Prediction of Breast Cancer Survivability using Ensemble Algorithms

Vincent F. Adegoke / Daqing Chen
Computer Science and Informatics, School of Engineering
London South Bank University,
London, United Kingdom
adegokev@lsbu.ac.uk / chend@lsbu.ac.uk

Ebad Banissi / Safia Barikzai
Computer Science and Informatics, School of Engineering
London South Bank University
London, United Kingdom
banisse@lsbu.ac.uk / barikzas@lsbu.ac.uk

Abstract — **In this paper, several ensemble cancer survivability predictive models are presented and tested based on three variants of AdaBoost algorithm. In the models we used Random Forest, Radial Basis Function Network and Neural Network algorithms as base learners while AdaBoostM1, Real AdaBoost and MultiBoostAB were used as ensemble techniques and ten other classifiers as standalone models. There has been major research in ensemble modelling in statistics, medicine, technology and artificial intelligence in the last three decades. This might be because of the effectiveness and reliability of the technique in medical diagnosis and incident predictions compare with the standalone classifiers. We used Wisconsin breast cancer dataset in training and testing the models. The performances of the ensemble and standalone models were evaluated using Accuracy, RMSE and confusion matrix predictive parameters. The result shows that despite the complexity of the ensemble models and the required training time, the models did not outperform most of the standalone classifiers.**

Keywords — adaboostm1; breast cancer; multiboostab; neural network; radial basis function network

I. Introduction

AdaBoost is a meta-learning method that trains several weak classifiers. It employs the knowledge of multiple and simple learners in producing a committee of classifiers. The final output of the model is a linear combination of a set of participating learners. The motive for doing this is to improve the confidence that we are making the right decision by weighing various opinions from the participating experts and combining them to reach a final decision [1]. Boosting has its roots in Probably Approximately Correct (PAC) learning framework that was introduced by Valiant [2] as cited by Schapire [3]. However, AdaBoost as a technique that combines several classifiers to form a strong classifier was first proposed by Freud and Schapire [4] in 1995. The primary goal of AdaBoost as an ensemble classifier is to improve the accuracy of the base classifier by constructing ensemble of decision rules that produce a strong classifier and perform better when they are combined than random guessing. During training the base classifiers are obtained sequentially using[1] reweighted versions of the training data taking into consideration the prediction accuracy of the previous classifier. Empirical study shows that the chosen base classifier can significantly affect the performance of AdaBoost models. Therefore, if the base learner is too strong, it may achieve high accuracy leaving only outliers and noisy instances with significant weight to be learned in the following iterations. The chosen base classifier should therefore be able to learn without significant decrease in the weight of the previously and correctly classified instances. AdaBoost has been widely used in many applications such as text classification, natural language processing, drug discovery and computational biology [5] vision and object recognition [6] industrial chemical fault diagnosis and medical diagnosis [7] such as breast cancer prediction.

Breast cancer is a disease that causes cells around the breast to change and grow out of control most of which are invasive or infiltrating. It is one of the most common causes of cancer related death among women in the world today. For example, in the USA in 2015 an estimated 231,840 new cases of invasive breast cancer were diagnosed among women and 60,290 additional cases of in situ breast cancer [8]. Also in the UK, over 55,222 women were diagnosed with new cases of the disease in 2014 which amounted to 11, 433 deaths [9] and the ailment reached 25.2% of women worldwide [10] . The disease is also a looming epidemic in the developing nations where advanced techniques for early detection and treatments are not readily available [11]. For example, WHO [12] asserted that breast cancer is the top cancer disease in women worldwide and is increasing at an alarming rate in other developing countries. It is a global leading cause of cancer deaths in most countries in Africa [13]. It is also the most common type of cancer among women in the Asian-pacific region. In 2012 alone it accounts for 18% of all cases and was the fourth most common cause of cancer related death that accounts for 9% [14].

Medically, breast cancer can be detected early during a screening examination through mammography or after a woman notices an unusual lump [8] in her breast. Due to advancement in technology and availability of patient medical records, computer aided diagnosis cancer detection systems have been developed to detect and therefore control the spread

978-1-5386-2101-1/17/$31.00 ©2017 IEEE

of the disease. Such systems rely on pattern recognition algorithms that are used to process and analyze medical information of images obtained from mammograms for diagnostic and decision making [15] [16]. Other breast images in use include ultrasound, MRI, and positron emission tomography (PET).

Recently algorithm methods have also been proposed to extract relevant patterns from patients' breast cancer datasets. For example, Yang *et al* [17] developed a genetic algorithm to detect the association of genotype frequencies of cancer cases and noncancer cases based on statistical analysis. The authors analyzed the possible breast cancer risks using odds-ratio and risk-ratio analysis. McGinley *et al* [18] applied Spiking Neural Networks algorithm as a novel tumor classification method in classifying tumors as either benign or malignant cancer. The performance of the technique was rated to outperform the existing UWB Radar imaging algorithm. In their work Fatemeh *et al* [19] proposed an algorithm based on Non-Subsampled Contourlet Transform and Super Resolution to improve the quality of digital mammography images. The authors then used AdaBoost algorithm to classify and determine the probability of a disease being a benign and malign cancer. Similarly, in breast mass cancer classification [20] the authors used computer-aided diagnosis (CAD) system for the processing and diagnosis of breast cancer. The technique was based on extreme learning machine (ELM) that eliminates interference in the preprocessing stages. In the feature selection, it uses the combination of support vector machine (SVM) and extreme learning machine (ELM). The optimal subset of feature vectors is then inputted into the combined classifiers for distinguishing malignant masses from benign ones.

