution D_t maintained on training examples but will also depend on the prior as well. This will be only possible if we are selecting from a finite classifier search space. Only a simple learning algorithm like decision stumps can be readily modified to incorporate this change but algorithms like Decision Trees learning, SVM, Artificial Neural Networks do not have a finite search space and hence can not be used with this approach.

A different approach, therefore, is proposed for selecting a prior incorporated base classifier $p_t(\bar{x})$ during ensemble learning. The proposed method modifies the weight distribution W_t using the error of prior during each iteration and then uses this modified distribution to select the base learning instance h_t. The weights are modified using the error rate, ϵ_p, of the prior w.r.t the running distribution and hence the prior effects the selection of each base learner instance. The weights are modified using a multiplicative factor that is

computed just like the regular multiplicative factor used by AdaBoost. Since the prior is used to modify the distribution only therefore this method is applicable to all AdaBoost variants and can be used without modifying the base learner.

Algorithm 2 shows the resulting variant of AdaBoost called AdaBoost-P that uses prior in each iteration of AdaBoost. During each iteration, a distribution obtained by modification of original weight distribution is used for base instance selection hence incorporating prior into boosting. During each iteration the examples incorrectly classified by the prior get larger weights and hence the base learning algorithm must focus on such examples resulting in prior playing a role in the selection of h_t. Finally, the prior incorporated classifier instance $p_t(\bar{x})$ is used to modify the running distribution to obtain the new distribution D_{t+1} and hence the prior effects the distribution to be used in the next iteration as well.

Algorithm 2 AdaBoost-P

Require: Examples $(\bar{x_1}, y_1) \ldots (\bar{x_n}, y_n)$ where $\bar{x_i}$ is a training instance and $y_i \in \{-1, +1\}$ and parameter T = total base learners in the ensemble $\pi(y_+|x)$: Domain knowledge in the form of Prior

1: set $D_1(i) = \frac{1}{n}$ for $i = 1 \ldots n$
2: **for** t =1 to T **do**
3: set $\epsilon_p = Pr[\pi(x_i) \neq y_i]$ $w.r.t$ D_t
4: Set $D_{temp}(i) = \frac{D_t(i).\exp(-\alpha_p \Gamma[y_i = \pi(x_i)])}{Z_t}$
where $\Gamma[q]$ is 1 or -1 if q is true or false respectively and Z_t is the normalization factor
5: Select a weak classifier instance h_t which has small error $w.r.t$ D_{tmp}
6: set $\epsilon_t = Pr[\pi(x_i).h_t(x_i) \neq y_i]$ $w.r.t$ D_t
7: set $\alpha_t = \frac{1}{2} log(\frac{1-\epsilon_t}{\epsilon_t})$
8: Set $D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i.\pi(x_i).h_t(x_i)}{Z_t}$
where Z_t is the normalization factor
9: **end for**
10: output classifier
$H(x) = \frac{\sum_{t=1}^{T} \alpha_t.h_t(x).\pi(y/x)}{\sum_{t=1}^{T} \alpha_t}$
class with maximum posterior estimate is the predicted class.

B. MULTICLASS LEARNING

The proposed method of prior incorporation into boosting can be extended readily to handle multiple classes. For this case it is assumed that the prior π gives a probability density over the possible classes with $\pi_y(\bar{x})$ being the probability of predicted class being y for a given instance $\bar{x}$.

Just like the binary case it is assumed that the base learning instances h_t also generate a class density estimate for a give input instance $\bar{x}$. Assuming these changes the equation 4 becomes

$$P(y|\bar{x}) = \delta.\pi_y(\bar{x}).\sum_{t=1}^{T} \alpha_t.h_t(\bar{x}) \tag{7}$$

and hence the equation 6 after incorporating the above change can be written as

$$P(y|\bar{x}) = \sum_{t=1}^{T} \alpha_t.\pi_y(\bar{x}).h_t(\bar{x}) \tag{8}$$

From this equation it is evident that the method of prior incorporation remains the same for multi-class learning problems as well.

III. EXPERIMENTAL SETTINGS

This section presents various experimental setting used to verify the effects of using prior in boosting based learning algorithms. These settings include the description of boosting algorithms, the base learners, the datasets and our methods of generating prior from the training data.

A. BOOSTING ALGORITHMS

In this paper we present the results of using prior for two boosting algorithms, AdaBoost-M1 by Freund and Schapire [1] and Multiclass AdaBoost by Zhu et al. [15]. AdaBoost-M1 performs well when a strong base learner is used and it's performance degrades when the base classifier error goes above 50 percent [15]. Multiclass AdaBoost modifies the computation of the mixing parameter α_t in such a way that it is positive for any performance better than random guessing. It has better convergence properties than AdaBoost-M1 even for weak base classifiers [15].

