Journal of Information Hiding and Multimedia Signal Processing ©2018 ISSN 2073-4212
Ubiquitous International Volume **9**, Number **3**, May **2018**

A Study of Mammogram Classification using AdaBoost with Decision Tree, KNN, SVM and Hybrid SVM-KNN as Component Classifiers

Jyoti Deshmukh and Udhav Bhosle

Department of Electronics and Telecommunication Engineering
Rajiv Gandhi Institute of Technology, Andheri (w), Mumbai-400053
University of Mumbai, INDIA
{jyoti.deshmukh, udhav.bhosle}@ mctrgit.ac.in
Received August, 2017; revised January, 2018

Abstract. *In this article, authors focus on designing an algorithm using Hybrid SVM-KNN as component classifier for AdaBoost. To obtain a set of effective SVM-KNN component classifiers, algorithm adaptively adjusts the kernel parameters of KNN using optimized parameters of SVM to form Hybrid SVM- KNN as component classifier. In the proposed method, first SVM is applied on training samples with $\sigma = 1000$. Initially equal weights are assigned to each tuple and updated weights are obtained. The updated weights from SVM are used as initial weights for KNN. Weighted average of predictions from both the models is obtained. Mammograms are represented by Grey Level Co-occurrence Matrix (GLCM) texture features. High dimensionality of feature space reduces the classification accuracy. Extraction of relevant features to distinguish between benign and malignant class is important to improve the classification performance. Authors use K-means clustering algorithm for features optimization to remove the redundant and irrelevant features from the original feature set. The performance of the AdaBoost with DT, KNN and SVM as component classifier is compared with AdaBoost with Hybrid SVM-KNN in terms of accuracy and area under Receiver Operating Characteristic curve (ROC) value. Mammogram images from standard Digital Database for Screening Mammography (DDSM) and Mammography Image Analysis Society (MIAS) datasets are used. The experimental results demonstrated that AdaBoost with Hybrid SVM-KNN as a component classifier gives better performance compared to other ensemble methods for mammogram classification.*
Keywords: Mammogram, AdaBoost, DT, KNN, SVM, SVM-KNN, Component classifier.

1. **Introduction.** Every day large numbers of mammograms are generated. These are analyzed and interpreted by relatively very few radiologists. So, use of computerized analysis of mammogram helps to reduce radiologists load and helps in early detection of breast cancer. Biopsy is used for identification of suspicious areas in the human breast. Biopsy is a painful method and out of all biopsy-cases, only 20-30 % cases are malignant. Thus computer aided diagnosis will help to lower down the unnecessary biopsies.

AdaBoost is mostly used as boosting method [1]. AdaBoost incorporates set of component classifiers through allocation of weights to input training samples. Initially it assigns equal weight to all training samples, and then after every boosting round, adaptively weights are adjusted. There is increase in weight of training samples, if they are wrongly classified by current component classifier, and decrease in weight if correctly classified. Different methods to perform weight update in AdaBoost are reported in [2].

Decision Tree (DT) [3], and Neural Network [4] are employed as component classifiers in AdaBoost. These researches present remarkable generalization performance of AdaBoost. In [5], authors present AdaBoost with properly designed RBFSVM component classifier. An effective classifier ensemble to overcome FP reduction problem in CADe by learning AdaBoost with different feature representation is reported in [6]. Decision trees are very good fit for AdaBoost. In decision tree parameter tuning is not required for boosting. DT's are fast to train and classify, which is needed in boosting, when 100s or 1000s to run before outputting the final decision [7]. KNN classifies objects into separate classes depending on nearest training samples in the neighbourhood. KNN classifier is used to estimate the class of an unknown instance from its neighboring instances. KNN caches all possible samples for the classification of new sample depending on the distance metric and returns the most common value. KNN reserve all training tuples in pattern space. The class of test tuple is decided by a majority vote of its neighboring tuples in the training pattern space [8]. Discrimination of mammograms either normal or abnormal is performed using deep learning in [9] and [10]. Convolutional Neural Network-Discrete Wavelet (CNN-DW) and Convolutional Neural Network-Curvelet Transform(CNN-CT) is used for classification in [10]. Multi-view deep residual neural network (mResNet) is employed for the fully automated mammogram classification in [11].

