© Universiti Tun Hussein Onn Malaysia Publisher's Office

IJIE

http://penerbit.uthm.edu.my/ojs/index.php/ijie
ISSN : 2229-838X e-ISSN : 2600-7916

The International Journal of Integrated Engineering

An Efficient Ensemble Method Using K-Fold Cross Validation for the Early Detection of Benign and Malignant Breast Cancer

Mahesh T R[1*], A. C. Kaladevi[2], Balajee J M[3], V Vivek[1], M. Prabu[4], V. Muthukumaran[5]

[1]Department of Computer Science and Engineering,
FET, JAIN (Deemed-to-be University), Bengaluru, INDIA

[2]Department of Computer Science and Engineering,
Sona College of Technology, Salem, INDIA

[3]School of information technology and engineering,
Vellore Institute of Technology, Vellore, Tamilnadu, INDIA

[4]School of Computing Science and Engineering,
VIT Bhopal University, INDIA

[5]Department of mathematics, School of Applied sciences,
REVA University, Bangalore-560064, INDIA

*Corresponding Author

DOI: https://doi.org/10.30880/ijie.2022.14.07.015
Received 26 April 2022; Accepted 1 July 2022; Available online 31 December 2022

Abstract: In comparison to all other malignancies, breast cancer is the most common form of cancer, among women. Breast cancer prediction has been studied by a number of researchers, and is considered as a serious threat to women. Clinicians are finding it difficult to create a treatment approach that will help patients live longer, due to the lack of solid predictive models which predicts the outcome in early stages by analyzing history of patient's data. Rates of this malignancy have been observed to rise, more with industrialization and urbanization, as well as with early detection facilities. It is still considerably more prevalent in very developed countries, but it is rapidly spreading to developing countries as well. The purpose of this work is to offer a report on the disease of breast cancer in which we used available technical breakthroughs to construct breast cancer survivability prediction models. The Machine Learning (ML) techniques, namely Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Decision Tree (DT) Classifier, Random Forests (RF), and Logistic Regression (LR)are used as base Learners and their performance has been compared with the ensemble method, eXtreme Gradient Boosting (XGBoost). For performance comparison, we employed k-fold cross validation method to measure the unbiased estimate of these prediction models. The results indicated that XGBoost outperformed with an accuracy of 97.81% compared to other ML algorithms.

Keywords: Machine learning, breast cancer, accuracy, prediction, recall, F1 score

*Corresponding author: trmahesh.1978@gmail.com

1. Introduction

The count and databases size containing medical data is continually growing. Medical data is continuously kept in various databases as a result of measurements, tests, prescriptions, and so on. This massive amount of data outstrips traditional approach's ability to evaluate and seek for interesting patterns and information contained within it. As a result, demand for innovative strategies and tools for uncovering meaningful information in large data repositories are increasing [1]. In recent years, cancer has been one of the deadliest diseases. Breast cancer is very common frequent cancer among women across the globe. The proper diagnosis of certain crucial information is a remarkable issue in the area of bioinformatics or medical research [2]. In the area of medicine, disease diagnosis is a demanding and challenging task. Many diagnostic institutions, hospitals, research centers, as well as numerous websites have a vast amount of medical diagnostic data. However, it is barely essential to categorize them to automate and speed up disease diagnosis. As per the report of American Cancer Society [3], the disease breast cancer affects more women than any other cancer. Overgrowth of the cells lining the breast ducts is the most common cause of breast tumors, which might be either benign or malignant [4]. The cells of a benign tumor develop improperly and produce a lump. A fibro adenoma is the most frequent kind of benign breast tumor [5]. To confirm the diagnosis, this may need to be surgically removed, and there might not be any need for any additional treatment. If left untreated, cancer cells in a malignant tumor have the ability to migrate beyond the breast. Breast cancer can typically be cured if detected at an early stage [6].

Some of the symptoms of breast cancer are a lump or a clump, changes in breast or nipple size or shape, breast or nipple color changes, nipple might become inverted, discharge of nipples, breast swell or thickening, or even consistent discomfort. Dimpling is a type of skin dimpling that occurs when the skin is, Irritated or flaky skin. Mammography or a portable cancer diagnostic instrument can be used to detect it early during a screening test. The breast cancer tissues change as the disease progresses, which can be connected to the cancer stage. The breast cancer stage (I–IV) indicates how far cancer has extended in that patient.

Stage I: Tumors grow slowly and spread unlikely; this stage can often require surgery to cure.
Stage II: Tumors grow and spread very less, but this stage may likely come back after the treatment.
Stage III: Tumors are rapidly dividing growth of cells but no dead cells are found, this stage grow quickly.
Stage IV: Tumors are actively dividing and tumor having both growth and dead tissues, this stage tumors can grow and spread quickly.

Different stages are discovered using statistical indications like tumor size, distant metastases, lymph node metastasis. Patients must have breast cancer surgery, chemotherapy, radiation, or endocrine therapy to stop cancer from further spreading. Breast cancer occurs in several forms. There are numerous methods for classification as well as prediction of this disease.

2. Related Work

Several studies focusing on breast cancer survival have been published. This research used a variety of ways to solve the problem and was able to obtain excellent classification accuracy [7]. In this work [8] predictive models have been employed based on Decision Tree classifiers to estimate breast cancer survivability of breast cancer and found that patients had an 86.52 percent survival rate. To deal with the imbalanced problem, they used the under-sampling C5 technique and the bagging algorithm, which improved the predicting accuracy of breast cancer disease. This work [9] has proved the utility of ensemble approaches in categorizing microarray data, as well as providing some theoretical explanations for their effectiveness. As a result, they recommend that when categorizing gene expression data for malignant samples, ensemble machine learning (ML) be considered.