B. BASE LEARNING ALGORITHM

The output of decision tree and other domain partitioning algorithms can be modified to give a generative output instead of a class label. For example, for each partition a count of instances for each class can be computed and probabilities can be assigned based on these counts instead of giving a classification decision based on majority count. Therefore decision trees have been used as base learns in all reported experimental results.

Results obtained using single node decision trees, also known as decision stumps, and with trees having $\lceil log(K) \rceil$ levels are included in this paper. Each decision stump partitions the space in two parts and hence is more suitable for binary classification tasks whereas a larger decision tree is a strong classifier that partitions the space in several parts with an independent class decision for each part.

C. DATASETS USED IN THE EXPERIMENTS

This paper presents experimental results on twelve multiclass datasets from the UCI machine learning repository by Frank and Asuncion [16]. A summary of these datasets is given in Table 1. Complete training set has been used to fit the model and test set used for evaluation when explicit train/test

partition is available for a dataset, 10-Fold cross validation has been in the absence of explicit train/test partition.

In Datasets of varying complexity including both synthetic and real datasets have been used in the experiments.

TABLE 1. Datasets used in our experiments.

Data set Name	Total Feature	Training Set	Test Set	Total Classes
Iris	4	150	C.V.	3
Pen Digit	16	7494	3498	10
Forest Fire	4	500	C.V.	4
Glass	10	214	C.V.	7
Vowel	10	528	462	11
Land State	36	4435	2000	8
Wine	13	214	C.V.	3
Waveform	21	300	4710	3
Yeast	8	980	504	10
Abalone	8	3133	1044	29
Letters	16	16000	4000	26
Segmentation	19	210	2100	8

D. GENERATING PRIOR

To study the effect of prior on ensemble learning, prior of varying accuracy have been used in the experiments. In all cases the prior has been extracted from the training data using the instance space structure. For the text categorization problem a method of generating prior was suggested by Schapire et al. [7] that uses expert assigned probabilities to form the prior assuming independence of occurrence of keywords in a dialogue. While this technique works well for the text categorization problems and in case of categorical features but can not be directly applied to real valued features. The requirement of human experts for computing prior is also a limitation of their proposed method.

Since the datasets from the UCI repository do not provide any prior available along with the data therefore we construct prior using the structure of instance space in all the experiments. A Gaussian is assumed to model each class with ML estimates of mean and standard deviation used to model each class. These are then used to assign class conditional probabilities for each instance. The information available in the structure of the instance space as captured by the training data is used to create a prior for each problem.

IV. RESULTS

Discussion of detailed results obtained for the experimental settings described in the previous section are presented here.

A. RESULTS

Our first set of results presents a comparison of Multiclass AdaBoost and AdaBoost-P for four datasets including the Pendigits, Handwritten Letter recognition, Forest Fire and the Waveform datasets. In these experiments a multi-split {Log(K) splits} decision tree learning algorithm has been used as the base classifier with 200 iterations of boosting to build the final ensemble. The comparison between test error rates of AdaBoost-P and Multiclass AdaBoost is given in figure 2.

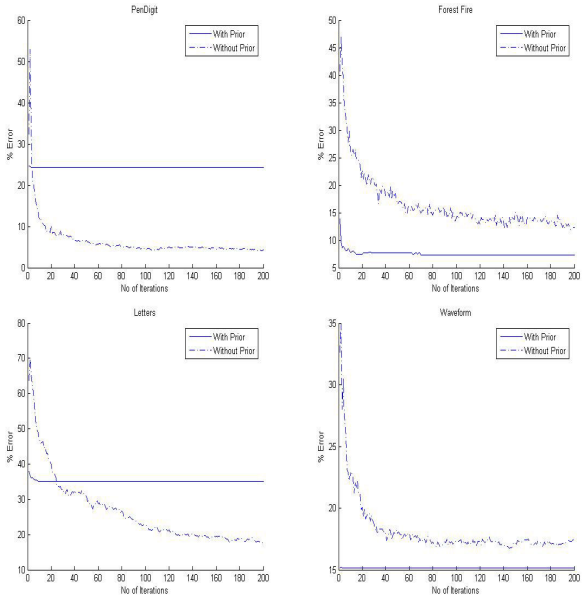


FIGURE 2. Test error: AdaBoost-P Verses Multiclass AdaBoost.

For the Forest Fire and Waveform datasets the prior significantly improve the accuracy of the boosted ensemble where as for the larger Pendigits and Letter recognition datasets the prior degrades the performance of Multiclass AdaBoost.

A similar comparison of AdaBoost-M1 and AdaBoost-P for the four datasets is given in the figures 3. From these experiments a similar conclusion can be drawn about the use of prior for creating an ensemble. From these experiments it is obvious that the prior can significantly improve the test error rate of the final ensemble even when AdaBoost-M1 diverges.