Authors propose AdaBoost with Hybrid SVM-KNN as component classifiers for mammogram classification by combining good features of KNN and SVM. DT, KNN and SVM are also implemented as component classifiers in AdaBoost for mammogram classification. In preprocessing step, GLCM features are optimized using K-means clustering algorithm. It removes the redundant and irrelevant features from the original feature set. The performance of the AdaBoost with DT, KNN, SVM and Hybrid SVM-KNN is compared with respect to accuracy and area under ROC curve (AUC) value. The paper outline is described as below. Proposed method is explained in Section 2, followed by proposed Hybrid SVM-KNN component classifier starting with AdaBoost with DT, SVM and KNN in Section 3. Experimental results are presented in Section 4. Conclusion and future work is reported in Section 5.

2. **Proposed Method.** Authors propose semi-supervised K-means clustering algorithm for feature optimization and Hybrid SVM-KNN classifier as component classifier in ensemble method for mammogram classification. In pre-processing mammogram is segmented for procuring region of interest (ROI). On ROI, histogram equalization is employed for enhancement. Mammograms are represented by GLCM texture features [12].

2.1. **Optimization of Texture Features.** Most discriminating features are identified from the original feature set by K-means clustering [7] algorithm. The feature vector and class label of the input mammograms (benign or malignant), are submitted to the optimization algorithm. Semi-supervised K-means clustering algorithm (as the class label of input mammogram is provided) is employed in optimization process. Feature optimization with clustering algorithm is described in Algorithm 1.

Algorithm1: Feature optimization with clustering algorithm

1. **Input**: GLCM features of all training mammograms with class labels (benign or malignant); $\{F_{ij}, C_k\}$ where i and j represents mammogram and its features ; k ϵ {0, 1 i.e. 0 for benign and 1 for malignant}
2. **Initialize**: number of clusters k = number of mammogram class = 2

(a) For column = 1 to 140
Apply K means clustering algorithm with K=2, to minimize objective function

$$J(V) = \sum_{l=1}^{C} \sum_{m=1}^{C_l} (\|X_l - V_m\|)^2$$

Where $||X_l - V_m||$ is the Euclidian distance between X_l and V_m
C is number of cluster centers, C_l is the number of data points in the l^{th} cluster.

(b) Compare data points of every column in each cluster with their original class.
(c) Obtain count for each feature in correctly clustered data points.
(d) Set a threshold for the minimum count of correctly clustered data points.
(e) Select features with count of correctly clustered data points above the threshold value. These features are considered as decisive features.
(f) Remove the features with wrongly clustered data points. These are considered as outliers and are removed from the feature space.

3. **Output**: Returns the optimized and consistent features.

3. **AdaBoost Algorithm.** AdaBoost [13] assigns a weight distribution w to training mammograms. Initially, the weights are uniform. In series of iterations, AdaBoost calls component classifiers. At iteration r, AdaBoost assigns training mammograms with a distribution w_r to component classifier. Component classifier h_r gets trained with training samples. After each iteration, distribution w_r is updated corresponding to classification results of training mammograms. Mammograms which are accurately classified and misclassified is assigned smaller and larger weights respectively. AdaBoost concentrates the mammograms which are hard to classify for component classifier. This step carried for R iterations. Lastly AdaBoost incorporates all the component classifier in one final hypothesis h. Component classifiers having lower training errors are assigned larger weights.

3.1. **AdaBoost algorithm with Decision Tree (DT) as component classifier.** For the input tuple X of unknown class, its attribute values are tested using DT [7] to predict its class by tracing a path from the root to a leaf node. Classification rules are derived from DT to provide tuple class information. Decision tree handles high dimensional data without prior domain knowledge. To identify the attributes, that correctly allocate the tuple into distinct classes, attribute selection measures are employed during tree construction.

