

Research Article

Novel Based Ensemble Machine Learning Classifiers for Detecting Breast Cancer

Taarun Srinivas ¹, **Aditya Krishna Karigiri Madhusudhan**,¹ **Joshuva Arockia Dhanraj** ¹,
Rajasekaran Chandra Sekaran,¹ **Neda Mostafaeipour** ², **Negar Mostafaeipour** ³,
and Ali Mostafaeipour ⁴

¹Centre for Automation and Robotics (ANRO), Department of Mechatronics Engineering, Hindustan Institute of Technology and Science, Padur, Chennai 603103, India

²Tehran University of Medical Sciences School of Medicine, Tehran, Iran

³Medical School, Rafsanjan University of Medical Science, Rafsanjan, Kerman, Iran

⁴Industrial Engineering Department, Yazd University, Yazd, Iran

Correspondence should be addressed to Neda Mostafaeipour; mostafaei.neda@yahoo.com and Ali Mostafaeipour; mostafaei@yazd.ac.ir

Received 16 February 2022; Revised 8 April 2022; Accepted 11 April 2022; Published 10 May 2022

Academic Editor: Ramin Ranjbarzadeh

Copyright © 2022 Taarun Srinivas et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Nowadays, for many industries, innovation revolves around two technological improvements, Artificial Intelligence (AI) and machine learning (ML). ML, a subset of AI, is the science of designing and applying algorithms that can learn and work on any activity from past experiences. Of all the innovations in the field of ML models, the most significant ones have turned out to be in medicine and healthcare, since it has assisted doctors in the treatment of different types of diseases. Among them, early detection of breast cancer using ML algorithms has piqued the interest of researchers in this area. Hence, in this work, 20 ML classifiers are discussed and implemented in Wisconsin's Breast Cancer dataset to classify breast cancer as malignant or benign. Out of 20, 9 algorithms are coded using Python in Colab notebooks and the remaining are executed using the Waikato Environment for Knowledge Analysis (WEKA) software. Among all, the stochastic gradient descent algorithm was found to yield the highest accuracy of 98%. The algorithms that gave the best results have been considered in the development of a novel ensemble model and the same was implemented in both WEKA and Python. The performance of the ensemble model in both platforms is compared based on metrics like accuracy, precision, recall, and sensitivity and investigated in detail. From this experimental comparative study, it was found that the ensemble model developed using Python has yielded an accuracy of 98.5% and that developed in the WEKA has yielded 97% accuracy.

1. Introduction

Any environment can be made smart and intelligent with the help of technologies like AI and ML [1]. These technologies not only reduce the manpower and labor time but also improve the quality of life by automating society and creating more job opportunities [2, 3]. A system that can apply some logic, evaluate the different options available, and decide on its own can be considered to be an application of Artificial Intelligence. To provide this intelligence, the system has to be trained; i.e., the system has to learn. This is

done through machine learning, a branch of AI, where the system makes a decision based on the data. In ML, the performance of a task improves based on previous data or experience. Thus, it can be understood that performance, task, and experience are the three main characteristics of ML. Figure 1 represents the basic flow of machine learning pictorially.

The applications of ML can be found in many areas such as object recognition, spam detection, speech recognition, and healthcare industries for survival predictions, tumor detection, and cancer detection. The use of AI and ML

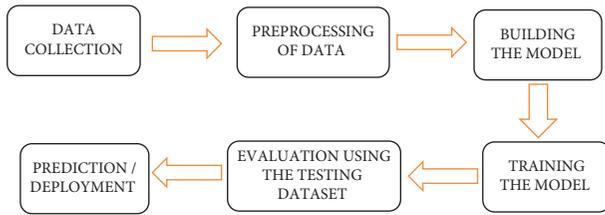


FIGURE 1: The basic flow of machine learning.

models assists doctors in the early detection and curing of cancer, tumors, and other deadly diseases. One such application in medicine is the detection of breast cancer using ML models. Breast cancer is the growth of unwanted cells in the breast and early detection of this cancer can lead to a better cure [4]. Many classification algorithms in machine learning can be used to detect breast cancer with high accuracy, but selecting a particular algorithm for breast cancer detection is a challenging task. There are many works done by researchers working in these areas and these kinds of works give rise to the establishment of smart hospitals and constructive exploitation of these technologies leads to the improvement in quality of life [5].