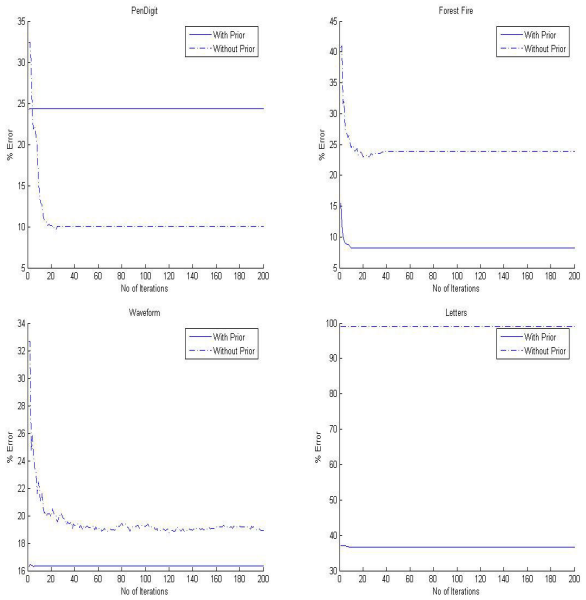


FIGURE 3. Test error: AdaBoost-P Verses AdaBoost-M1.

Table 2 gives a detailed comparison of test error rates of AdaBoost-M1 and AdaBoost-P for the 12 multiclass learning problems. In these experiments a decision stump has been used as the base learning algorithm and 200 iterations of boosting have been used to build the final ensemble.

It is clear from the results presented in Table 2 that for a simple learning algorithm the use of prior significantly improved the accuracy of final ensemble. This enhancement in the accuracy is due to the use of relatively accurate prior generated from the training data using a Gaussian model.

TABLE 2. Comparison for decision stump test error rates of AdaBoost-M1 vs AdaBoost-P.

Data set Name	AdaBoost M1	AdaBoost P
Iris	5.62	5.0
Pen Digit	99.17	26.98
Forest Fire	21.76	8.23
Glass	30.81	10.25
Vowel	95.46	65.01
Land State	99.95	23.53
Wine	5.26	2.14
Waveform	21.59	16.69
Yeast	60.99	52.47
Abalone	99.91	78.27
Letters	99.15	38.69
Segmentation	99.95	24.46

A similar comparison of Multiclass AdaBoost and AdaBoost-P for decision stumps with 200 iteration is given in Table 3. These results reveal that the use of prior is very effective

TABLE 3. Comparison for decision stump test error rates of multiclass AdaBoost AdaBoost-P.

Data set Name	Multiclass AdaBoost	AdaBoost P
Iris	6.25	5
Pen Digit	29.32	26.95
Forest Fire	15.09	8.82
Glass	20.49	8.43
Vowel	71.49	66.09
Land State	49.47	23.53
Wine	5.82	2.63
Waveform	17.10	15.29
Yeast	65.74	51.28
Abalone	82.29	78.66
Letters	61.03	38.69
Segmentation	11.23	23.75

The next set of results presents a comparison similar to the above for AdaBoost-M1 and for Multiclass AdaBoost with AdaBoost-P. In these experiments a multi-split {Log(K) splits} decision tree learning algorithm has been used as the base classifier with 200 iterations of boosting to build the final ensemble. The comparison of test error rates of AdaBoost-M1, AdaBoost-P is given in table 5 whereas the comparison of test error rates of Multiclass AdaBoost and AdaBoost-P is given in table 6. It is clear from the presented results that both AdaBoost-P can significantly improve the test error rate of the final ensemble.

TABLE 4. Comparison for multi-split decision tree test error rates of AdaBoost-M1 vs AdaBoost-P.

Data set Name	AdaBoost M1	AdaBoost P
Iris	5.00	4.37
Pen Digit	10.06	24.34
Forest Fire	23.92	8.23
Glass	2.23	6.71
Vowel	63.28	57.01
Land State	22.48	21.98
Wine	3.74	3.18
Waveform	18.93	16.33
Yeast	43.56	50.29
Abalone		77.60
Letters		36.66
Segmentation	10.04	22.17

TABLE 5. Comparison for multi-split decision tree test error rates of multiclass AdaBoost AdaBoost-P.