For detecting cardiac illnesses, [10] three prominent data mining (DM) algorithms have been presented: CART, ID3 and DT, with the findings demonstrating that CART achieved higher accuracy in very less time. This work [11] conducted a study to figure out the most commonly used data mining algorithms in modern medical diagnosis and to assess their effectiveness on a different set of medical datasets, and Naive Bayes, RBF Network, Simple Logistic, J48, and Decision Tree were chosen as the five algorithms. From high-dimensional profile data [12] and revealed numerous varied and significant rules for credible predictions. Low-rank features were present in the discovered rules, and these characteristics were occasionally required for classifiers in order to reach complete accuracy. This work [13] proposed a new classification approach combining decision trees with bagging and clustering called tree bagging as well as weighted clustering (TBWC). This algorithm has been tested on two different medical datasets: cardiocography1 and cardiocography2, as well as non-medical datasets. In this [14], 202,932 breast cancer patient records were taken and pre-classified them into two groups: those who "survived" (93,273) and those who "did not survive" (109,659). The accuracy of the prediction of survivorship was in the region of 93 percent. In the domain of chemometrics relevant to the pharmaceutical sector [15], it was found that when DT-based ensemble approaches are combined with backward elimination strategy (BES), particularly the boosting tree model, higher classification performance for substances is obtained.

In this study [16], employed ensemble methods and Neural Networks to obtain 92.3% lower accuracy than earlier investigations. This article [17] employed the back propagation approach with 94.2% accuracy. According to this, the

results revealed that the SVM-RBF kernel gives better performance compared to other classifiers, achieving 96.84% accuracy in Wisconsin dataset breast cancer. They employed SVM, KNN, Random Forest, Nave Bayes, and ANN as classification algorithms.

There are numerous methods for classification as well as prediction of this disease [18]. Breast cancers may be classified using a unique ensemble classification algorithm proposed in this study. We used SVM, LR, RF, DT, NB and kNN as base learners for the proposed Blended Ensemble classification model namely XGBoost. In addition, Wisconsin Breast Cancer Dataset from Kaggle and the UCI ML repository are used to assess the performance of the suggested technique [19]. The purpose of the research is to detect and categorize malignant and benign patients and improve prediction accuracy.

3. Methodology

The data set is taken from Kaggle. It consists of 10 independent or input variables are considered as 'X'namely, "Sample code number", "Clump Thickness", "Uniformity of Cell Size", "Uniformity of Cell Shape", "Marginal Adhesion", "Single Epithelial Cell Size", "Bare Nuclei", "Bland Chromatin", "Normal Nucleoli", "Mitoses" and one dependent or output variable is considered as 'Y' consisting of class labels. However, the first feature sample code number is not considered for processing as it does not have any significance. The benign and malignant samples are shown in figure 1.

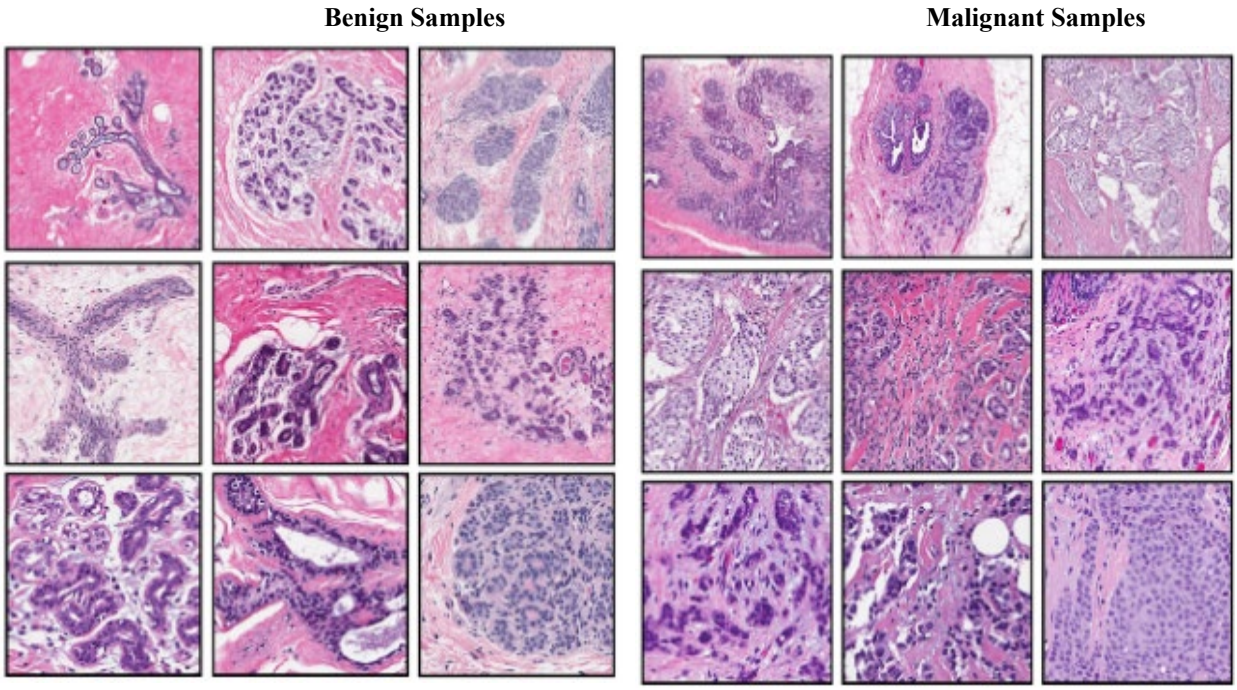

Fig. 1 - Benign and malignant samples

There are about 137 samples in the dataset. Segregation of data is done where in X contains all the input variables and Y contains the class label that is the output variable. After which, data splitting is done where in 70% of the entire dataset is taken training and 30% of the dataset is test data for achieving better accurate results in classification model. The many stages of the procedure are represented in figure 2.