Chandrasekaran et al. [6] developed a fully integrated analog-based machine learning classifier that uses Artificial Neural Network (ANN) algorithms for breast cancer prediction. The developed ANN model mainly focused on nine attributes of breast cancer, such as (1) bland chromatin, (2) mitosis, (3) uniformity of cell shape, (4) marginal adhesion, (5) single epithelial cell size, (6) uniformity of cell size, (7) clump thickness, (8) normal nucleoli, and (9) bare nuclei. The authors achieved a mean accuracy of 96.9% with optimal energy consumption. Later, a Deep Neural Network with Support Value (DNNS) was developed by Vaka et al. [7] for predicting breast cancer. The model was developed in three stages. In the first stage, preprocessing was carried out by removing noise from the images in the dataset. In the second stage, the geometrical and textual features were detected from the preprocessed image. In the third stage, each image was segmented using the histosigmoid based fuzzy classifier. The authors achieved an accuracy of 97.21%. Mojriani et al. [8] developed a multilayer fuzzy expert system by integrating an Extreme Learning Machine (ELM) classification model with Radial Basis Function (RBF). The authors compared the proposed methodology with the linear Support Vector Machine (SVM) model. Accuracy, Precision, True Positive (TP rate), False Positive (FP rate), True Negative (TN rate), and False Negative (FN rate) were considered as the performance metrics to evaluate the proposed ML model. The authors achieved an accuracy of 98% and the proposed methodology was effective in detecting breast cancer when compared with the linear SVM model. Chaurasia and Pal [9] predicted and compared six machine learning algorithms. Classification and Regression Trees (CART), Linear Regression, Multilayer Perceptron (MLP), Naive Bayes (NB), K Nearest Neighbors (KNN), and SVM were the basic learners considered for prediction. Along with basic learners, a few ensemble algorithms like AdaBoost M1, Random Forest (RF), and Support Vector Clustering (SVC) were integrated

to yield high accuracies. The authors divided the training and test set in the ratio of 80 : 20, respectively. Accuracy was the metric considered by the authors and they achieved a good accuracy of 95.17%. Similar to [9], a comparative analysis using data visualization and ML applications for breast cancer detection and diagnosis was carried out by [10] and achieved an accuracy of 98% using the logistic regression model.

Having studied the recent works of these authors, it is understood that individual algorithms like either deep neural networks or ensemble algorithms have been used at a time. To further improve the accuracy of existing methodologies, a novel combination of ensemble and optimization-based approaches is required to detect breast cancer with high accuracy. In this paper, 20 different ML classification algorithms have been selected for predicting breast cancer as benign or malignant using Wisconsin's Breast Cancer dataset. Out of these 20, 9 were implemented in Google Colab notebooks using Python programming language and the remaining 11 were executed in WEKA data mining software. Along with the 20 algorithms, we propose a novel machine learning ensemble-based algorithm. The algorithms for ensemble were chosen based on the performance of the 20 algorithms. The ensemble model developed was made to run on both platforms and the results obtained were compared and investigated in detail. Thus, the work in this paper can assist doctors in accurately detecting and diagnosing breast cancer in the early stages. This is an attempt to improve and automate society, establish the concept of smart hospitals, and contribute to the development of these technologies.

The rest of this paper is organized as follows. Section 2 introduces the dataset and discusses the EDA performed to understand the importance and contribution of individual features. Section 3 describes the working of the selected 20 algorithms. Section 4 explains the evaluation parameters considered to investigate the performance of the algorithm and Section 5 discusses the results obtained and explains in detail the comparative study carried out in both platforms before concluding in Section 6.

2. Wisconsin's Breast Cancer Dataset

The Breast Cancer dataset from the University of Wisconsin Hospital was considered for carrying out the performance analysis of 20 different classification algorithms. This is a small dataset consisting of data from 569 patients with 31 different input features and a single output feature. The 31 different features are independent of each other and the output feature called "diagnosis" is dependent on the input features. The output feature "diagnosis" has two labels: malignant and benign.