Data set Name	Multiclass AdaBoost	AdaBoost P
Iris	5.62	4.37
Pen Digit	4.22	24.26
Forest Fire	12.35	7.25
Glass	2.23	6.18
Vowel	49.89	62.63
Land State	38.08	21.58
Wine	3.18	3.18
Waveform	17.35	15.16
Yeast	54.05	48.11
Abalone	74.73	78.18
Letters	17.69	35.01
Segmentation	6.13	22.27

B. CASE STUDY

The results presented in the previous sections show promising contribution of the prior when combined with the base classifiers. This opens the door for further research to explore the role of prior, especially in situations where strong/good heuristics seem plausible. To this end, the current study has been extended to include a case. The primary focus of present study is to see the impact of incorporating prior into AdaBoost therefore we take the problem of detecting roads in aerial images [11] as of our primary case study. This dataset presents has been used extensively [17], [18] [19] in the past and presents a challenging problem of class imbalance and noise handling during classification. This case study presents several interesting challenges of class skew and noise in target labels. As we are primarily interested in evaluating the prior incorporation method into AdaBoost therefore this study focuses on the two variants of AdaBoost i.e. with and without prior. No attempt is made to compare the performance of our method with several other methods used to address this problem. To study the impact of prior onto learning an ensemble we will use prior knowledge of varying quality. Although the basic prior is computed using the rules of color falling in a certain range and also using the assumption that roads are long connected components and hence if a region has high probability of being a road segment

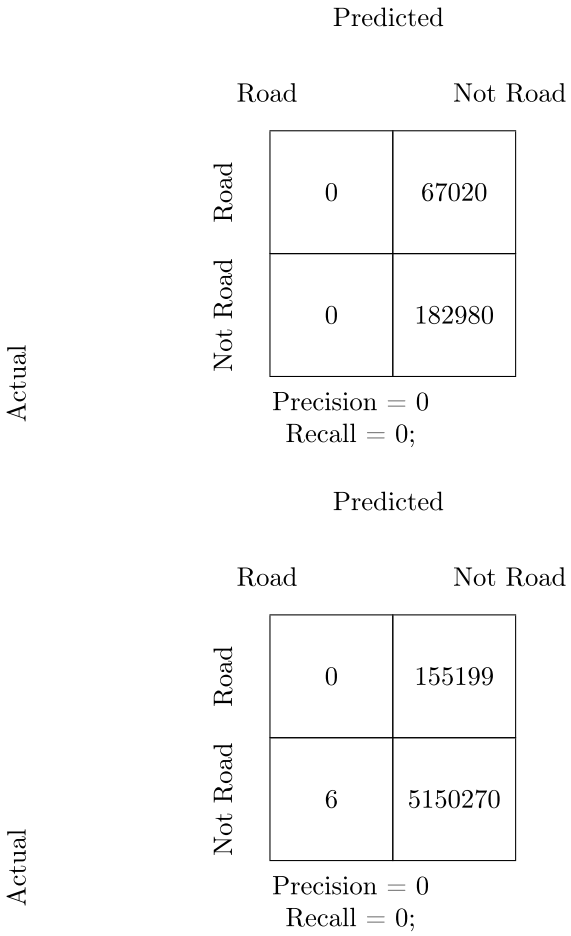


FIGURE 4. AdaBoost training and test error: Road detection without prior original split available in data(4% Roads).

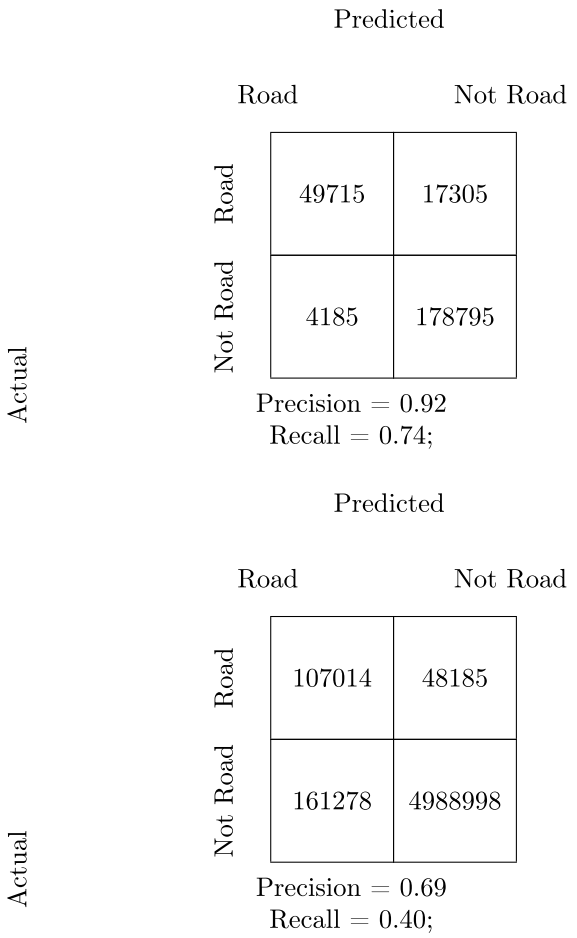


FIGURE 5. AdaBoost training and test error. Road detection with prior original split available in data.

then its neighboring segments also are highly likely to be part of the road segment. The details of generating prior are given in next section.