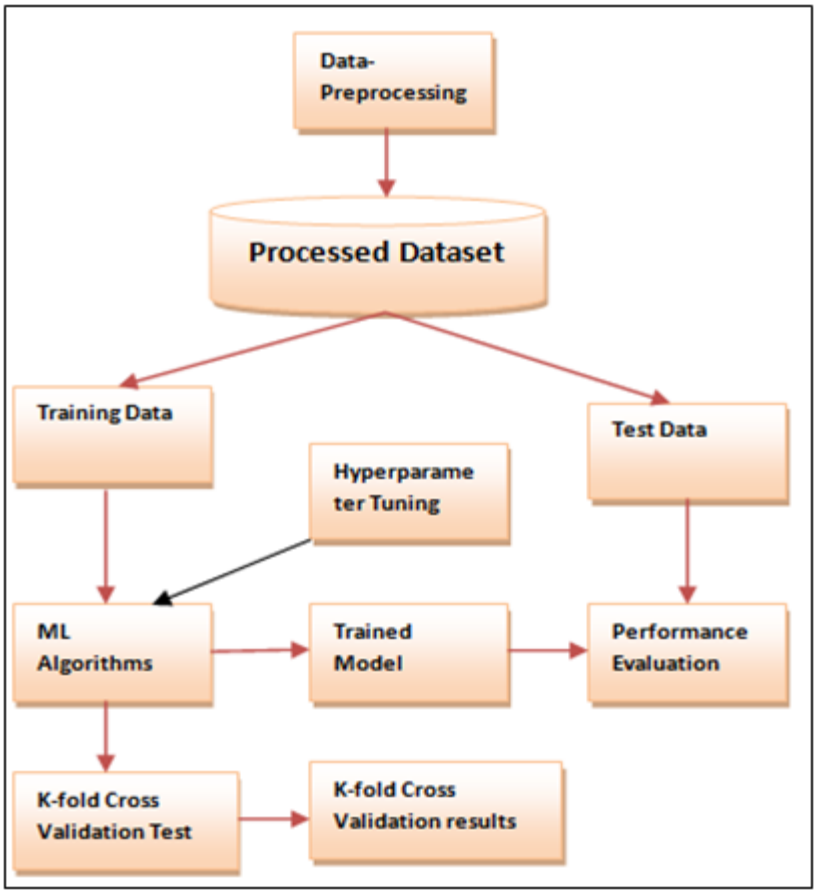

Fig. 2 - The proposed process diagram

The first step is to collect the data that are required for pre-processing so as to improve the data quality by using data cleaning techniques, data transformation and data normalization. Data pre-processing is a DM technique that entails transforming the raw data into a suitable format which can be understood. In reality, the real-world data is inadequate, not consistent, and deficient, and it is almost always riddled with inaccuracies. In the second phase, Data pre-processing is a DM technique for filtering data into a useful format as real-world data is almost always available in a variety of formats. It isn't available in the way that is needed, so it needs to be filtered in a way that one can understand. For data preprocessing, the standardization method is employed to transform the dataset into a usable format. Feature selection, which is called as attribute selection in ML and statistics, is the process of choosing a subgroup of relevant attributes for use in the creation of model.

Understanding Classification necessitates familiarity with training data. The classification process comprises of two stages:

- Training: The phrase "training" refers to a set of scheduled classes. According to the class label attribute, each sample is presumed to associate to a preset class [20].
- Classification: This is used to categorize unknown objects and calculate the model's accuracy. Accuracy rate is proportion of test data categorized properly by the model. Over-fitting will occur if the test set is not separate from the training set [21].

The following six ML algorithms have been implemented to have comparison with respect to performance metrics.

A. Regression Analysis (RA)

RA is a set of procedures of statistics for relationships estimation between a dependent variable and one or more independent variables [22]. It is being used to determine how actually strong a relationship, between variables is as well as to predict how they are going to interact in the future. The model predicts P(Y=1) as a function of X, is one of the simplest ML algorithms that can be used for various classification problems. Here we model the log odds as a linear function of the explanatory variable. For logistic regression we use natural logarithm as shown in equation (1).

$$log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x \quad (1)$$

Where β_0 is the intercept and β_1 is the slope. LR can be extended to situations involving outcome variables with three or more categories.

B. K nearest neighbors (k-NN)

K-nearest neighbors (KNN) is a simple, easy-to-implement supervised machine learning algorithm that may be used to solve both classification and regression problems. Similar objects, according to the KNN algorithm, are close together [23]. To put it another way, items that are related are close to one another. For distance metrics, Euclidean metric can be used and finally, the input x gets assigned to the class with the largest probability as shown in equation (2).

$$d(x, x') = \sqrt{(x_1 - x_1')^2} + \ldots + \sqrt{(x_n - x_n')^2}$$
$$P(y = j|X = x) = \frac{1}{K}\sum_{i \in A}^{n} I\,(y^{(i)} = j) \quad (2)$$

C. Decision Tree Classifier

It is one of the most extensively used and practical approaches used in Machine Learning since it are simple to use and interpret. Classification as well as regression both problems can be solved with decision trees [24]. The name suggests that it use a tree-like flowchart to display the predictions that result from a sequence of splits which are feature-based.

D. Random Forest (RF) Classifier

RF works by using the training data to create several decision trees. In the case of classification, every tree suggests output as a class, also the class with the maximum number of outputs is chosen as the final outcome [25]. We must specify the number of trees we wish to build in this algorithm. RF is such a technique for aggregating or even bagging bootstrap data. This method is being used to reduce an important parameter called variance in the outcomes [26].

E. Support Vector Classifiers

Support Vector Machines (SVM) is supervised learning algorithm that can be used to classify, predict, and discover outliers [27]. SVM works effectively in high-dimensional spaces. It is still effective when the number of dimensions exceeds the number of samples, because the decision function only employs a subset of training data, it is memory efficient. Both dense and sparse sample vectors are accepted as input by scikit-support learns vector machines [28]. To use an SVM to create predictions for sparse data, however, it must be fitted on sparse data, for correct prediction.

F. Ensemble Method

The above-mentioned algorithms are used as base learners. XGBoost, an ensemble method is used with k-fold cross validation. The working of this ensemble method and the evaluation with respect to the performance metrics is discussed in the next section.

Ensemble Method with K-FOLD Cross Validation

The bagging technique controls for high variance in the model. However, boosting plays an important role in order to deal with both bias as well as variance. Boosting is a sequential method that works on the principle of ensemble. It combines a group of ineffective learners to increase prediction accuracy [29]. Outcomes of the model are weighed at any instant t, based on the results of the previous instant t-1. The outcomes that were accurately predicted are given a lower weight, whereas those that were misclassified are given a larger weight [30].

A weak learner is one who is only marginally better than guessing at random. By integrating different models, ensemble learning improves machine learning results as shown in Figure 3.

Boosting is an ensemble modeling strategy that aims to create a strong classifier out of a large number of weak ones. It is accomplished by constructing a model from a sequence of weak models. To begin this process, a model is created using the training data. The second model is then created, which attempts to correct the faults in the first model. This approach is iterated, until the complete training data is properly predicted.