Algorithm 2: AdaBoost with DT as component classifier

1. **Input:** Mammogram images with class label (benign or malignant) i.e. (X_1, C_1), (X_2, C_2),..., (X_n, C_n); Feature pool F= $\{f_m$, m=1,..., n$\}$; Number of iterations = R
2. **Initialization:** Weight of each mammogram = $\frac{1}{N}$; $\forall$ i (i= 1,..., N)
3. **For r = 1 to R do**
   (a) Generate a training set by sampling with $\{w_i(r)\}$
   (b) Train base classifier h_r (DT Classifier) using this training set
   /* DT Classification Algorithm */
      (i) Choose the best attributes(s) to split the remaining instances to make a decision node
      (ii) Repeat this process recursively for each child
      (iii) Stop when
         (A) All the instances have the same target attribute value
         (B) There are no more attributes
         (C) There are no more instances

(c) Compute the training error of h_r

$$\epsilon_r = \sum_{i=1}^{N} w_i(r).I[C_i \neq h_r(X_i)]$$

where I ϵ (-1, 1), I_A is indicator of A; we assume (ϵ_r <0.5)

(d) Set

$$\alpha_r = log(\frac{1-\epsilon_r}{\epsilon_r})$$

(We have $\alpha_r > 0$)

(e) Update the weights by

$$w_i^{'}(r+1) = w_i(r)exp(\alpha_r I[C_i \neq h_r(X_i)])$$

$$w_i(r+1) = \frac{w_i^{'}(r+1)}{\sum_i w_i^{'}(r+1)}$$

4. **Output:**

$$h(x) = Sign(\sum_{r=1}^{R} \alpha_r h_r(X))$$

3.2. **AdaBoost algorithm with K-Nearest Neighbour (KNN) as component classifier.** KNN classifier [8] uses learning by analogy concept. It compares test mammogram with training mammograms closer to it. KNN represents training mammogram with n attributes. KNN finds the K nearest training mammograms to test mammogram in the pattern space. K training mammograms becomes the K nearest neighbours for the test mammogram. Among K nearest neighbours, most common class is assigned to test mammogram.

Algorithm 3: AdaBoost with KNN as component classifier

1. **Input:** I:Set of all training mammograms with class label: I = {(I_1, C_1), (I_2, C_2),..., (I_N, C_N)}; I_t: Represent test mammogram; I_i = {f_{i1}, f_{i2},..., f_{in}} represent the feature vector of I_i training mammogram; I_t = {f_{t1}, f_{t2},..., f_{tn}} represent the feature vector of I_t test mammogram; K= 10; Number of iterations = R
2. **Initialization:** Weight of each mammogram = $\frac{1}{N}$; $\forall$ i (i= 1,..., N)
3. **For r = 1 to R do**
   (a) Generate a training set by sampling with {w_i(r)}
   (b) Train base classifier h_r (KNN Classifier) using this training set
   /* KNN Classification Algorithm */
   (i) Add all training mammograms to pattern space.
   (ii) Find Euclidian distance d(I_i, I_t) between all training and test mammogram

$$d(I_i, I_t) = \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{n}(f_{ij} - f_{tj})^2}$$

   (iii) Find first K training mammograms closest to test mammogram with respect to Euclidian distance.
   (iv) Find the class that represents the maximum of the K mammogram.
   (v) Return the majority class and assign the test mammogram to majority class.

(c) Compute the training error of h_r

$$\epsilon_r = \sum_{i=1}^{N} w_i(r).I[C_i \neq h_r(X_i)]$$

where I ϵ (-1, 1), I_A is indicator of A; we assume (ϵ_r <0.5)

(d) Set

$$\alpha_r = log(\frac{1-\epsilon_r}{\epsilon_r})$$

(We have $\alpha_r > 0$)

(e) Update the weights by

$$w_i^{'}(r+1) = w_i(r)exp(\alpha_r I[C_i \neq h_r(X_i)])$$

$$w_i(r+1) = \frac{w_i^{'}(r+1)}{\sum_i w_i^{'}(r+1)}$$

4. **Output:**

$$h(x) = Sign(\sum_{r=1}^{R} \alpha_r h_r(X))$$

3.3. **AdaBoost algorithm with Support Vector Machine (SVM) as component classifier.** An optimal separating hyper plane is found by SVM in the feature space. RadialBasis Function (RBF) is used in this work to nonlinearly map samples to a high-dimensional feature space.