In this paper, we presented and tested ensemble models using three variants of AdaBoost algorithm namely: AdaBoostM1, Real AdaBoost and MultiBoostingAB to predict the class of breast cancer. We used Random Forest, Neural Network and Radial Basis Function Network as base classifiers for predicting cancer survivability among women. We used the Wisconsin breast cancer datasets obtained from UCI repository. We also applied standalone models using 10 different classifiers and compare their performances with the ensemble models. In both cases, we applied 10-fold cross validation technique to evaluate the predictive performance of the models. The rest of this paper is organized as follows: section 2 covers basic concepts of AdaBoost algorithm. Section 3 contains variants of AdaBoost and base classifiers used in this paper. The experimental setting and methodology are discussed in section 4. Section 5 cover results and discussions. The conclusion and future works are discussed in section 6.

II. BASIC CONCEPTS OF ADABOOST

In this section, we describe briefly the theoretical background of the associated AdaBoost variants and the base algorithms that forms committee of experts used in the ensemble models.

A. Theoretical properties of AdaBoost

The success of AdaBoost have been attributed to the algorithm's ability to reduce the training error and accelerate convergence after several iterations [21]. AdaBoost properties have been covered recently in several studies [22]. Most variants of the algorithm were developed in targeting specific issues or problems such as object detection, letter recognition, text categorization, multi-class predictions, optimization issues, etc.

B. AdaBoost as a technique

The concept of AdaBoost as a meta-learning method is based on the idea that better algorithm can be created by combining multiple instances of a simple classifiers. Each instance is trained on the same training dataset with different weights assigned to each instance based on its classification accuracy. The description here follows Schapire [23] : assume we are given a number labelled training examples such that $M = \{(x_1, y_1),\ (x_2, y_2),.,(x_n, y_n)\}$ where $x_i \in \mathcal{R}^M$ and the label $y_n \in \{-1, 1\}$. On each iteration $t = 1, \dots, T$, a distribution D_t is computed over the M training examples. A given weak learner is applied to find a weak hypothesis $h_t\colon \mathcal{R} \to \{-1, 1\}$. The aim of the weak learner is to find a weak hypothesis with low weighted error ε_t relative to D_t. The final classifier H(x) as shown in Fig. 1 (Neural Network as a committee of classifiers) is computed as a weighted majority of the weak hypothesis $\mathrm{h_t}$ by vote where each hypothesis is assigned a weight α_t. The final classifier is given in (1):

$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right) \tag{1}$$

The accuracy of the hypothesis is calculated as an error measure this is given in (2):

$$\varepsilon_t = Pr_i{\sim}D_t[h_t(i) \neq\ y_i] \tag{2}$$

The weight of the hypothesis is a linear combination of all the hypotheses of the participating experts. This is given in (3):

$$\alpha_t = \frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right) \tag{3}$$

The distribution vector D_t is expressed as in (4)

$$D_{t+1}(i) = \frac{D_t(i)\exp\left(-\alpha_t y_i h_t(x_i)\right)}{Z_t} \tag{4}$$

where $\mathrm{Z_t}$ is a normalization factor such that the weights add up to 1. This makes $\mathrm{D_{t+1}}$ a normal distribution.

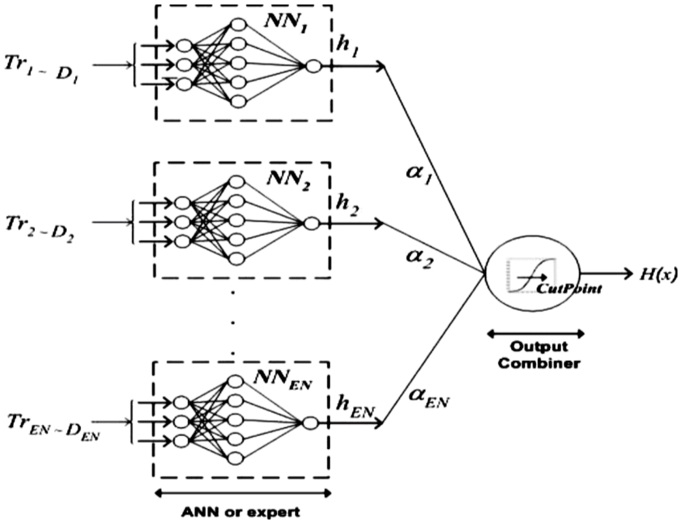
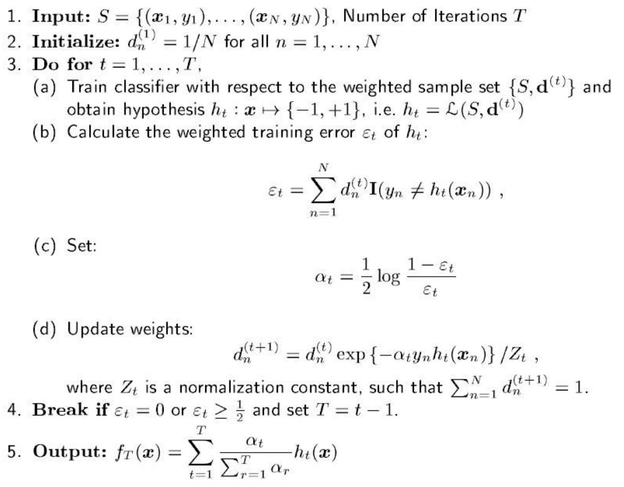