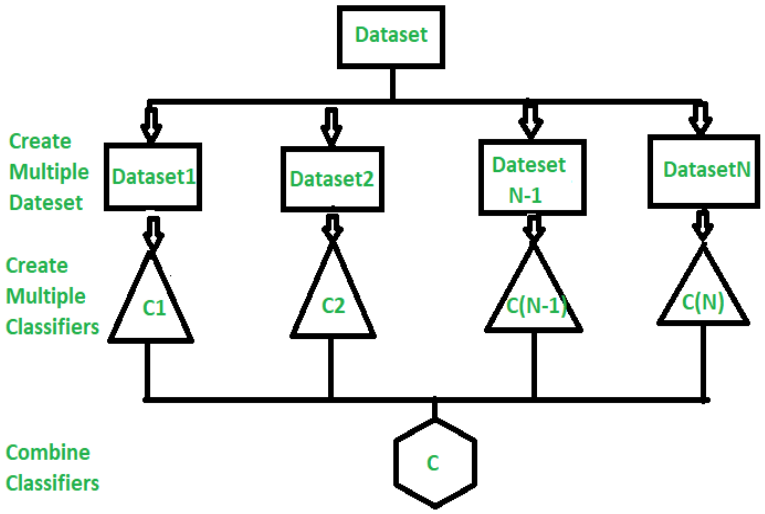

Fig. 3 - Ensemble classifier

There are three different types of gradient boosting machine (GBM) parameters

- Tree-specific parameters which affect every individual tree in the model,
- Boosting parameters which affect the boosting operation
- Miscellaneous parameters which affect overall functioning.

The overall functioning of the XG Boost is shown in table 1.

Table 1 - Overall functioning of XG boost

1: booster [default=gbtree]

- *At each iteration, choose the type of model to run. There are two options:*
  - *gbtree: tree-based models*
  - *gblinear: linear models*

2: Silent [default=0]:

- *Silent mode is set to 1, which means no running messages are printed.*
- *It's best to leave it at 0, as the messages may aid in comprehending the model.*

3: Thread [default to maximum number of threads available if not set]

- *The number of cores in the system should be put here for parallel processing.*
- *If you want to run on all cores, leave the value blank and the algorithm will figure it out.*

Despite the fact that there are two sorts of boosters, we will only consider the tree booster because it consistently outperforms the linear booster, which is why the later is rarely utilized.

We strive to recover the function y=f(x) by approximately estimating f(x) while measuring how good the mapping is using a loss function L(y,x) and then take average over all the dataset points to get the final cost, given the dataset {(xi,yi)}i=1….,n, where x are the features and y is the target, to solve any supervised Machine Learning problem.

XGBoost uses K additive trees to create the ensemble model as shown in equation (3)

$$\hat{y} = \hat{f}(x) = \sum_{i=0}^{K} \hat{f}i\,(x), \hat{f}i(x) \in F \quad (3)$$

Where $F = f(x) = \omega_{q(x)}(q: R^m \rightarrow T, \omega \in R^T)$

The tree structure that transfers an input to the relevant leaf index at which it finishes up is represented by q. The number of leaves on the tree is denoted by T. Each leaf of a regression tree contains a continuous score. The score on the i-th leaf is represented by ω_i.

We minimize the following regularized objective to learn the set of functions employed in the model as shown in equations (4) and (5).

$$l(\emptyset) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega\ (f_k) \quad (4)$$

Where,

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda||\omega|k|^2 \quad (5)$$

Here l is a differentiable convex loss function that measures the difference between the prediction y_i and the target y_i .The second term penalizes the complexity of the model. To minimize over-fitting, the additional regularization term smoothes out the final learned weights. Trees with deeper depth have more leaf nodes, which can lead to overfitting on the training data, as only a few samples end up in each leaf node. As a result, we impose a penalty for the amount of leaf nodes to limit the depth and overfitting. The objective reverts to classic gradient tree boosting when the regularization parameter is set to zero.

To minimize the following objective function, as shown in equation (6) we need to add f_t for the t-th iteration.

$$l^{(t)} = \sum_i l(\hat{y}_i^{t-1} + f_t(x_i), \quad y_i) + \sum_t \Omega(f_k) \quad (6)$$

We can approximate our objective function second-order via Taylor series expansion. A Taylor series is a function's series expansion around a point. The expansion of a real function f(x) around a point x=a is called a one-dimensional Taylor series, and it is given by the equation (7).

$$f(x) = f(a) + f'(a)(x-a) + \frac{f''}{2!}(x-a)^2 + \cdots + \frac{f^n}{n!}(x-a)^n + \cdots \quad (7)$$

We can approximate our function as shown in equation (8) by using second order approximation while disregarding higher order terms.

$$l^{(t)} = \sum_i^n [\, l(\hat{y}_i^{t-1}, y_i) + g_i f_t(x_i) + \frac{1}{2} h_i f_i^2(x_i)] + \Omega\ (f_k) \quad (8)$$

To make the objective function simpler, we can delete the constant terms as depicted in equation (9).

$$l^{(t)} = \sum_i^n [\, g_i f_t(x_i) + \frac{1}{2} h_i f_i^2(x_i)] + \Omega\ (f_k) \quad (9)$$

Let Ij={ i|q(x_i=j)} represents the instance set of leaf j, that is, the collection of all the input data points that ended up in the j-th leaf node. So, for a particular tree, if one of our input data points ends up at the j-th leaf node after all of the decisions, we will include it in our set Ij as shown below in equation (10).

$$\begin{aligned} l^{(t)} &= \sum_i^n [\, g_i f_t(x_i) + \frac{1}{2} h_i f_i^2(x_i)] + \gamma T + \frac{1}{2}\lambda||\omega||^2 \\ &= \sum_{i=1}^n [\, (\sum_{i \in I_j} g_i)\omega_j + \frac{1}{2}(\sum_{i \in I_j} h_i + \lambda)\omega_j^2] + \gamma T \end{aligned} \quad (10)$$

We can determine the optimal weight ω_j^* of leaf j for a fixed tree structure q(x) by differentiating the previous equation with regard to w and equating to 0 as shown in (11).

$$\omega_j^* = \frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda} \quad (11)$$

For the time being, we will assume that we have a tree structure q for which we have determined the best weights at each leaf node. If you look at the above ω_j^* equation, you will note that the leaf nodes are missing, i.e. I_j hasn't been calculated yet. Based on this, we have is the ability to determine the best leaf node weights for every tree topology. The following step is used to locate the best tree structure that minimizes the loss, and then we would be done with our tree search.