2.1. Exploratory Data Analysis (EDA). The Breast Cancer dataset consists of data from 569 patients, out of which 359 patients were diagnosed as benign and 212 patients were diagnosed as malignant. The blue region from Figure 2 represents the percentage share of patients who were

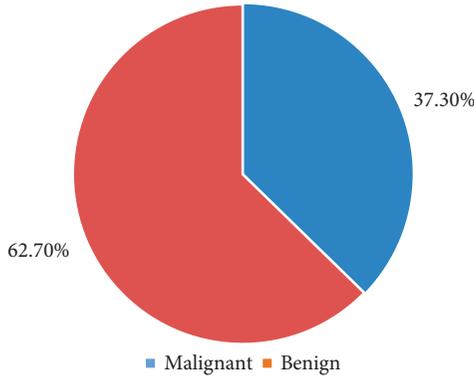


FIGURE 2: Benign and malignant cases.

diagnosed as malignant (37.3%) and the red region represents the percentage share of patients who were diagnosed as benign (62.7%).

As the percentage shares are not equal, there is a class imbalance and the predictions may not be generalized. To overcome this class imbalance problem and to have generalized results, the number of data samples must be equal for both labels. Inferring from Figure 3, if the mean radius is higher, then the patient is more likely to be diagnosed as malignant and if it is lower, then the patient is diagnosed as benign. So, the mean radius is directly proportional to the chances of a patient having breast cancer or not.

Similarly, from Figure 4 it is understood that people who were diagnosed as malignant have mean compactness in the range of 0.05 to 0.27 and those who have been diagnosed as benign have mean compactness in the range of 0 to 0.18. As the mean compactness increases, the chances of a person likely having cancer cells are more. When the cancer is detected as malignant, the mean area of spread of cancer is between 500 and 1800 mm and when the same has been detected as benign, the mean area of spread of cancer is between 0 and 700 mm and their respective sum is depicted in Figure 5. Early cancer detection helps to stop the spread of cancer when compared to malignant cases. Curing of malignant cases is very hard as the area of spread will be high.

3. Brief Description of the Classification Algorithms

The chart given in Figure 6 explains the types of ML. The algorithms listed in the chart below are the most common and popularly used classifiers. Instance-based algorithms such as KNN and statistical-based algorithms such as Naive Bayes, Bayes Net, Neural Network, Decision Trees, and more are experimentally verified in the next section.

3.1. Logistic Regression. It is a widely used machine learning approach for categorization. The logistic regression algorithm draws a decision boundary between the different input features. The model then maps the inputs to the outputs by computing the cost function. The sigmoid function, also called a logistic function, is used for predicting the target

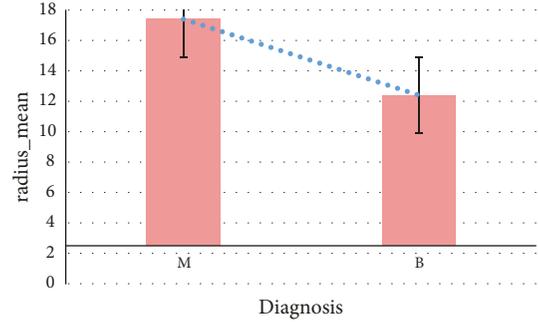


FIGURE 3: Diagnosis vs. radius mean.

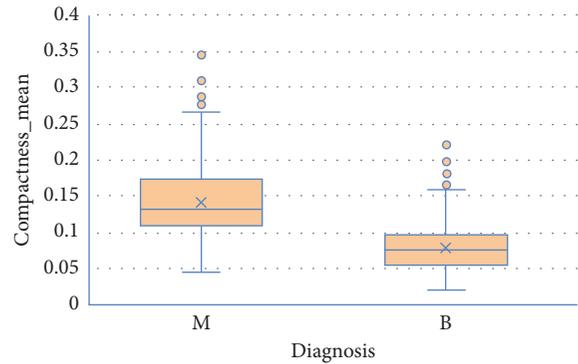


FIGURE 4: Diagnosis vs. compactness mean.