Fig. 1 A typical ensemble models showing a committee of neural network experts. Each classifier h_i has an associated weight contribution α_i

During the training process, there is always a difference from the predicted values and the expected values. The deviation or the expected error over the committee of classifiers can be expressed as in (5):

$$E_{err} = \frac{1}{N}\sum_{i=1}^{N}[y_i(x) - h_i(x)]^2 \tag{5}$$

The pseudocode for AdaBoost algorithm is as shown in Fig. 2



```
1. Input: $S = \{(\boldsymbol{x}_1, y_1), \dots, (\boldsymbol{x}_N, y_N)\}$, Number of Iterations $T$
2. Initialize: $d_n^{(1)} = 1/N$ for all $n = 1, \dots, N$
3. Do for $t = 1, \dots, T$,
    (a) Train classifier with respect to the weighted sample set $\{S, \mathbf{d}^{(t)}\}$ and
        obtain hypothesis $h_t : \boldsymbol{x} \mapsto \{-1, +1\}$, i.e. $h_t = \mathcal{L}(S, \mathbf{d}^{(t)})$
    (b) Calculate the weighted training error $\varepsilon_t$ of $h_t$:
            $\varepsilon_t = \sum_{n=1}^{N} d_n^{(t)} \mathbf{I}(y_n \neq h_t(\boldsymbol{x}_n))$ ,
    (c) Set:
            $\alpha_t = \frac{1}{2} \log \frac{1 - \varepsilon_t}{\varepsilon_t}$
    (d) Update weights:
            $d_n^{(t+1)} = d_n^{(t)} \exp\{-\alpha_t y_n h_t(\boldsymbol{x}_n)\} / Z_t$ ,
        where $Z_t$ is a normalization constant, such that $\sum_{n=1}^{N} d_n^{(t+1)} = 1$.
4. Break if $\varepsilon_t = 0$ or $\varepsilon_t \geq \frac{1}{2}$ and set $T = t - 1$.
5. Output: $f_T(\boldsymbol{x}) = \sum_{t=1}^{T} \frac{\alpha_t}{\sum_{r=1}^{T} \alpha_r} h_t(\boldsymbol{x})$
```

Fig. 2 AdaBoost as a method in training committee of classifiers that makes the final decision

III. VARIANTS OF ADABOOST & COMMITTEES OF EXPERTS

Boosting Methods

AdaBoostM1(AdaM1)

This is the original version of AdaBoost algorithms that was first introduced by Freund and Schapire [24]. It significantly generates classifiers whose performance is a little better than random guessing. As show in Fig. 1 and Fig. 2, the algorithm calls the base classifier repeatedly T times while training, where T is the number of iterations. It combines the trained classifiers to obtain the final classifier.

A. Real AdaBoost (RA)

The Real AdaBoost [25] is the generalization and improvement of the AdaBoostM1. The main difference to the standard AdaBoost involves computing and applying the estimate of the probabilities that each training pattern in the base classifier belongs to the current weight distribution. On the other hand, standard AdaBoost classifies the input patterns and compute the weighted error rate based on clarification accuracy. However, Real AdaBoost algorithm uses Newton stepping rather than exact optimization at each step.

B. MultiBoostingAB(MBAB)

This is another variant of AdaBoost technique that is used in training base classifiers to form decision committees [26]. MBAB combines Boosting with Wagging algorithms. It therefore harnesses AdaBoost high bias and Wagging superior variance reduction. Empirical study shows that the algorithm performs better than AdaBoost or Wagging on several UCI datasets when C4.5 tree was used as the base learning algorithm.

Committee of Experts

In this section, we briefly describe the architecture of the base classifiers that were used as committee of experts in our study.

A. Random Forest (RF) – Random forest is a meta-estimator algorithm that fits several decision tree classifiers on various sub-samples of the dataset. It operates by constructing an ensemble of several decision trees at training time as shown in Fig. 3. It outputs the class that is the mode of the classes as mean prediction of the individual tree [27].

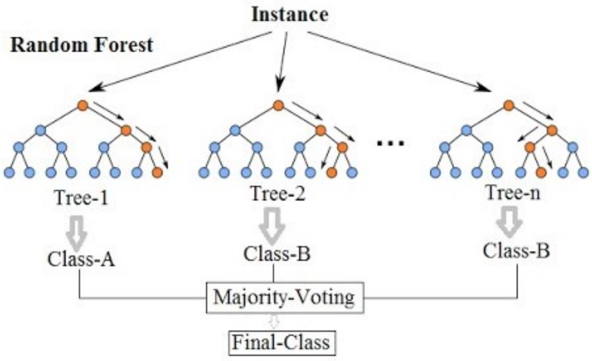


Fig. 3 Random Forest as an ensemble of several decision trees

B. Artificial Neural Network (ANN) – Artificial Neural network is a family of artificial intelligence models like the biological brain. It is capable of estimating complex and non-linear functions that depend on many inputs. ANN is generally presented as systems of interconnected neurons that exchange messages between each other. In this study, we used multilayer perception (MLP) with backpropagation which is a popular architecture of neural network algorithm [28]. The architecture of the ANN experts is as depict in Fig. 1. The network output of an MLP [28] can be expressed in (6)

$$F(x_1, \dots, x_{m_0}) = \sum_{i=1}^{m_1} \alpha_i \varphi(\sum_{j=1}^{m_0} w_{ij}x_j + b_i) \qquad (6)$$

where,

m_0 is the number of input nodes and a single layer consisting of m_1 neurons,

$x_1, \dots, x_{m_0}$ are the inputs, hidden neuron i has synaptic weights $w_{i_1}, \dots, w_{m_0}$ and bias b_i and

$\alpha_1, \dots, \alpha_{m_1}$ define the synaptic weights of the output layer.