We now replace ω_j^* in the preceding equation as shown below in equation (12) with the appropriate optimal value for the provided tree structure q.

$$l^{(t)}(q) = -\frac{1}{2}\sum_{j=1}^T \frac{\left(\sum_{i \in I_j} g_i\right)^2}{\sum_{i \in I_j} h_i + \lambda} + \gamma T \quad (12)$$

The quality of a tree structure q can be measured using the above equation. The score is similar to the impurity score in terms of evaluating trees, but it is calculated for a broader variety of objective functions. It is normally difficult to list all of the possible tree architectures q. Instead, a greedy algorithm is used, which starts with a single leaf and iteratively adds branches to the tree. Assume that I_L and I_R are the left and right node instance sets after the split. If we assume $I= I_L \cup I_R$, the loss decrease after the split is shown in equation (13).

$$l_{split} = \frac{1}{2}\left[\frac{\left(\sum_{i\in I_L} g_i\right)^2}{\sum_{i\in I_L} h_i + \lambda} + \frac{\left(\sum_{i\in I_R} g_i\right)^2}{\sum_{i\in I_R} h_i + \lambda} - \frac{(\sum_{i\in I} g_i)^2}{\sum_{i\in I} h_i + \lambda}\right] - \gamma(t+1-t)$$
$$= \frac{1}{2}\left[\frac{\left(\sum_{i\in I_L} g_i\right)^2}{\sum_{i\in I_L} h_i + \lambda} + \frac{\left(\sum_{i\in I_R} g_i\right)^2}{\sum_{i\in I_R} h_i + \lambda} - \frac{(\sum_{i\in I} g_i)^2}{\sum_{i\in I} h_i + \lambda}\right] - \gamma \quad (13)$$

i.e., Total loss after splitting minus total loss prior to splitting
Similar to the gini index or entropy, this score can be used to evaluate split candidates.
XGBoost uses well-known metrics namely Mean Squared Error (MSE) as shown in (14) and Mean Absolute Error (MAE) as shown in (15) to assess a regression model's performance.

$$MSE = \frac{\sum_{i=1}^{n}(y_i - y_i^p)^2}{n} \quad (14)$$

$$MAE = \frac{\sum_{i-1}^{n}|y_i - y_i^p}{n} \quad (15)$$

As discussed earlier, XGBoost calculates the Gain of the root node using Similarity Score rather than Gini or Entropy as shown below in Equations (16),(17) and (18).

$$SimilarityScore = \frac{(\sum \text{Residual}_i)^2}{\sum|\text{Previous Probability}_i * (1 - \text{Previous Probability}_i)| + \lambda} \quad (16)$$

$$Gain = \text{Left}_{\text{Similarity}} + \text{Right}_{\text{Similarity}} - \text{Root}_{\text{Similarity}} \quad (17)$$

$$\text{Output Value} = \frac{(\sum \text{Residual}_i)}{\sum|\text{Previous Probability}_i * (1 - \text{Previous Probability}_i)| + \lambda} \quad (18)$$

Cross-validation is a technique used in applied ML to estimate a ML model's skill on unknown data. XGBoost allows the user to perform cross-validation at each iteration of the boosting process, making it simple to obtain the correct number of boosting iterations in a single run. The general procedure of k-fold Cross Validation is shown in table 2.

Table 2. Procedure for K- Fold Cross Validation

Step_1: Randomly shuffle the given dataset.
Step_2: Organize the data into k number of groups.
Step_3: For each distinct group, write:
- *a. Take the group as test data set*
- *b. Take the remaining groups as a training data set*
- *c. Fit a model on the training set and evaluate it on the test set*
- *d. Keep the evaluation score but discard the model.*

Step_4: Using the model evaluation scores sample , the model's ability is summarized.

Result and Discussions

The dataset used is Breast Disease UCI (UC Irvine) that was taken from Kaggle and UCI ML (UC Irvine Machine Learning) repositories. It consists of 10 input variables and the heatmap is depicted in figure 4. Color-coded systems are used to create heat maps, which are graphical representations of data. Heat Maps are primarily used to better represent the amount of locations/events within a dataset and to guide users to the most important sections on data visualizations. A heatmap is a 2-dimensional graphical representation of data which uses colors to depict the individual values in a matrix is generated using python tool.