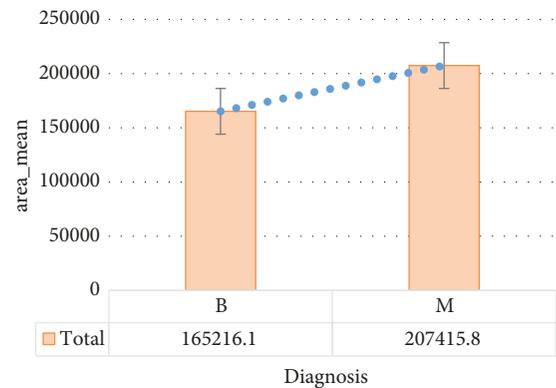


FIGURE 5: Diagnosis vs. area mean.

labels. Equations (1)–(3) mathematically describe the sigmoid function [11].

$$h(x) = g(\theta^T x), \tag{1}$$

$$g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}, \tag{2}$$

$$g(z) = \frac{1}{1 + e^{-z}}, \tag{3}$$

where $h(x)$ is defined as the model and $g(\theta^T x)$ is the sigmoid function. θ^T is the model parameters and x represents the input features. After the sigmoid function is computed, the cost function is calculated and the results from the cost

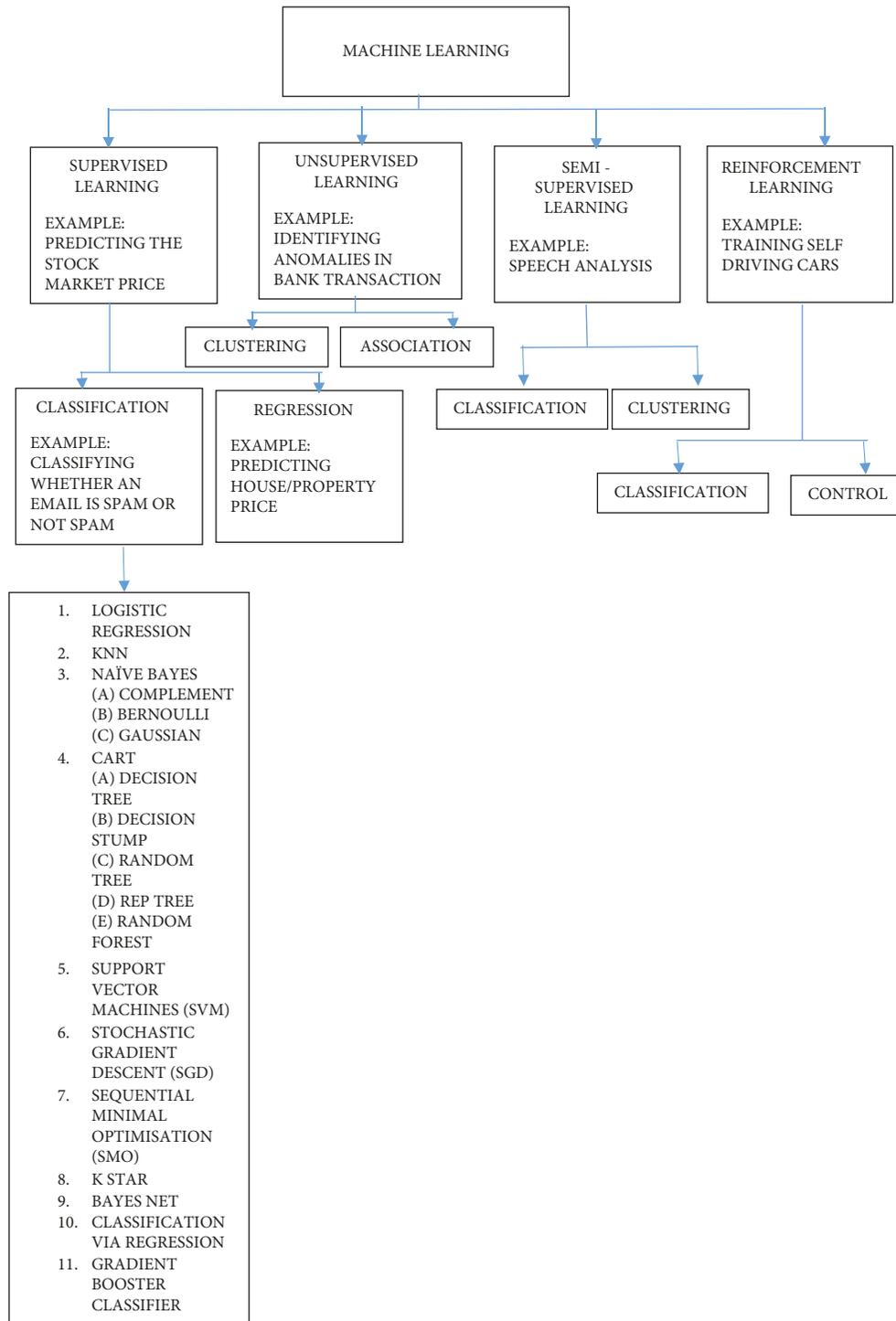


FIGURE 6: Types of machine learning and classification algorithms.

function are later optimized using a gradient descent algorithm.