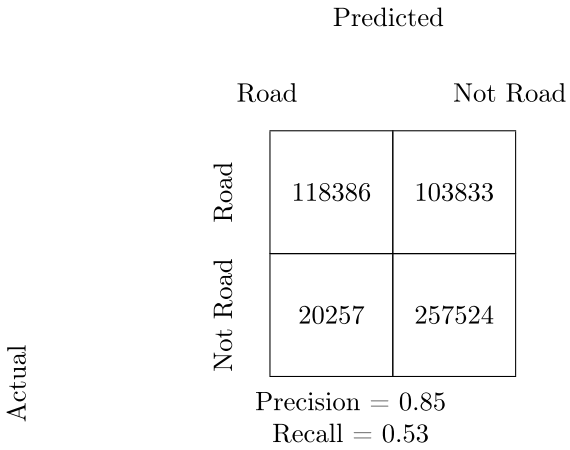
The Massachusetts Roads dataset [20] is the official state maintained transportation dataset and represents all public and some private roadways in Massachusetts. This dataset has an image resolution of 1 meter per pixel, and an image contains $1500 \times 1500 \times 3$ pixels. There are 1171 aerial images covering an area of more than 2600 square kilometers. The whole data set is randomly split so we have 1108 images as training data,14 images as validation data and 49 images as test set. The target maps were generated by rasterizing road center lines obtained from the Open Street Map project and hence are not totally accurate.

As we are going to label each segment/pixel as belonging to a road or otherwise therefore we have represented each segment as nine features with one of the feature value representing average intensity value of the segment under consideration and eight values each representing average intensity value of a neighboring segment. All reported experiments considered a segment of size 3×3 and hence each image is considered to be consisting of small 3×3 parts with each part either labeled as belonging to the road segment or otherwise. Next we describe a method of creating prior for each of the segment. In this case, a prior is a probability distribution and is provided as a single number specifying the probability that the associated frame belonging to a road segment i.e. the positive class.

1) PRIOR FOR ROAD SEGMENTATION

After visualizing the data in more depth priors are defined on base two characteristics of object i.e. road color and road structure.

- As far as road color prior is concerned we concluded that road object has certain RGB color values which distinguish the object from other objects. Usually in rural area roads are of brown color while in urban area grey color is prominent. So color can be one of primary distinguishing characteristic in roads. So we define threshold value based on RGB values to identify roads and used it as prior information.
- The second clue is the road structure itself. In any image we can see that road segment is connected means there is no patches in object. So this continuity leads to another observation that we can give more weight



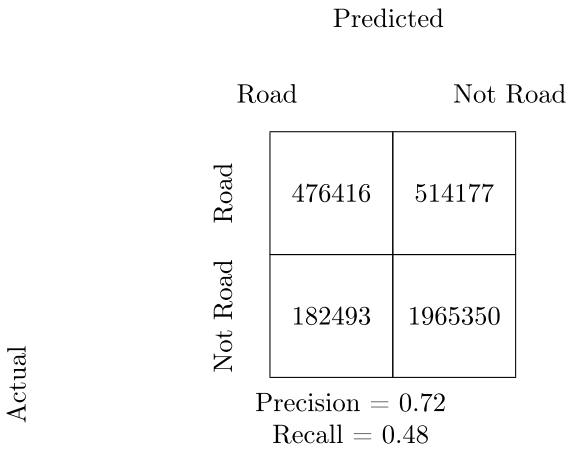
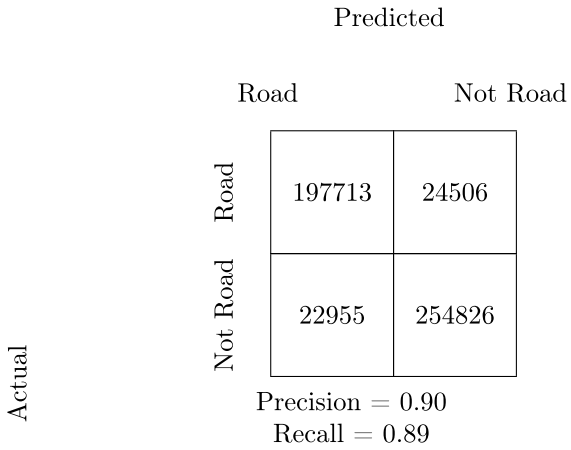


FIGURE 6. AdaBoost training and test error: Road detection without prior original split available in data(4% Roads).