C. Radial Basis Function Network (RBFN) **-** RBFN can be viewed as an alternative to the MLP neural network for non-linear modelling. It uses radial basis function as an activation function and has only one hidden layer. It can be trained in many ways, unlike the MLP that are typically trained with backpropagation algorithms. The network output can be expressed in (7)

$$y(x) = \sum_{j=1}^{M} w_{kj}\phi_j(w_{kj}) + w_{k0} \qquad (7)$$

where the ϕ_j are the basis functions, w_{kj} are the output layer weights and w_{k0} are the bias weights.

IV. EXPERIMENTAL SETUP AND METHODOLOGY

A. Cancer survivability datasets

During the study, all implementations were carried out using WEKA (Waikato Environment for Knowledge Analysis) [29] software. WEKA is an open data mining suite implemented in JAVA for data classification, clustering, regression and visualization. We run all simulations on an Intel Core i5-3210M CPU 2.5GHz PC, equipped with 6.00 GB of RAM Windows 8.1 64-bit machine. The algorithmic settings of the boosting methods and the base classifiers are as presented in Table 1. In the study, we used the Wisconsin cancer survivability dataset.

TABLE 1 BASIC ALGORITHMIC SETTINGS: STANDALONE AND ENSEMBLE MODELS

Parameter/ Algorithm	**Batch Size**	**Seed**	**Debug**	**Check Capability**	**Specific Parameter**
RF	100	1	False	True	RP[a] = 50
ANN	100	0	False	True	LR[b] = 0.3
RBFN	100	-	False	True	MSD[c] = 0.1
RIPPER	100	1	False	True	Folds = 3
Naïve Bayes	100	-	False	True	USD[d] = False
SMO	100	1	False	True	TP[e] = 0.001
SVM	100	1	False	True	Loss = 0.1
C4.5	100	1	False	True	CF[f] = 0.25
K-NN	100	-	False	True	MS[g] = True
Logistic Regression	100	-	False	True	UCGD[h] = False
Ensemble Models	100	1	False	True	Use Resampling

[a] Remove Percentage, [b] Learning rate, [c] Minimum standard deviation, [d] Use Supervised Discretization, [e] Tolerance Parameter, [f] Confidence Factor, [g] Mean Squared, [h] Use Conjugate Gradient Descent

The dataset was obtained from the UCI Irvine machine learning repository. The raw data comprises of 699 instances and 10 attributes. We removed 16 instances of the data with missing values to obtain 683 instances with a binary class attribute that was used. The binary class comprises of 444 (65%) instances of Begin cancer and 239 (35%) instance of malignant cancer. Table 2 shows the attributes of the dataset.

TABLE 2 ATTRIBUTES OF WISCONSIN SURVIVABILITY DATASET

No	**Attributes**	**Domain**
1	Sample code number	Id number
2	Clump thickness	1-10
3	Uniformity of cell size	1-10
4	Uniformity of cell shape	1-10
5	Marginal adhesion	1-10
6	Single epithelial cell size	1-10
7	Bare Nuclei	1-10
8	Bland Chromatin	1-10
9	Normal nucleoli	1-10

10	Mitoses	1-10
11	Class	Benign - 2 Malignant - 4

B. *Stratified 10-fold csross-validation*

In the experimental setup, we used 10-fold cross validation with each dataset divided into ten parts. Nine parts were used to train each model and one part was used for testing. We applied this method as empirical studies shows that stratified cross-validation generates comparison results with lower bias and lower variance. The process of 10-fold validation involves the following four outline steps:

a) For each fold train, the classifier using all the folds except one

b) Use the left-out fold to test the model by calculating the cross-validation metrics

c) Run the $k - fold$ cross validation several times (in our case 10 times)

d) Average the cross-validation metrics across the subsets to get the final cross validation metrics.

C. *Evaluation and Performance methods*

In evaluating the performance of the models, we applied six different statistical metrics namely the classification Accuracy, TP rate, FP rate, RMSE, Precision, ROC Area. Fig. 4 is a confusion matrix for a classifier with two classes False and True (True Positives - TP, True Negative - TN, False Positive - FP and the False Negative - FN). The accuracy of an algorithm is the percentage measure of instances that were correctly classified. The accuracy and RMSE are as represented in (8) and (9) respectively.

$$Accuraccy = \frac{TP + TN}{TP + FP + TN + FN} \tag{8}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{9}$$

The sensitivity and the specificity measurements of the models are as represented in (10) and (11) respectively.