3.2. K Nearest Neighbors (KNN). KNN is a widely used method for various real-time applications dealing with the classification of classes. The classification can be either binary in which only two output classes will be available or

multiclass classification, in which more than two output classes will be available. The KNN algorithm is a lazy learning and highly computational algorithm [12]. This algorithm uses the principle of matching the characteristic label of an unknown data sample with the characteristic label of a neighboring data sample based on Euclidean distance or, in some cases, the Manhattan distance [13]. Euclidean distance is the square root of the distance

between the two points in the sample space. The following equation (4) gives the Euclidean distance formula [14]:

$$d(x_i, x_j) = \sqrt{\sum_{n=1}^m (x_{in} - x_{jn})^2}, \quad (4)$$

where $d(x_i, x_j)$ is the distance between a known data sample and an unknown data sample and x_i, x_j are the location of the data samples. The “ K ” in KNN is the total number of neighbors that are selected for making predictions. It should be purely user-defined for fine prediction or else should be an odd square of the total number of data in the dataset chosen [15]. When a new, unknown data sample is introduced into the search space, the distance between the unknown data sample and the k nearest neighbors is determined. The least proximity value among the unfamiliar data sample to the nearest neighbor will determine the class for the unknown data sample.

3.3. Naive Bayes Classifier. It is a Bayes theorem-based conditional probabilistic machine learning model.

3.3.1. Bayes Theorem. The Bayes theorem in probability and statistics defines the likelihood of every occurrence based on prior knowledge of circumstances that may be relevant to the event. The following equation expresses the Bayes theorem mathematically [16]:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}, \quad (5)$$

where A and B are events and $P(B) \neq 0$.

Hence, utilizing this hypothesis, it is feasible to discover the likelihood of an incident, say A , given that B has happened. In this case, B represents the evidence and A represents the hypothesis. The assumption here is that the existence of one trait has no effect on the other. As a result, it is said to be naive. Saritas and Yasar used the Naive Bayes model and ANN in Breast Cancer Coimbra Dataset and found that the classes were classified with an accuracy of 86.95% with ANN and 83.54% with Naive Bayes algorithms, thus proving that Naive Bayes algorithms could be used for early breast cancer detection [17].

3.3.2. Types of Naive Bayes Classifier

(1) Gaussian Naive Bayes Classifier (GNB). When the predictors take on a continuous value in GNB, we assume that these values have a Gaussian or normal distribution. Consider training data with a continuous attribute “ X .” After dividing the data by class, the mean and variance of “ X ” in each class are computed. The probability distribution of some observation value “ X_i ” given a class “ Y ” can be computed by using the following equation [16]:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} e^{-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}}, \quad (6)$$

where σ_y is the variance of values associated with class y and μ_y is the mean of values associated with the class y .

Piryonesi and El-Diraby explored the performance of different classification algorithms and used it for the examination of asphalt pavement degradation data. According to their findings, GNB can significantly improve the classifier’s accuracy [18].

(2) Bernoulli’s Naive Bayes Classifier (BNB). Features in the multivariate Bernoulli event model are independent Booleans (binary variables) that describe inputs. This model, like the multinomial model, is popular for document classification problems where binary term occurrence characteristics are utilized rather than term frequencies.

(3) Complement Naive Bayes Classifier (CNB). The complement Naive Bayes classifier, also called CNB, works well with imbalanced datasets. The main principle behind the CNB is that the probability of a feature or data sample is calculated for all the classes to which the data sample belongs and not just for a single particular class. The main disadvantage of CNB is that the model tends to overfit easily when there is a large number of data samples in the dataset.