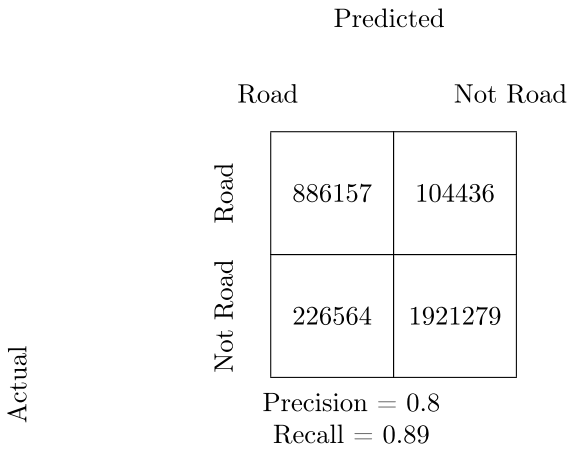


FIGURE 7. AdaBoost training and test error. Road detection with prior original split available in data.

to neighboring pixels in order to correctly classify the object. So road structure is used as second prior information

Specifically, we assumed that road color components varies about a mean value (110) color with a certain standard deviation (20) and hence a Gaussian function is used to assign confidence values to a segment based on its color values. Furthermore, the confidence score of a segment is used to assign a confidence score to all eight connected neighbors and for each segment the total confidence score is accumulated to assign an overall confidence score to a segment. The same mechanism is used to assign negative score to segments having color values clearly out of the color range of roads and the assumption of connectivity is also used for non-road components. Finally these scores are accumulated and are used along with the sigmoid function to assign probability of being part of a road segment. The prior obtained using this procedure can be used to assign labels to each segment of an image either by assigning the most probable label to the segment or by using a threshold such that all segments with a probability value greater than the certain threshold being member of positive class and member of negative class otherwise

2) CASE STUDY EXPERIMENTAL SETTINGS

A set of experiments have been conducted to verify the effects of prior on ensemble learning using AdaBoost algorithm. Single node decision trees have been used as base classifiers in each of these experiments and an ensemble of one hundred classifier has been created in each case. In all these experiments the algorithm converged to its best performance well before hundred components. In the first experiment we used prior generated by the procedure defined in previous section. In this set of experiments the ratio of positive and negative examples has been varied to see the effect of prior under various class imbalance conditions. The results are reported for three different class imbalance conditions. Furthermore, all reported results have been obtained using the examples created using first 100 images from the dataset with 10% of the data used for training purpose and remaining 90% data used as test samples.

3) CASE STUDY RESULTS

The first set of results, shown in figures 4 and 5, has been obtained with the an imbalanced training and test dataset.The original distribution is significantly more imbalanced and results in similar results. The second set of results, shown in

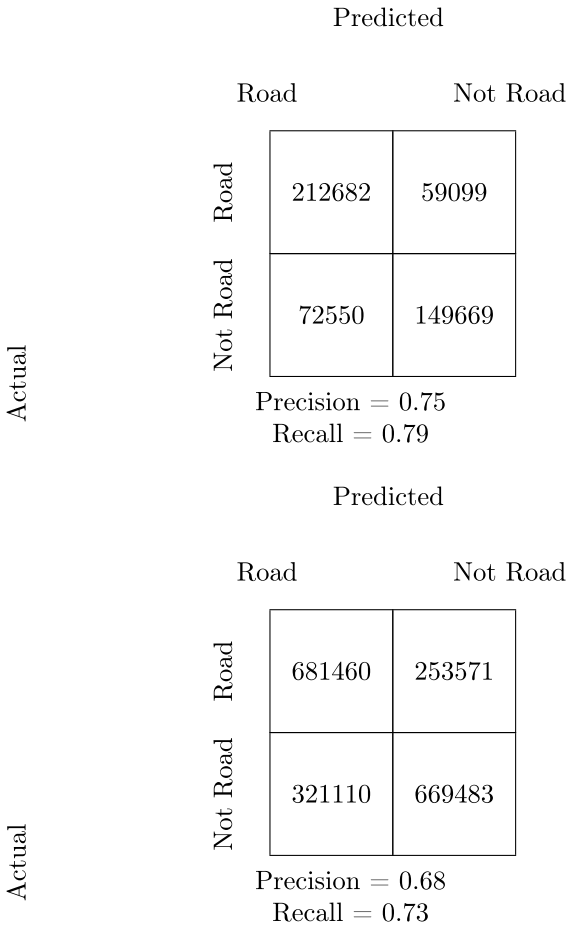


FIGURE 8. AdaBoost training and test error: Road detection without prior original split available in data(4% Roads).