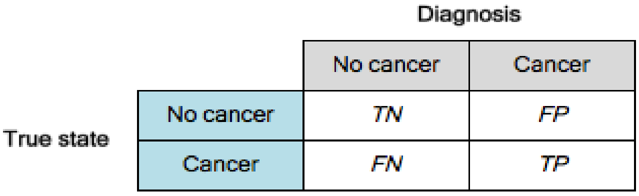
$$Se = \frac{TP}{TP + FP} \tag{10}$$

$$Sp = \frac{TN}{TN + FN} \tag{11}$$

Receiver Operator Characteristic (ROC) curves is a metric that shows how the number of correctly classified positive examples varies with the number of incorrectly classified negative examples for binary classifier when the threshold discrimination is varied. Precision Recall curves (PRC) on the other hand give a more informative of an algorithm's performance when dealing with highly skewed datasets. The true positive rate (the precision) and the false positive rate metrics are as expressed in (12) and (13) respectively.

$$Tpr = \frac{TP}{TP + FN} \tag{12}$$

$$Fpr = \frac{FP}{FP + TN} \tag{13}$$



		Diagnosis	
		No cancer	Cancer
True state	No cancer	*TN*	*FP*
	Cancer	*FN*	*TP*

Fig. 4 The confusion matrix used for two class prediction

V. Results and Discussions

A. *Classification comparison among the ensemble models*

The metric performance of the ensemble models is as illustrated in Table 3.

1. *True Positive and False Positive Rates*

The table shows that AdaM1 + RF and MBAB + RF models have the best TP rate of 0.97 compared with RA+ANN model that has the worst TP rate of 0.88. In term of FP Rate, MBAB+RBFN have the least value of 0.03 followed by AdaM1+RF and RA+RF models with 0.04. RA+ANN have the largest FP rate value of 0.22 while MBAB+RBFN models have the smallest value of 0.03.
This is illustrated in Fig. 5.

2. *Accuracy and ROC*

AdaM1+RF and MBAB+ANN models have the best accuracy prediction value of 0.97. On the other hand, RA + ANN model has the worst prediction accuracy value of 0.88. This was followed by AdaM1+ RBFN and RA + RBFN models with 0.95 prediction accuracy. The ROC area value of all the models is 0.99 apart from AdaM1+ANN and RA + ANN models that have ROC area of 0.98, and AdaM1+ RF model that has ROC area of 0.97. This is illustrated in Fig. 6.

3. *RMSE and Execution Time*

The performance of RA+ANN model in terms of RMSE metric is worst with value of 0.29 while AdaM1 + RF and MBAB + RF models have the least RMSE value of 0.17. It takes 11.73s for MBAB+ANN model to train its base classifier. This is the longest execution time for all the presented models. On the other hand, the execution time for MBAB+RF was 0.14s. The RMSE and the execution time performance for the ensemble models are as illustrated in Fig. 7 and Fig. 8 respectively.

4. *PRC Area and Precision*

The PRC area of most of the models have value of 0.99 apart from AdaM1+ANN model with PRC area of 0.97, and MBAB+ANN and RA ANN models with PRC area of 0.98. In term of Precision values RA+ANN model has the least value of 0.90 while PRC value for other models ranges from 0.95 to 0.97.

In the ensemble models, we found that MBAB+RF model performs best with accuracy value of 0.97, RMSE value of 0.17, TP Rate value of 0.97, FP rate value of 0.05, Precision value of 0.97, ROC area value of 0.99, PRC area value of 0.98 and execution time of 0.14 seconds. On the other hand, RA+ANN model performs worst with prediction accuracy of 0.88, RMSE value of 0.29, TP rate value of 0.88, FP value rate of 0.22, Precision value of 0.90, Recall curve value of 0.88, ROC area value of 0.98, PRC area value of 0.98 and execution time of 4.61seconds.

B. Classification comparisons among the single models

Table 4 shows the metric performance of the standalone models namely: Random forest, Neural network, Radial basis function, RIPPER, Naïve Bayes, SMO, SVM, C4.5, KNN and Logistics Regression.

1. True Positive and False Positive Rates

In term of TP rate RF, Naïve Bayes, SMO and Logistics Regression have the best TP rate value of 0.97 while C4.5 has the worst TP rate value of 0.94. In term of FP rate Naïve Bayes has least value 0.02 followed by ANN, C4.5 and K-NN with value of 0.06. The TP rate and FP rate performances are as illustrated in Fig. 9.

2. Accuracy and ROC

ANN and K-NN classifiers have the least Accuracy prediction value of 0.95 while accuracy prediction values of other classifiers are between 0.96 and 0.97. The ROC area of most of the classifiers ranges from 0.97 to 0.99 apart from RIPPER and C4.5 with ROC area value of 0.96. The accuracy and ROC performances of the algorithms are illustrated in Fig. 10.

3. RMSE and Execution Time

In term of RMSE metric, Table 4 shows that SVM performs best with RMSE value of 0.04 and C4.5 performs worst with RMSE value of 0.22, followed by RBF with RMSE value 0.18. The result reveals that despite the performance of ANN compare to other standalone classifiers it takes ANN 1.17s to train its base classifiers compare to C4.5 that requires 0.01s The RMSE performance and the execution time of the standalone models are illustrated in Fig. 11 and Fig. 12

4. PRC Area and Precision

As shown in the Table 3 K-NN and SMO algorithms have PRC area of 0.96 while ANN and RBBN have PRC area of 0.98. Logistic Regression and RF have PRC area of 0.99 while PRC areas of other models range between 0.94 and 0.95. The precision values of C4.5 and K-NN are 0.94 and 0.95 respectively while the precision value for other classifiers ranges from 0.96 and 0.97.