3.4. CART

3.4.1. Decision Tree. The decision tree algorithm is used for many classification problems. The main advantage of the decision tree is that it finds hidden patterns from the data by pruning the dataset and delivers high accuracy [19]. Pruning is the process of removing unwanted nodes from the decision tree that does not provide meaningful insights into the classification process. The decision tree consists of three components, mainly the root, node, and leaves. Root and node classify the unknown data based on if-then criteria and deliver the output to the next layer of nodes. Leaves are the ends of a decision tree that provides the filtered result after pruning. There are different algorithms to implement a decision tree and one such simple way to implement the same is using an Iterative Dichotomiser 3 (ID3) algorithm. The ID3 algorithm uses information gain and entropy to determine the output labels. Entropy measures the impurity of data and information gain is used to determine the amount of information that the data contributes to the final outcome [20]. The information gain and the entropy are measured using equations (7) and (8) [21].

$$H(s) = \sum_{c \in C} -P(c) \log_2 P(c), \quad (7)$$

where $H(s)$ is the entropy, s is the input data, and c is the number of output classes.

$$IG(A, S) = H(S) - \sum_{t \in T} P(t)H(t), \quad (8)$$

where $IG(A, S)$ is the information gain from the data S depending on output A , t is the subset after the data is split, $P(t)$ is the number of elements in t , and $H(t)$ is the entropy of the subset.

3.4.2. Decision Stump. The decision stump algorithm is similar to decision trees. In this algorithm, there is only one decision tree which consists of one root and a maximum of two leaf nodes for output predictions [22]. Because of this reason, decision stumps are weak learners and do not subclassify like decision trees. The algorithm terminates as soon as the first classification is executed. The overall prediction can be determined using the following equation [23]:

$$F(x) = \sum_{i=1}^n g^i(x_i), \quad (9)$$

where x is the input features, g^i is the predicted value of the i^{th} feature, and x_i is the i^{th} feature.

3.4.3. Random Trees. Random tree classifiers are a combination of decision trees and random forests. It is also called Random Forest Tree (RFT). It belongs to the category of ensemble classifiers. Random Trees improve the performance of weak learners as well as the accuracy of the machine learning model. A random tree is defined in three steps as (i) defining the method of splitting the leaves in the tree; (ii) selecting the predictor type for every leaf in the tree; (iii) providing the randomness into the leaves. One major disadvantage of random trees is that the classifier often tends to overfit the model [24].

3.4.4. Reptree. Reptree, also called representative tree algorithm, belongs to the family of decision trees and is one of the fast learner algorithms [25]. In this algorithm, multiple decision trees are created and the one with the highest probability is chosen as the representative, and based on this, other trees will be classified [26]. The reptree algorithm is a combination or ensemble of various individual trees or individual learners.

3.4.5. Random Forests. Random forest is the combination of many decision trees by randomly selecting the features and bootstrapping [27]. The random forest will first select a limited number of features from the entire set of input features and split those random features into nodes. The nodes, in turn, are split into subnodes or daughter nodes and the process is carried out until the required nodes are satisfied. The same process is repeated for “ n ” number of times creating “ D ” decision trees. The pseudocode for random forest is given in Table 1.

3.5. Support Vector Machines (SVM). SVMs provide highly accurate results with linear data as well as nonlinear data. In SVM, the input features are defined as input vectors, and multiple hyperplanes are drawn between these input features. The hyperplane acts as a decision boundary in SVM. The main objective of a hyperplane is to maximize the distance between two or more input features [28]. Because of this characteristic of the hyperplane, SVMs work well with nonlinear data. A hyperplane can be expressed using the following equation [29]:

$$W^T x + b = 0, \quad (10)$$

where x is the input vector containing the input features, b is the bias, and W is the weight that defines the distance separating the hyperplane. The best hyperplane is decided by finding different hyperplanes which classify the labels in the best way; then it will choose the one which is farthest from the data points or the one which has a maximum margin as depicted in Figure 7.

3.6. Stochastic Gradient Descent (SGD). SGD is a sophisticated version of gradient descent (GD). The only difference between GD and SGD is that the gradient descent is calculated for only a limited number of data samples. SGD selects data samples in a randomized manner and computes the gradients. Once the gradient is computed, the weights and the corresponding output labels are updated for all the data samples that were considered. This is an iterative process and continues until a minimum cost is reached [30]. The following equation is used to calculate the gradient descent [31]:

$$\nabla f(x) = \frac{1}{n} \sum_{i=1}^n \nabla f_i(x), \quad (11)$$

where $f(x)$ is the objective function, n is the total number of data samples, $\nabla f(x)$ is the gradient descent of objective function, and $\nabla f_i(x)$ is the gradient descent that is computed for limited data samples randomly. Once the gradients are computed for the selected data samples, the weights are updated using the following equation [32]:

$$x \leftarrow x - \alpha \nabla f_i(x), \quad (12)$$

where x is the data sample and α is the learning rate.