Algorithm 4: AdaBoost with SVM as component classifier

1. **Input:** Mammogram images with class label (benign or malignant) i.e. (X_1, C_1), (X_2, C_2),..., (X_n, C_n); Feature pool F= {f_m, m=1,..., n}; Number of iterations = R
2. **Initialization:** Weight of each mammogram = $\frac{1}{N}$; $\forall$ i (i= 1,..., N); $\sigma = 1000$
3. **For r = 1 to R do**
   (a) Generate a training set by sampling with {w_i(r)}
   (b) Train base classifier h_r (SVM Classifier) using this training set

$$SVM(RBF)kernelfunction : k(X_i, X_j) = exp(-\frac{\|X_i - X_j\|^2}{2\sigma^2})$$

   (c) Compute the training error of h_r

$$\epsilon_r = \sum_{i=1}^{N} w_i(r).I[C_i \neq h_r(X_i)]$$

where I ϵ (-1, 1), I_A is indicator of A; we assume (ϵ_r <0.5)

   (d) Set

$$\alpha_r = log(\frac{1-\epsilon_r}{\epsilon_r})$$

(We have $\alpha_r > 0$)

   (e) Update the weights by

$$w_i^{'}(r+1) = w_i(r)exp(\alpha_r I[C_i \neq h_r(X_i)])$$

$$w_i(r+1) = \frac{w_i'(r+1)}{\sum_i w_i'(r+1)}$$

4. **Output:**

$$h(x) = Sign(\sum_{r=1}^{R} \alpha_r h_r(X))$$

3.4. **AdaBoost algorithm with proposed Hybrid SVM-KNN algorithm as component classifier.** Author proposed Hybrid SVM- KNN classifier as component classifiers in AdaBoost by combining features of KNN and SVM.

Algorithm 5: AdaBoost with Hybrid SVM - KNN as component classifier

1. **Input:** Mammogram images with class label (benign or malignant) i.e. (X_1, C_1), (X_2, C_2),..., (X_n, C_n); Feature pool F= {f_m, m=1,..., n}; Number of iterations = R
2. **Initialization:** Weight of each mammogram = $\frac{1}{N}$; ∀ i (i= 1,..., N); $\sigma = 1000$

$$SVM(RBF)kernelfunction : k(X_i, X_j) = exp(-\frac{\|X_i - X_j\|^2}{2\sigma^2})$$

3. **For r = 1 to R do**
   (a) Generate a training set by sampling with {w_i(r)}
   (b) Train base classifier h_r (Proposed Hybrid SVM - KNN Classifier) using this training set
   /* Proposed Hybrid SVM - KNN Classification Algorithm */
   (i) Apply SVM classifier on mammogram data set with K-fold cross-validation and K=10.
   (ii) Update the weights.
   (iii) According to Wolfe dual form, weight minimization is

$$Minimize : w(\alpha) = -\sum_{i=1}^{N} \alpha_i + \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} y_i y_j \alpha_i \alpha_j k(X_i, X_j)$$

$$Subject - to : \sum_{i=1}^{N} y_i \alpha_i = 0, \forall_i; 0 \leq \alpha_i \leq C$$

   (iv) Predict the test mammogram class using the cross validated model with minimum weight.
   (v) Apply weighted K-Nearest Neighbor Classifier with number of nearest neighbors K=10 on mammogram data set.
   (vi) Apply K-fold cross validation with K=10.
   (vii) Weight contribution of each k neighbor
   (viii) Set initial weights of KNN = updated minimum weights of SVM.
   (ix) X_t is test mammogram image

$$\hat{f}(X_t) \leftarrow \frac{\sum_{i=1}^{k} w_i f(X_i)}{\sum_{i=1}^{k} w_i}$$

   (x) Predict the test mammogram class using the cross validated model with minimum weight.
   (xi) Take weighted average of predictions from both the models.