In the standalone classifiers in terms of Accuracy, RMSE, Precision and Execution time metrics ANN performs worst with 0.95, 0.2, 0.97 and 1.17 respectively. On the hand, Random Forrest perform best with Accuracy, RMSE, Precision and Execution time metrics values of 0.97, 0.17, 0.97 and 0.13 respectively.

C. Classification comparison between stand-alone classifier and committee of classifiers

The results for the standalone and the ensemble classifiers are as outlined in Table 3 and Table 4 respectively. Despite the complexity of some of the ensemble models that requires more time for training and testing, the results show that there are no significant differences or improvements in their performance metrics compared to the standalone classifiers.

VI. CONCLUDING REMARKS AND FUTURE WORK

In this paper, we presented and tested ensemble models using three variants of AdaBoost algorithms namely AdaBoostM1, Real AdaBoost and MultiAdaBoostAB. We applied Random Forest, Neural Network and Radial Basis Function Network algorithms as base learners. We compared the performance of the ensemble models with standalone classifiers.

The results of our experiment show that the success of the boosting algorithms depend on the choice of base classifier, the data and tuning properties of the ensemble technique. However, we found that the complexity of the chosen algorithm and ensemble approach does not necessarily lead to improvement or better results on the datasets used in this study. The dataset used in the study is linearly separable therefore; there is a need for further study using more complex and imbalance datasets.

The results of the study also show that all ANN models require most time in training their committee of classifiers compared to others ensemble models. For example, MBAB+ANN require 11.73s, AdaM1+ANN require 4.77s and RA-ANN requires 4.61s. On the other hand, RBFN models require least time in training its base classifiers: MBAB+RBFN, AdaM1+RBFN and RA-RBFN requires 0.37, 0.41 and 0.28 seconds respectively.

In the future, we intend to further explore boosting with more emphasis on diversity of the committee of classifiers and experiments on more complex and imbalance cancer datasets. Future work will also involve exploration of algorithm settings to improve classification accuracy.

TABLE 3 PERFORMANCE COMPARISON AMONG THE COMMITTEE OF CLASSIFIERS

Metrics / Models	Accuracy	RMSE	TP Rate	FP Rate	Precision	Recall Curve	ROC Area	PRC Area	Time(s)
AdaM1 + RF	0.97	0.17	0.97	0.04	0.97	0.97	0.99	0.99	0.2
AdaM1 + ANN	0.96	0.2	0.96	0.05	0.96	0.96	0.97	0.97	4.77
AdaM1 + RBFN	0.95	0.2	0.95	0.08	0.96	0.95	0.99	0.98	0.41
RA + RF	0.96	0.18	0.96	0.04	0.97	0.96	0.99	0.99	1.44
RA + ANN	0.88	0.29	0.88	0.22	0.9	0.88	0.98	0.98	4.61
RA+ RBF	0.95	0.2	0.95	0.06	0.95	0.95	0.99	0.99	0.28
MBAB + RF	0.97	0.17	0.97	0.04	0.97	0.97	0.99	0.98	0.14
MBAB+ ANN	0.96	0.19	0.96	0.05	0.96	0.96	0.98	0.98	11.73
MBAB + RBFN	0.96	0.19	0.96	0.03	0.96	0.96	0.99	0.98	0.37

TABLE 4 PERFORMANCE COMPARISON AMONG STANDALONE OF CLASSIFIERS

Metrics / Algorithms	Accuracy	RMSE	TP Rate	FP Rate	Precision	Recall Curve	ROC Area	PRC Area	Time(s)
Random Forest	0.97	0.17	0.97	0.04	0.97	0.97	0.99	0.99	0.13
ANN	0.95	0.2	0.96	0.06	0.97	0.96	0.99	0.98	1.17
RBFN	0.96	0.18	0.96	0.04	0.96	0.96	0.99	0.98	0.05
RIPPER	0.96	0.2	0.96	0.04	0.96	0.96	0.96	0.95	0.06
Naïve Bayes	0.97	0.16	0.97	0.02	0.97	0.97	0.99	0.99	0.03
SMO	0.97	0.17	0.97	0.03	0.97	0.97	0.97	0.96	0.05
SVM	0.96	0.04	0.96	0.03	0.96	0.96	0.97	0.95	0.05
C4.5	0.95	0.22	0.94	0.06	0.94	0.94	0.96	0.94	0.01
K-NN	0.95	0.21	0.95	0.06	0.95	0.95	0.97	0.96	0.02
Logistics Regression	0.97	0.17	0.97	0.05	0.97	0.97	0.99	0.99	0.05