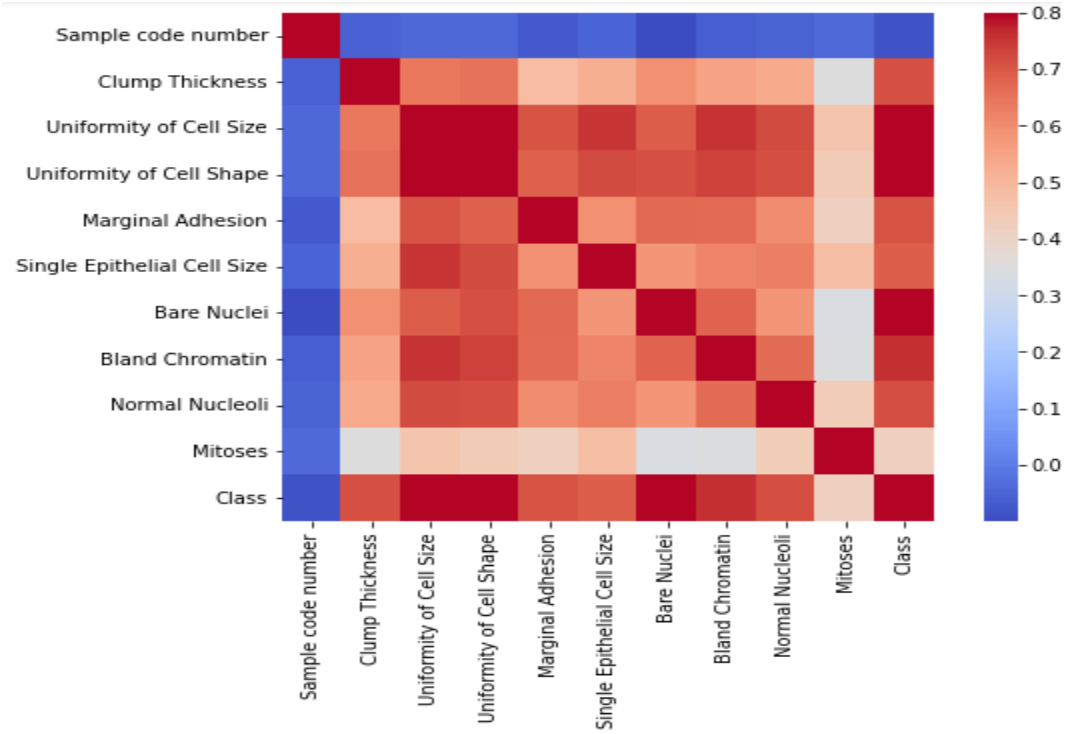

Fig. 4 - Heatmap

Using confusion matrix, the accuracy had been determined and the Confusion matrices were created using a model's predictions on a data set. Also, one can grasp the strengths and shortcomings of the model by looking at this confusion matrix, and the comparison is done with alternative models to see, which one is determined to be the best for prediction. The confusion matrix when XGBoost algorithm is applied on the given dataset is depicted in figure 5. As stated earlier, there are 137 samples in the dataset.

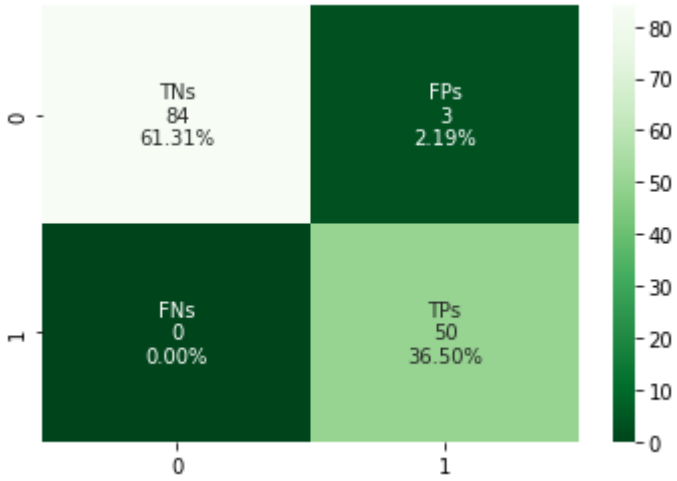

Fig. 5 - Confusion matrix

"Yes" as well as "no" are the two potential predicted classes. If we were recommending the presence of breast cancer, "yes" would definitely indicate that they have the condition, while "no" would indicate that definitely they do not have the condition.

True positives (TPs): These are examples where the prediction was yes, indicating that they have the breast cancer disease and also, it was predicted correctly. As shown in the figure, 50 out of 137 samples are predicted as TPs resulting in 36.50%.

True negatives (TNs): The model predicted that they wouldn't have the disease, and actually it was found that they don't. The TNs rate is 61.31% where it predicts 84 out of 137 samples as TNs.

False positives (FPs): Model projected that they would have the disease, but actually it was found that they don't have the disease. This is also known as "Type I error".

False negatives (FNs): These are those who test negative but actually have been found to have the disease. This is also known as "Type II mistake".

FNs rate is 0% and FPs rate is 2%. This clearly shows XGBoost provides significantly very less error rate. The XGBoost classifier had made a total of 137 predictions mapping to the 137 records of the patients. Out of those 137 cases, the classifier predicted "yes" for 134 times, and "no" for 3 times.

Accuracy: If the dataset is not balanced, accuracy may not be a good measure. The number of correctly classified instances divided by the total number of data instances is referred as accuracy. The Accuracy is computed using the Equation (19). So looking at the confusion matrix depicted in figure 5, the Accuracy of the XGBoost classifier is **97.81%.**

$$Accuracy = \frac{TNs + TPs}{Tns + TPs + FPs + FNs} \tag{19}$$

Precision: Precision is one of the performance metrics that is going to measure how many correct positive forecasts have been made. So, precision estimates the accuracy of the minority class, then, the ratio of accurately predicted positive instances divided by the total number of positive examples predicted, is used to compute it. The precision is computed using the Equation (20). So looking at the confusion matrix depicted in figure 5, the precision of the XGBoost classifier is found to be**94.3%.**

$$Precision = \frac{TPs}{TPs + FPs} \tag{20}$$

A good classifier should have a precision of 100% (high), only when both numerator as well as denominator are identical, i.e. TP = TP +FP, can precision become 100%. However, XGBoost classifier has provided a very good precision rate.

Recall: Recall is a metric that measures how many correct positive predictions were produced out of all possible positive predictions. Unlike precision, which only considers the right positive predictions out of all positive predictions, recall considers the positive predictions that were missed. In this approach, recall provides some indication of the positive class' coverage. The recall is computed using the Equation (21). So, by looking at the confusion matrix depicted in figure 5, the precision of the XGBoost classifier isfound to be**100%.**

$$recall = \frac{TPs}{TPs + FNs} \tag{21}$$

F1 Score: We want both accuracy and recall to be of the value one, in a good classifier, which also means FP and FN should be zero. As a result, we require a statistic that considers both precision as well as recall. The F1-score is a measure that takes precision and recall into account, and is defined as follows, as shown in Equation (22).

$$F1\ Score = 2 * \frac{precision * recall}{precision + recall} \tag{22}$$

XGBoost classifier provides the F1 score of **97.06%.** Only when precision and recall are both 100 percent, it results in F1 Score to also become 100 percent. That means when both precision and recall both are high, then only, F1 score will be high. The F1 score is a better measure than accuracy since, it is the harmonic mean of precision and recall. XGBoost classifier's performance against all the mentioned performance metrics has been depicted in Figure 6