(c) Compute the training error of h_r

$$\epsilon_r = \sum_{i=1}^{N} w_i(r).I[C_i \neq h_r(X_i)]$$

where I ϵ (-1, 1), I_A is indicator of A; we assume (ϵ_r <0.5)

(d) Set

$$\alpha_r = log(\frac{1-\epsilon_r}{\epsilon_r})$$

(We have $\alpha_r > 0$)

(e) Update the weights by

$$w_i^{'}(r+1) = w_i(r)exp(\alpha_r I[C_i \neq h_r(X_i)])$$

$$w_i(r+1) = \frac{w_i^{'}(r+1)}{\sum_i w_i^{'}(r+1)}$$

4. **Output:**

$$h(x) = Sign(\sum_{r=1}^{R} \alpha_r h_r(X))$$

4. **Experimental Results.** Experiment is performed in MATLAB environment. Mammograms from MIAS [14] and DDSM [15] database are used. Experiments are carried out on 64 bit I5, 2.50 GHZ processor with 4 GB RAM. Total 320(training=288, testing=32) abnormal mammograms from DDSM and total 110(training=85, testing=25) abnormal mammograms from MIAS database are used. ROI is obtained by cropping the mammogram. From MIAS and DDSM database center of abnormal area of mammogram is obtained, and a square are of size [256 x 256] is cropped with the center point.

Feature vector is generated using GLCM method. GLCM represents texture of mammogram by Co-occurrence matrix. Co-occurrence matrices are found for 0^0, 45^0, 90^0, 135^0 directions and 1, 2, 3, 4, 5 distances. Seven texture features namely step, variance, entropy, energy, homogeneity, 3^{rd} moment and inverse variance are calculated. Thus, each mammogram has feature vector of size $[140 \text{ X } 1]^T$ [12]. From 140 original features, 9 for DDSM and 10 for MIAS most discriminative features are obtained by employing semi-supervised K-means clustering algorithm. Features with count of correctly clustered instances above the threshold value of 60 % are considered as decisive features. Thus, it gives optimized feature vector. Table 1 and Table 2 list the sample feature vector and optimized feature vector for DDSM database. Table 3 shows the comparison of AdaBoost

Table 1. Feature vector

Image No.	Features (1-140)								
(1-288)	1	2	3		136	137	138	139	140
1	261632	261120	260096	...	4.33	5.32	5.64	5.60	5.83
2	261632	261120	260096	...	7.244	8.94	9.16	9.24	9.30
3	261632	261120	260096	...	3.64	4.54	4.71	4.70	4.70
4	261632	261120	260096	...	8.22	9.84	9.955	10.00	10.08
5	261632	261120	260096	...	6.40	8.58	9.16	9.52	9.71
:	:	:	:	:	:	:	:	:	:
288	261632	261120	260096	...	2.34	3.22	3.2622	3.39	3.41

Table 2. Optimized feature vector

Image No. (1-288)	9 Optimized Features (26-136)								
	26	27	32	37	66	72	76	86	136
1	0.99	0.99	0.82	0.99	4.33	1.03	0.98	1.06	0.98
2	0.98	0.99	1.43	0.98	7.24	1.74	0.98	1.74	1.64
3	0.99	0.99	0.69	0.99	3.64	0.88	0.99	0.89	0.87
4	0.98	0.99	1.63	0.98	8.22	1.91	0.98	1.91	1.85
5	0.98	0.99	1.20	0.98	6.4	1.67	0.98	1.64	1.54
:	:	:	:	:	:	:	:	:	:
288	0.99	0.99	0.39	0.99	2.34	0.62	0.99	0.55	0.542

Table 3. Comparative analysis of AdaBoost with DT, KNN, SVM and Hybrid SVM-KNN component classifiers