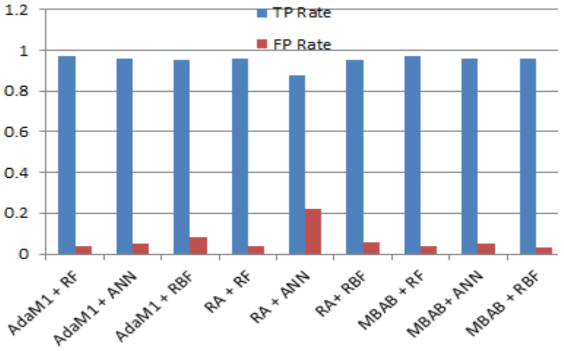


Fig. 5 TP and FP rates: Committee of classifiers

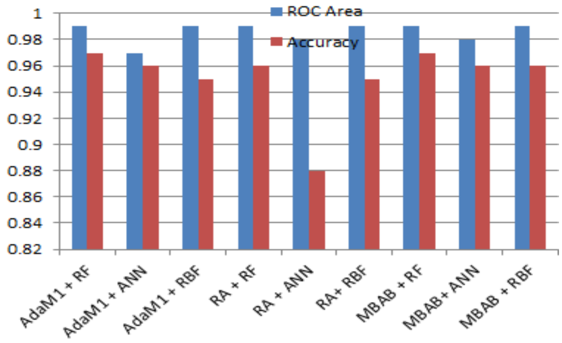


Fig. 6 Accuracy and ROC: Committee of classifiers

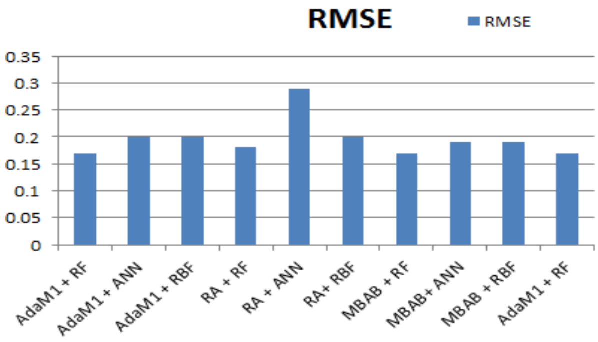


Fig. 7 RMSE: Committee of classifiers

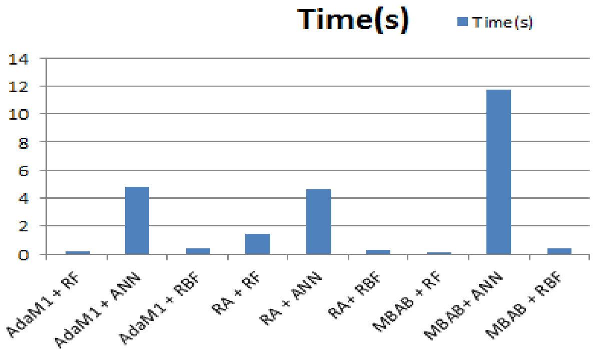


Fig. 8 Running time: Committee of classifiers

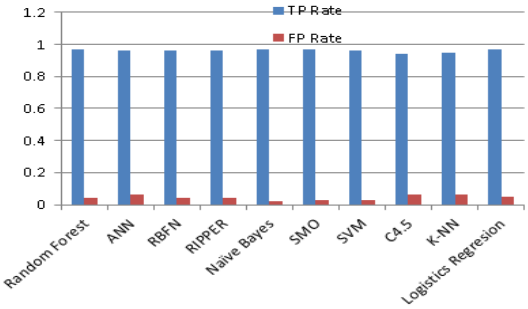


Fig. 9 TP and FP rates: Stand-alone classifiers

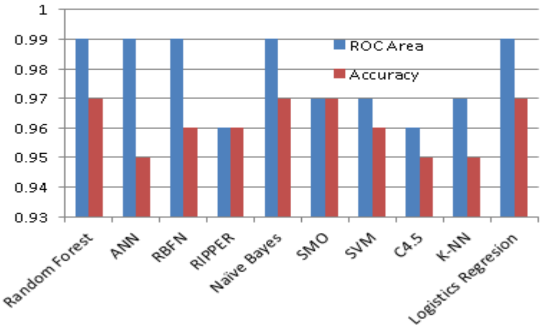


Fig. 10 Accuracy and ROC: Stand-alone classifiers

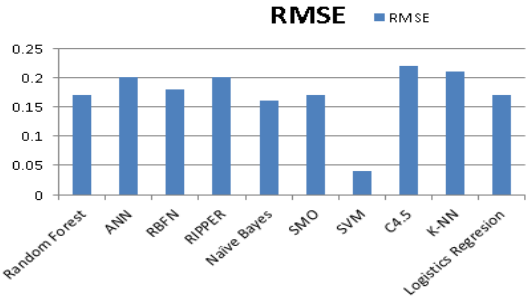


Fig. 11 RMSE: Stand-alone classifiers

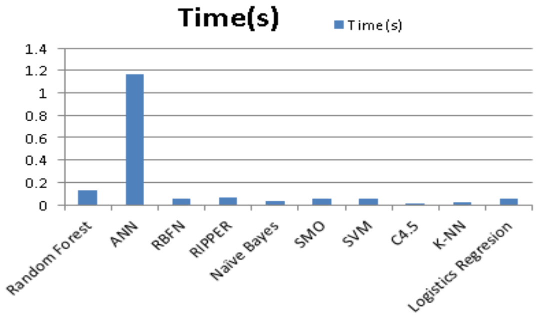


Fig. 12 Time of running: Stand-alone classifiers

VII. References

[1] R. Polikar, "Ensemble Based Systems in Decision Making," *IEEE Circuits and System magazine,* vol. 6, no. 3, pp. 21-45, 2006.

[2] L.G. Valiant, "A theory of the learnable," *Communications of the ACM,* pp. 1134-1142, Nov. 1984.