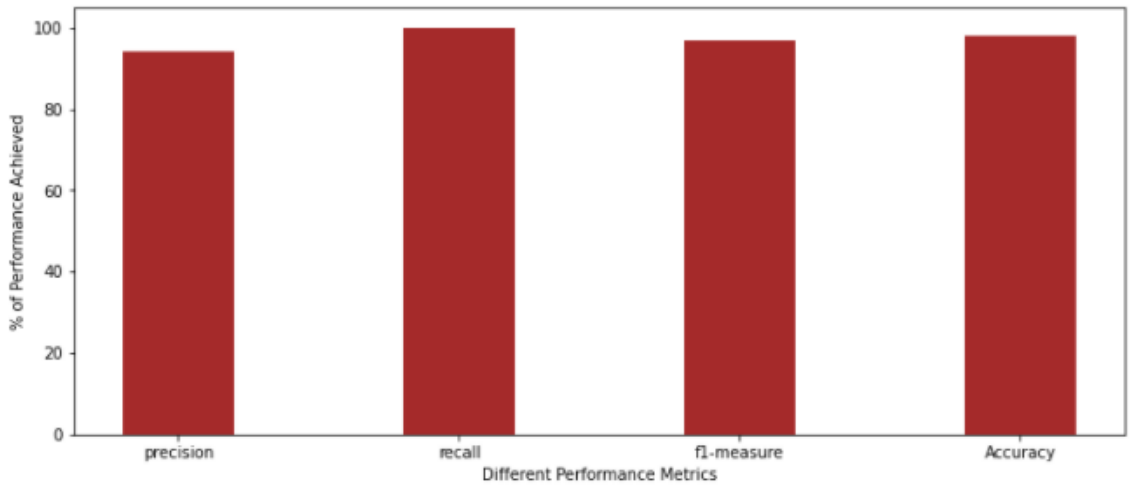

Fig. 6 - Different performance metrics

The accuracy of LR, KNN, RF, DT, SVM and XGBoost classifiers are 96.32%, 94.11%, 95.22%, 91.19% , 93.46% and 97.81 respectively as shown in table 2. The table also shows how the six ML algorithms have been evaluated with respect to the different performance metrics.

Table 3 - Performance of various ML techniques in classification of breast cancer

ML Techniques	Accuracy	Recall	Precision	F1-Score
LR	96.32	89.14	95.17	94.27
KNN	94.11	92.36	93.42	92.08
RF	95.22	93.56	92.38	94.36
DT	91.19	88.72	90.15	92.21
SVM	93.46	94.48	89.17	92.11
XGBoost	97.81	100	94.3	97.06

Out of six ML algorithms when evaluated with respect to several performance metrics, like precision, recall and F1 score, it was observed that XGBoost has been evaluated with a very good accuracy compared to other classifiers. The Performance Evaluation of Different ML techniques in Breast Cancer Classification are depicted in the figure 7.

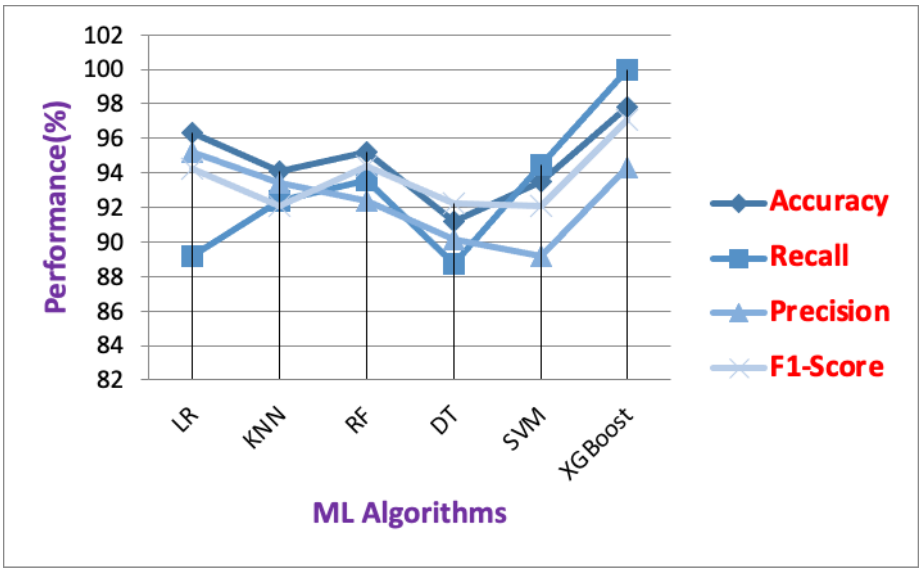

Fig. 7 - Performance evaluation of different ML techniques in breast cancer classification

conclusion

Breast cancer, if detected early, can save many women lives. These programs assist patients and clinicians in gathering as much information as possible in the real world. The objective is to determine the best suitable algorithm for forecasting the occurrence of the disease of breast cancer accurately. The main objective of this paper, is to highlight all of the previous and existing studies of ML algorithms that have been used to predict breast cancer. To indicate survivability in this study, we employed a binary variable in the raw dataset, where benign is denoted by a value of "0" and malignant is denoted by a value of "1". We employed a 10-fold cross validation strategy to test the unbiased prediction accuracy of these algorithms. For each of the six prediction models, we have repeated this procedure, as a result of which, we were able to compare these six models using less skewed prediction performance measures. With a classification accuracy of 97.81 percent, the XGBoost was observed to have performed the best.

References

[1] Y. Lu, J.-Y. Li, Y.-T. Su, and A.-A. Liu, ‘‘A review of breast cancer detection in medical images,’’ in Proc. IEEE Vis. Commun. Image Process. (VCIP), Dec. 2018, pp. 1–4

[2] Y.-S. Sun, Z. Zhao, Z.-N. Yang, F. Xu, H.-J. Lu, Z.-Y. Zhu, W. Shi, J. Jiang, P.-P. Yao, and H.-P. Zhu, ‘‘Risk factors and preventions of breast cancer,’’ Int. J. Biol. Sci., vol. 13, no. 11, p. 1387, 2017.

[3] Y. Khourdifi and M. Bahaj, "Feature Selection with Fast Correlation-Based Filter for Breast Cancer Prediction and Classification Using Machine Learning Algorithms," 2018 International Symposium on Advanced Electrical and Communication Technologies (ISAECT), Rabat, Morocco, 2018, pp. 1-6.