No. of Itra-tions	AdaBoost with DT			AdaBoost with KNN			AdaBoost with SVM			AdaBoost with Hybrid SVM-KNN		
	Acc-uracy %	AUC	Error	Acc-uracy %	AUC	Error	Acc-uracy %	AUC	Error	Acc-uracy %	AUC	Error
10	81.25	0.855	0.450	81.27	0.921	0.338	62.50	0.455	0.34	87.50	0.945	0.24
20	81.25	0.855	0.413	87.50	0.935	0.318	65.62	0.586	0.32	87.50	0.945	0.23
30	81.25	0.855	0.412	87.50	0.950	0.319	62.50	0.832	0.28	87.50	0.968	0.15
40	81.25	0.863	0.412	87.50	0.942	0.312	65.62	0.605	0.28	84.37	0.931	0.16
50	81.25	0.863	0.413	87.50	0.950	0.305	68.75	0.707	0.34	87.50	0.937	0.23
60	81.25	0.863	0.412	87.50	0.949	0.312	65.62	0.460	0.29	87.50	0.878	0.20
70	81.25	0.863	0.412	87.50	0.947	0.318	71.87	0.398	0.28	87.50	0.968	0.19
80	81.25	0.925	0.370	87.50	0.943	0.315	68.75	0.367	0.39	87.50	0.962	0.17
90	81.25	0.925	0.369	87.50	0.953	0.315	71.87	0.515	0.35	87.50	0.968	0.15
100	**87.50**	**0.925**	**0.367**	**90.62**	**0.960**	**0.302**	**84.37**	**0.906**	**0.18**	**90.62**	**0.984**	**0.15**
110	81.25	0.925	0.365	90.62	0.953	0.315	81.25	0.605	0.24	90.62	0.962	0.16
120	84.37	0.917	0.362	87.50	0.945	0.318	71.87	0.558	0.25	87.50	0.953	0.20
130	78.12	0.798	0.310	87.50	0.947	0.315	78.12	0.644	0.28	87.50	0.933	0.25
140	78.12	0.890	0.310	87.50	0.945	0.325	81.25	0.632	0.28	87.50	0.933	0.16
150	**81.25**	**0.878**	**0.315**	**87.50**	**0.947**	**0.325**	**71.87**	**0.339**	**0.30**	**87.50**	**0.926**	**0.25**

with DT, KNN, SVM and proposed Hybrid SVM-KNN in terms of number of iterations, accuracy, AUC, and error measure for DDSM database. Experimentation is performed starting with 10 components classifier and went up to 150. It is observed that AdaBoost with DT, KNN, SVM and proposed Hybrid SVM-KNN gives better performance at iteration number 100. So, rounds (iterations)=100 is used for AdaBoost algorithm with DT, KNN, SVM and Hybrid SVM-KNN classifiers respectively. Table 4 lists performance measures of AdaBoost with DT, KNN, SVM and proposed Hybrid SVM-KNN for DDSM and MIAS database. The performances of these classifiers are compared with respect to sensitivity, specificity, precision, recall, accuracy and AUC value. Result shows that AdaBoost with proposed Hybrid SVM-KNN gives better performance compared to AdaBoost with DT, KNN, and SVM in terms of accuracy and AUC.

TABLE 4. Performance measures of the classifiers for DDSM and MIAS dataset

Data set	Classifier	Performance measures					
		Sensitivity	Specificity	Precision	Recall	Accuracy	AUC
DDSM	DT	81.25%	93.75%	92.85%	81.25%	87.5%	0.925
	KNN	93.75%	87.5%	88.23%	93.75%	90.625%	0.960
	SVM	87.5%	81.25%	82.35%	87.5%	84.375%	0.906
	SVM-KNN	**100**	**81.25%**	**84.21%**	**100%**	**90.625%**	**0.984**
MIAS	DT	42.85%	45.45%	50%	42.85%	44%	0.743
	KNN	35.71%	54.54%	50%	35.71%	44%	0.457
	SVM	57.14%	90.90%	88.88%	57.14%	72%	0.675
	SVM-KNN	**85.71%**	**72.72%**	**80%**	**85.71%**	**80%**	**0.781**