[3] Y. Freund and R. Schapire, "A brief introduction to boosting," in *Proceedings of the 16th international joint conference on Artificial intelligence*, 1999.

[4] Y. Freund and R. Schapire, "A decision-theoretic generalization of on-line learning and application to boosting," *Journal of computer and system sciences,* vol. 55, pp. 119-139, 1997.

[5] J. Lin and Y. Wang, "Using a Novel AdaBoost Algorithm and Chous Pseudo Amino Acid Composition for Predicting Protein Subcellular," *Proten Pept Lett (US National Library of Medicine National Institutes of Health),* vol. 18, no. 12, pp. 1219-25, December 2011.

[6] P. Viola and M. J. Jones, "Robust Real-Time Face Detection," *International Journal of Computer Vision,* vol. 57, no. 2, pp. 137-154, 2004.

[7] K. A. Abuhasel and A. M. Iliyasu, "A combined AdaBoost and NEWFM Technique for Medical Data classification," *Lecture Notes in Electrical Engineering,* vol. 339, pp. 801-809, 18 February 2015.

[8] A. C. Society, "American Cancer Society. Breast Cancer Facts & Figures 2015-2016," 2015. [Online]. Available: https://www.cancer.org/content/dam/cancer-org/research/cancer-facts-and-statistics/breast-cancer-facts-and-figures/breast-cancer-facts-and-figures-2015-2016.pdf. [Accessed 05 May 2017].

[9] C. R. UK, "Cancer Resarch UK," [Online]. Available: http://www.cancerresearchuk.org/health-professional/cancer-statistics/statistics-by-cancer-type/breast-cancer. [Accessed 05 05 2017].

[10] S. Kwon and S. Lee, "Recent Advances in Microwave Imaging for Breast Cancer Detection," *International Journal of Biomedical Imaging,* 2016.

[11] S. Formenti, A. Arslan and S. Love, "Global Breast Cancer: The Lessons to Bring Home," *International Journal of Breast Cancer,* 2012.

[12] WHO, "Breast cancer: prevention and control," World Health Organization, 2017.

[13] C. Fitzmaurice, D. Dicker and A. P. e. al, "The Global Burden of Cancer 2013," *JAMA Oncology,* vol. 1(5):690, pp. 505-527, 2015.

[14] D. Youlden, S.M.Cramb, C. Yip and P. Baade, "Incidence and mortality of female breast cancer in the Asia-Pacific region," *Cancer Biol Med,* vol. 11, pp. 101-115, 2014.

[15] S.Sapate and S.Talbar, " Mammograms, An Overview of Pectoral Muscle Extraction Algorithms Applied to Digital," *Studies in Computational Intelligence,* 2016.

[16] H. Weedon-Fekjær, P. Romundstad and L. Vatten, "Modern mammography screening and breast cancermortality: population study," *MMJ,* vol. 348:g3701, 2014.

[17] C.-H. Yang, Y.-U. Lin, L.-Y. Chuang and H.-W. Chang, "Evaluation of Breast Cancer Susceptibility Using Improved Genetic Algorithms to Generate Genotype SNP Barcodes," *IEEE/ACM Transactions on Computational Biology and Bioinformatics,* vol. 10, no. 2, pp. 361-371, 2013.

[18] B. McGinley, M. O'Halloran, R. Conceicao, F.Morgan, M.Glavin and E.Jones, "SPIKING NEURAL NETWORKS FOR BREAST CANCER CLASSIFICATION USING RADAR TARGET SIGNATURES," *Progress In Electromagnetics Research C,* vol. 17, pp. 74-94, 2010.

[19] F. Pak, H. Kanan and A. Alikhassi, "Breast cancer detection and classification in digitalmammography based on Non-SubsampledContourlet Transform (NSCT) and Super Resolution," *ComputerMethodsandProgramsinBiomedicine,* vol. 122, pp. 89-107, 2015.

[20] W. Xie, Y. Li and Y. Ma, "Breast mass classification in digital mammography based on extreme learning machine," *Neurocomputing,* vol. 73, no. 3, pp. 930-941, 2015.

[21] I. Mukherjee, C. Rudin and R. Schapire, "The Rate of Convergence of AdaBoost," *Journal of Machine Learning Research,* vol. 14, pp. 2315-2347, 2013.

[22] Y. Freud and R. E. Schapire, Boosting Foundations and Algorithms, MIT Press, 2014.

[23] R. Schapire, "Explaining AdaBoost," in *Empirical Inference*, Springer, 2013, pp. 37-52.

[24] Y. Freund and R. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences,* pp. 119-139, 1997.

[25] R. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Machine Learning,* vol. 37, no. 3, p. 297–336, 1999.

[26] G. Webb, "MultiBoosting: A Technique for Combining Boosting and Wagging," *Machine Learning,* vol. 40, no. 2, 2000.

[27] H. Trevor, T. Robert and F. Jerome, The Elements of Statistical Learning, 2nd ed., Springer Series in Statistics, 2009.

[28] S. Haykin, Neural Networks and Learning Machines: A Comprehensive Foundation, Third ed., Prentice Hall, 2008.

[29] I. Witten, E. Frank and M. Hall, Data Mining Practical Machine Learning Tools and Techniques, 3rd ed., San Francisco, California: Morgan Kaufmann Publishers, 2011.