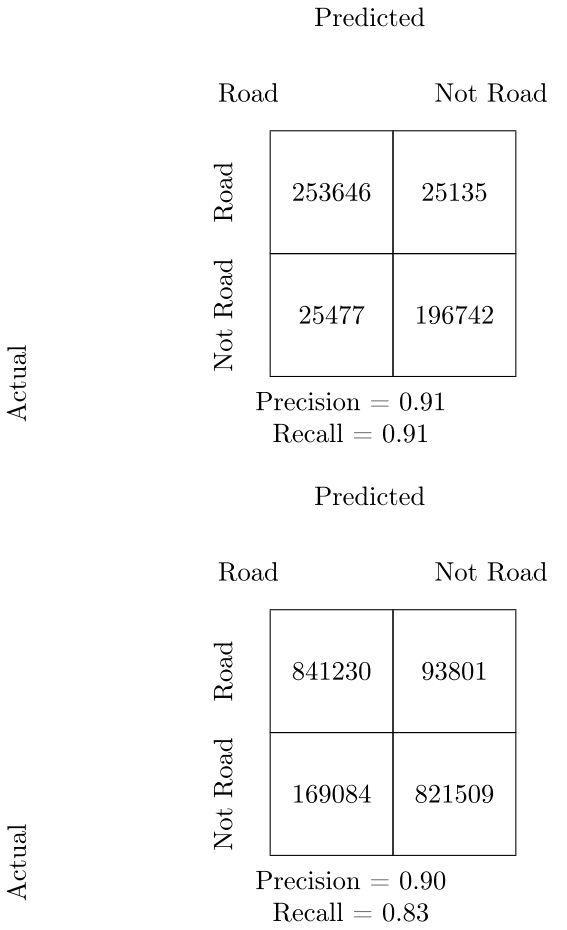


FIGURE 9. AdaBoost training and test error. Road detection with prior original split available in data.

figures 6 and 7, have been obtained with the positive class representing about 25 percent of total examples. The third reported set of results is for the case when the positive class has a slightly larger representation in the training and test data

The first set of experiments compares the standard AdaBoost and AdaBoost-P for the road detection in Aerial images while the distribution is skewed in favour of negative class (About 75% samples belong to negative class). Similar results have been obtained for all imbalanced problems and the AdaBoost fails drastically whenever the classes are imbalanced. Figure 4 shows the confusion matrix, precision and recall for AdaBoost without prior. It is clear from this result that AdaBoost fails to learn any pattern in this case. This is a typical class skew problems that causes most learning algorithms to over fit. Figure 5 shows the results of AdaBoost-P for the same problem. The prior incorporation method has a magical effect in this case and the ensemble created has significantly improved recall and precision.

The second and third set of experiment compare the performance of AdaBoost with and without prior in case of a balanced problem created by sampling. The comparison for second set of experiment is presented in figures 6 and 7 whereas the comparison for the third set of experiments is presented in figures 8 and 9 respectively. In the second experiment the road class has about 45% representation in the data whereas in the last experiment road samples form around 55% of the data. In both these cases the prior has affected the ensemble positively and an improved test accuracy, precision and recall has been obtained.

V. CONCLUSION

A novel method, AdaBoost-P, of incorporating prior into boosting based ensemble learning has been presented. The effectiveness of proposed method for improving the accuracy and convergence rate is shown empirically. The presented results show that AdaBoost-P is more effective especially when prior is relatively accurate and decision stump is used as learning algorithm. The presented case study is shows significantly improved accuracy measures for the case of road segmentation in Ariel images. The task is extremely difficult and ensemble method without prior did not learn any pattern whereas an amazing improvement is obtained using a simple prior based on color and road structure assumptions only

REFERENCES

[1] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, Aug. 1997.