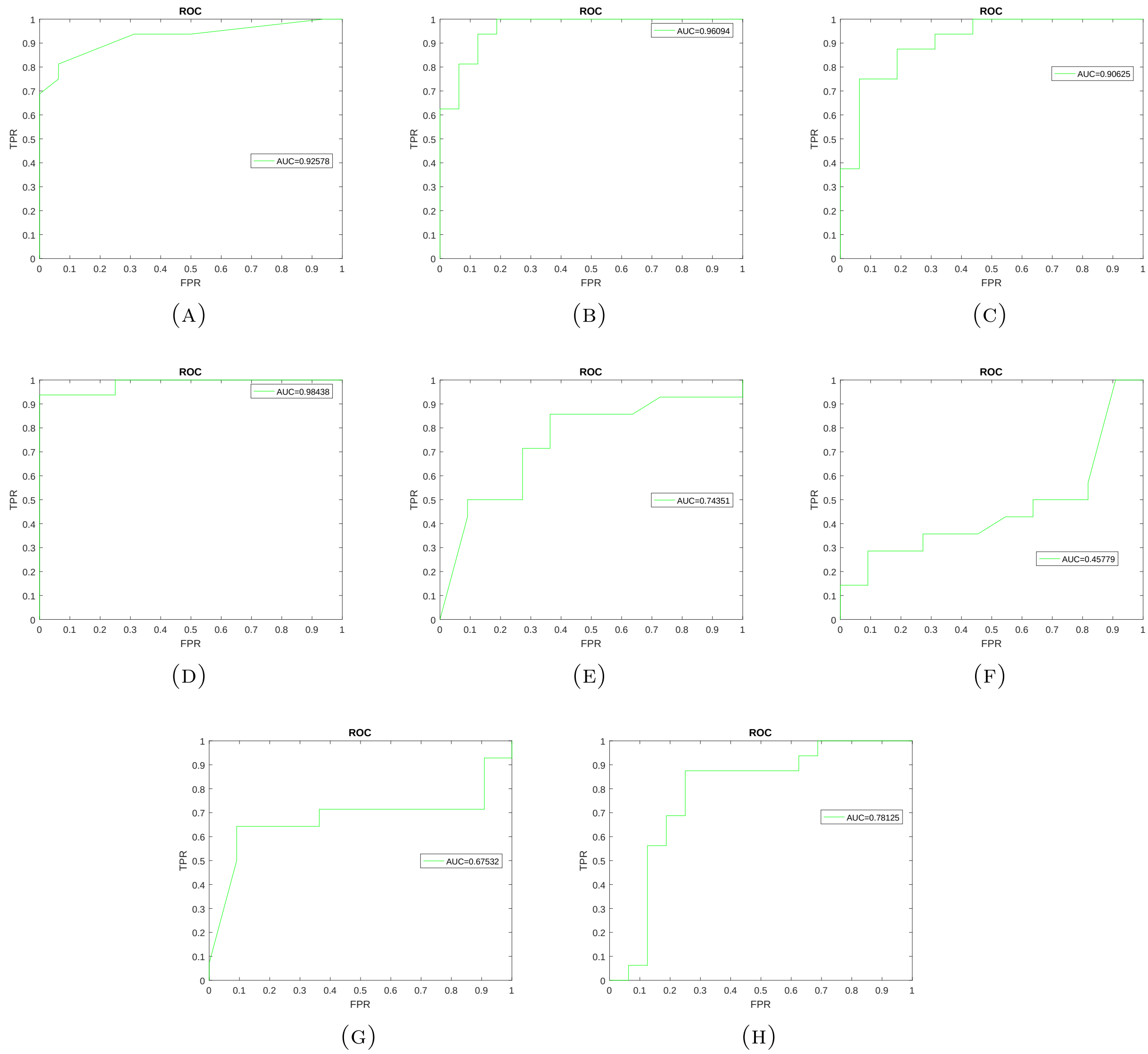

FIGURE 1. (A)to (D) and (E) to (H) are ROC curves of AdaBoost with DT, KNN, SVM and Hybrid SVM-KNN for DDSM and MIAS data respectively

The proposed Hybrid SVM-KNN classifier for large dataset(DDSM) is compared with other classifiers reported in literature. The ensemble classifier in [6] is based on combination of different features, reported 0.878 AUC. Deep learning using statistical and textual features [9] reported 87.3% accuracy and 0.9 AUC. CNN-WT and CNN-CT based Deep learning with dense scale invariant features in [10] reported 81.83% accuracy, 0.846 AUC and 83.74% accuracy, 0.855 AUC for CNN-WT and CNN-CT respectively. Multi-view deep residual neural network (mResNet) method [11] reported 0.8 AUC. Proposed AdaBoost with Hybrid SVM-KNN achieves 90.625% accuracy and 0.984 AUC. It exhibits that the proposed technique achieves better results compared to above methods. Figure 1 (A) to (D) and (E) to (H) shows the ROC plot for AdaBoost with DT, KNN, SVM, and proposed Hybrid SVM-KNN for DDSM and MIAS dataset.