[4] H. Sharma and S. Kumar, ‘‘A survey on decision tree algorithms of classification in data mining,’’ Int. J. Sci. Res., vol. 5, no. 4, pp. 2094–2097, 2016.

[5] Dubey, A., 2016. Applications of machine learning: cutting edge technology in HIV diagnosis, treatment and further research. Computational Molecular Biology, 6(3)

[6] Keles, M. Kaya, "Breast Cancer Prediction and Detection Using Data Mining Classification Algorithms: A Comparative Study." TehnickiVjesnik - Technical Gazette, vol. 26, no. 1, 2019, p. 149.

[7] K. K. Jha, A. K. Jha, K. Rathore and T. R. Mahesh, "Forecasting of Heart Diseases in Early Stages Using Machine Learning Approaches," 2021 International Conference on Forensics, Analytics, Big Data, Security (FABS), 2021, pp. 1-5

[8] Liu Y-Q, Wang C and Zhang L. Decision tree based predictive models for breast cancer survivability on imbalanced data. In: 3rd international conference on bioinformatics and biomedical engineering, 11-13 June 2009, Beijing, China, 2009.

[9] Tan AC and Gilbert D. Ensemble machine learning on gene expression data for cancer classification. Appl Bioinformatics 2003; 2: S75–S83

[10] Chaurasia V and Pal S. Data mining techniques: to predict and resolve breast cancer survivability. Int J ComputSci Mobile Comput 2014; 3: 10–22.

[11] Pal S. and Chaurasia V, A novel approach for breast cancer detection using data mining techniques. Int J Innovative Res Comput CommunEng 2014; 2: 2456–2465.

[12] Li J, Liu H and Ng S-K, Discovery of significant rules for classifying cancer diagnosis data. Bioinformatics 2003;19: ii93–ii102.doi: 10.1093/bioinformatics/btg1066

[13] Vongsuchoto N, Kaewchinporn C and Srisawat A. A combination of decision tree learning and clustering for data classification. In: 2011 eighth international joint conference on computer science and software engineering (JCSSE), MAY 11-13, 2011, Faculty of ICT, Mahidol University, Nakhon Pathom,Thailand.

[14] Delen D, Walker G and Kadam A. Predicting breast cancer survivability: a comparison of three data mining methods. ArtifIntell Med 2005; 34: 113–127.

[15] Galatzer-Levy, I.R., Karstoft, K., Statnikov, A.I., & Shalev, A.Y. (2014). Quantitative forecasting of PTSD from early trauma responses: a Machine Learning application. Journal of psychiatric research, 59, 68-76

[16] R. K. Kavitha1, D. D. Rangasamy, “Breast Cancer Survivability Using Adaptive Voting Ensemble Machine Learning Algorithm Adaboost and CART Algorithm” Volume 3, Special Issue 1, February 2014

[17] Keles, M. Kaya, "Breast Cancer Prediction and Detection Using Data Mining Classification Algorithms: A Comparative Study." TehnickiVjesnik - Technical Gazette, vol. 26, no. 1, 2019, p. 149.

[18] B. Akbugday, "Classification of Breast Cancer Data Using Machine Learning Algorithms," 2019 Medical Technologies Congress (TIPTEKNO), Izmir, Turkey, 2019, pp. 1-4.

[19] Deshmukh, S. D Medhekar, D. S. &Bote, M. P. (2013). “Heart disease prediction system using naive Bayes”. Int. J. Enhanced Res. Sci. Technol. Eng, 2(3).

[20] Ansari, U., Soni, J., Sharma, D., &Soni, S. (2011).” Intelligent and effective heart disease prediction system using weighted associative classifiers”. International Journal on Computer Science and Engineering, 3(6), 2385-2392.

[21] Chitra, R., & Seenivasagam, V. (2013). “Review of heart disease prediction system using data mining and

hybrid intelligent techniques". ICTACT journal on soft computing, 3(04), 605-09.

[22] P. Shrestha, A. Singh, R. Garg, I. Sarraf, T. R. Mahesh and G. Sindhu Madhuri, "Early-Stage Detection of Scoliosis Using Machine Learning Algorithms," 2021 International Conference on Forensics, Analytics, Big Data, Security (FABS), 2021, pp. 1-4.

[23] Anze Vavpetic and Nada Lavrac . "Relational and semantic data mining". In Proceedings of the Thirteenth International Conference on Logic Programming and Nonmonotonic Reasoning, pages 20–31, Lexington, KY, USA, 2015.

[24] M. R. Sarveshvar, A. Gogoi, A. K. Chaubey, S. Rohit and T. R. Mahesh, "Performance of different Machine Learning Techniques for the Prediction of Heart Diseases," 2021 International Conference on Forensics, Analytics, Big Data, Security (FABS), 2021, pp. 1-4.

[25] Delen, D.; Walker, G.; Kadam, A. Predicting breast cancer survivability: A comparison of three data mining methods. Artif. Intell. Med. 2005, 34, 113–127

[26] K. R. Lakshmi and S.Prem Kumar,"Utilization of Data Mining Techniques for Prediction of Diabetes Disease Survivability", International Journal of Scientific &Engineering Research, Volume 4, Issue 6, June-2013

[27] H. K. Shashikala, T. R. Mahesh, V. Vivek, M. G. Sindhu, C. Saravanan and T. Z. Baig, "Early Detection of Spondylosis using Point-Based Image Processing Techniques," 2021 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT), 2021, pp. 655-659.

[28] Dharahas Reddy T. and T. R. Mahesh, "A Pragmatic Approach for Detecting Brain Tumors Using Machine Learning Algorithms", Bioscience Biotechnology Research Communications Special Issue Vol 14 No (11) (2021).

[29] Kuldeep Sharma, Mahesh T R, Big Data Technology for Developing Learning Resources, Journal of Physics: Conference Series IOP Publishing Ltd, May 2021.

[30] Ch. Shravya, K. Pravalika, Shaik Subhani, "Prediction of Breast Cancer Using Supervised Machine Learning Techniques", International Journal of Innovative Technology and Exploring Engineering (IJITEE), Volume-8 Issue-6, April 2019.