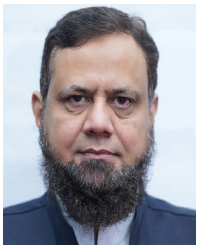
[2] R. E. Schapire, "The strength of weak learnability," *J. Mach. Learn.*, vol. 5, no. 2, pp. 197–227, Jun. 1990.
[3] L. G. Valiant, "A theory of the learnable," *Commun. ACM*, vol. 27, no. 11, pp. 1134–1142, Nov. 1984.
[4] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Mach. Learn.*, vol. 37, no. 3, pp. 297–336, Dec. 1999.
[5] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: A statistical view of boosting (with discussion and a rejoinder by the authors)," *Ann. Statist.*, vol. 28, no. 2, pp. 337–407, Apr. 2000.
[6] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Statist.*, vol. 29, no. 5, pp. 1189–1232, Oct. 2001.
[7] R. E. Schapire, M. Rochery, M. Rahim, and N. Gupta, "Incorp orating prior knowledge into Bo osting," in *Proc. 9th Int. Conf. Mach. Learn. (ICML)*, P. Langley, Ed. Stanford, CA, USA: Morgan Kaufmann, 2002, pp. 1207–1216.
[8] R. E. Schapire, M. Rochery, M. Rahim, and N. Gupta, "Boosting with prior knowledge for call classification," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 2, pp. 174–181, Mar. 2005.
[9] M. Rochery, R. Schapire, M. Rahim, N. Gupta, G. Riccardi, S. Bangalore, H. Alshawi, and S. Douglas, "Combining prior knowledge and boosting for call classification in spoken language dialogue," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 1, P. Langley, Ed. Stanford, CA, USA: Morgan Kaufmann, May 2002, pp. 29–32.
[10] M. Collins, R. E. Schapire, and Y. Singer, "Logistic regression, AdaBoost and Bregman distances," *Mach. Learn.*, vol. 48, no. 2, pp. 253–285, Jul. 2002.
[11] V. Mnih, *Machine Learning for Aerial Image Labeling*. Toronto, ON, Canada: Univ. Toronto, 2013.
[12] M. Baig and M. M. Awais, "Global reweighting and weight vector based strategy for multiclass boosting," in *Proc. Int. Conf. Neural Inf. Process.*, P. Langley, Ed. Stanford, CA, USA: Morgan Kaufmann, 2012, pp. 452–459.
[13] T. H. J. Friedman and R. Tibshirani, "Additive logistic regression: A statistical view of boosting," *Ann. Statist.*, vol. 38, no. 2, pp. 337–374, 2000.
[14] C. A. L. Bailer-Jones and K. Smith, "Combining probabilities," Data Process. Anal. Consortium (DPAC), Max Planck Inst. Astron., Heidelberg, Germany, Tech. Rep. GAIA-C8-TN-MPIA-CBJ-053, 2011.
[15] T. Hastie, S. Rosset, J. Zhu, and H. Zou, "Multi-class AdaBoost," *Statist. Interface*, vol. 2, no. 3, pp. 349–360, 2009.
[16] A. Frank and A. Asuncion, "UCI machine learning repository," in *Cognitive Skills and Their Acquisition*, J. R. Anderson, Ed. Irvine, CA, USA: Univ. California, School Inf. Comput. Sci., 2010, ch. 1, pp. 1–51.
[17] M. Yang, Y. Yuan, and G. Liu, "SDUNet: Road extraction via spatial enhanced and densely connected UNet," *Pattern Recognit.*, vol. 126, Jun. 2022, Art. no. 108549.
[18] V. Chaudhary, P. K. Buttar, and M. K. Sachan, "Satellite imagery analysis for road segmentation using U-Net architecture," *J. Supercomput.*, vol. 78, no. 10, pp. 12710–12725, Jul. 2022.
[19] H. Xu, H. He, Y. Zhang, L. Ma, and J. Li, "A comparative study of loss functions for road segmentation in remotely sensed road datasets," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 116, Feb. 2023, Art. no. 103159.
[20] V. Mnih, "Machine learning for aerial image labeling," Ph.D. dissertation, Graduate Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, 2013.

MUBASHER BAIG received the M.Sc. degree in mathematics from the University of the Punjab, in 1997, and the M.S. and Ph.D. degrees in computer sciences from the Lahore University of Management Sciences (LUMS), in 2009 and 2016, respectively.

He is currently an Assistant Professor with the National University of Computer and Emerging Sciences (NUCES-FAST), Lahore, Pakistan. His research interests include artificial intelligence and machine learning.

TAHIR EJAZ received the M.S. degree in computer science degree from the Lahore University of Management Sciences (LUMS), Lahore, and the Ph.D. degree from TU Dortmund University, in 2010. He was awarded a fully-funded Ph.D. scholarship from the Government of Pakistan. He is currently an Assistant Professor with the Department of Computer Science, National University of Computer and Emerging Sciences (NUCES-FAST). His research interests include the design and analysis of algorithms, data mining, and machine learning.

KHAWAJA M. FAHAD received the B.S., M.S., and Ph.D. degrees in computer science from the Lahore University of Management Sciences (LUMS), Lahore. He is currently an Assistant Professor with the Department of Computer Science, National University of Computer and Emerging Sciences (NUCES-FAST). His research interests include the design and analysis of algorithms, data mining, and machine learning.



SYED ASIF MEHMOOD GILANI received the Ph.D. degree in digital imaging from the University of Patras, Greece, in 2002.

He is currently a Professor with the Department of Computer Science, FAST National University of Computer and Emerging Sciences, Lahore, Pakistan. He has supervised numerous M.S. and Ph.D. students in the area of digital image processing and computer vision. His research interests include digital image processing, computer vision, and multimedia data security.

MIAN M. AWAIS (Senior Member, IEEE) received the Ph.D. degree from Imperial College London.

Prior to joining the Lahore University of Management Sciences (LUMS), he conducted European Union research and development projects for a U.K.-based SME. His Ph.D. work is related to the development of online models for parametric estimation of solid fuel-fired industrial boilers. He is currently a Professor of computer science with the Department of Computer Science, Syed Babar Ali School of Science and Engineering, LUMS. His research interests include artificial intelligence, applied computational intelligence, and machine learning.

SANA SAEED received the M.S. degree from the National University of Computers and Emerging Sciences. Her research interest includes applied machine learning.