5. **Conclusions.** Authors propose AdaBoost with Hybrid SVM-KNN as component classifier for mammogram classification. Proposed scheme gives classification accuracy of 90.625% for DDSM and 80% for MIAS images respectively, and 0.9843 AUC for DDSM and 0.7812 AUC for MIAS images respectively. Results of proposed scheme are compared with AdaBoost with DT, KNN and SVM as weak classifier. Results reveal that proposed AdaBoost with Hybrid SVM-KNN outperforms other ensemble methods. Future work includes more theoretical analysis and comparison with AdaBoost with all weak learners.

Acknowledgments. The authors express sincere gratitude to Dr. Thomas Deserno, Dr. J. Suckling and co-authors, for permitting the use of 'DDSM', 'MIAS'dataset respectively.

REFERENCES

[1] Y. Freund, R. E. Schapire, A desicion-theoretic generalization of on-line learning and an application to boosting, in *European conference on computational learning theory*, Springer, Berlin, Heidelberg, pp. 23-37, 13 Mar 1995.

[2] L. Kuncheva, C. Whitaker, Using diversity with three variants of boosting: Aggressive, conservative, and inverse, *Multiple Classifier Systems*, pp. 717-20, 2002.

[3] T. G. Dietterich, An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization, *Machine learning*, vol.40, no.2, pp.139-57, 1 Aug 2000.

[4] H. Schwenk, Y. Bengio, Boosting neural networks, *Neural Computation*, vol.12, no.8, pp. 1869-87, Aug 2000,.

[5] X. Li, L. Wang, E. Sung, AdaBoost with SVM-based component classifiers, *Engineering Applications of Artificial Intelligence*, vol.21, no.5, pp. 785-95, 31 Aug 2008.

[6] J. Y. Choi, D. H. Kim, K. N. Plataniotis, Y. M. Ro, Classifier ensemble generation and selection with multiple feature representations for classification applications in computer-aided detection and diagnosis on mammography, *Expert Systems with Applications*, vol.46, pp.106-21, 15 Mar 2016.

[7] J. Han, J. Pei, and M. Kamber, Data mining: concepts and techniques, Elsevier, 2011.

[8] N. S. Altman, An introduction to kernel and nearest-neighbor nonparametric regression, *The American Statistician*, vol.46, no.3, pp. 175185, 1992.

[9] J. Wang, X. Yang, H. Cai, W. Tan, C. Jin, and L. Li, Discrimination of breast cancer with microcalcifications on mammography by deep learning, *Scientific reports*, 6,pp. 27327, Jun 2016.

[10] M. M. Jadoon, Q. Zhang, I. U. Haq, S. Butt, and A. Jadoon,Three-Class Mammogram Classification Based on Descriptive CNN Features, *BioMed research international*, Jan 2017.

[11] N. Dhungel, G. Carneiro, and A. P. Bradley, Fully automated classification of mammograms using deep residual neural networks, *Biomedical Imaging (ISBI 2017), 2017 IEEE 14th International Symposium* , IEEE, pp. 310-314, April 2017.

[12] R. M. Haralick, K. Shanmugam, Textural features for image classification, *IEEE Transactions on systems, man, and cybernetics*, vol.3, no.6, pp. 610-621, 1973.

[13] R. E. Schapire, Y. Singer, Improved boosting algorithms using confidence-rated predictions, in *Proceedings of the eleventh annual conference on Computational learning theory*, ACM, pp. 80-91, 24 Jul 1998.

[14] J. Suckling et al., The mammographic image analysis society digital mammogram database, *ExerptaMedica. International Congress Series*, Vol. 1069, pp. 375-378, 1994.

[15] T. M. Deserno, M. Soiron, and J. E. de Oliveir, Texture patterns extracted from digitizes mammograms of different BI-RADS classes, *Image Retrieval in Medical Applications Project, release*, vol.1, 2012.