3.7. Sequential Minimal Optimization (SMO). SMO is used for solving quadratic programming problems. It is used to train the SVM classifier using a Gaussian or polynomial kernel [32]. It converts the attributes or features into binary values and also replaces missing values in the input features. SMO breaks down a quadratic problem into linear complexity. Then this linear complexity is solved using Lagrange multipliers α_1 and α_2 . The constraints for the selection of Lagrange multipliers are given in equations (13) and (14) [33]:

$$0 \leq \alpha_1, \alpha_2 \leq M, \quad (13)$$

$$y_1 \alpha_1 + y_2 \alpha_2 = K, \quad (14)$$

where M is an SVM hyperparameter and K is the kernel function, both supplied by the user; and the variables α_1, α_2 are Lagrange multipliers. SMO can be used when there is a class imbalance between the input features and works well with binary as well as multiclass classification [34].

3.8. K Star. K star or K^* is one of the lazy algorithms used for many complex classification problems. It is also called an instance-based classification algorithm [35]. Every data

TABLE 1: Pseudocode for random forest.

(1)	Start
(2)	Define the input features Training the data
(3)	(i) Select “ i ” random features from the entire set of input features i (ii) Determine the nodes from the “ i ” features using the best split point
(4)	Repeat (i) The above three steps until a specific number of nodes are reached
(5)	Repeat (i) The above steps for n times to create m decision trees
(6)	Testing the data (i) Feed-in a new data sample and predict the output label (ii) The node which gives high prediction among the other output nodes will be taken as the final prediction
(7)	Stop

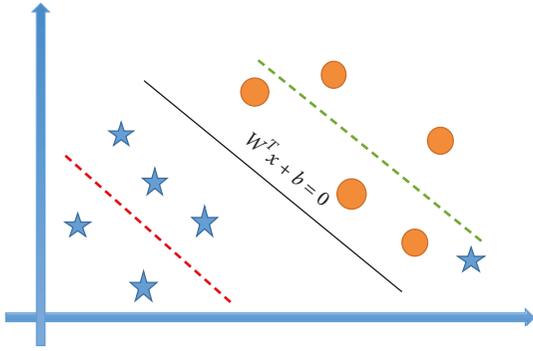


FIGURE 7: SVM hyperplane.

sample is an instance of this algorithm. The class of an instance of an unknown data sample depends on the entropic distance between that unknown data sample and the known data sample in the search space [36].

3.9. Bayes Net. A Bayesian Network, also called Bayes Net, is one of the classification algorithms used for solving machine learning problems. A Bayes Net is constructed using DAGs—Directed Acyclic Graphs. These DAGs consist of nodes and edges. The nodes represent a random variable, and the edges are used to establish the connection between two or multiple different nodes and carry the information regarding the probabilities between nodes. The only prerequisite of a DAG is that all the nodes must be connected in an open-loop and there must be no closed connection between the two nodes. The information sharing between the nodes depends on the Parental Markov Condition (PMC). PMC states that a variable represented by a node is independent of all variables in a nondescendant node and conditional on the parent node. The probabilities are calculated using the conditional probability formula as mentioned in the following equation [37]:

$$P(A|B) = \frac{P(A \cap B)}{P(B)}. \quad (15)$$

3.10. Classification via Regression. In classification via regression algorithm, first regression is carried out. Classification is performed from the output features obtained from the regression model and finally, the output class is predicted using

logistic regression. The main disadvantage of this algorithm is that it provides fluctuating results. It is both a weak learner and a strong learner. The algorithm performs well when the number of input features is limited and if the input features are more, the accuracies may vary.

3.11. Gradient Boosting Classifier. Gradient boosting classifier is an ensemble of various weak learners which is used for both regressions and classification problems. A gradient boosting classifier performs gradient descent on a weak learner-like decision tree. This loss function can be defined as in the following equation[38]:

$$f(x) = px + q + e, \quad (16)$$

where $f(x)$ is the loss function, e is the error, x is the vector containing input features, and p, q are weights to the loss function. The gradient boosting classifier can be applied to more complex datasets and it provides highly accurate results for nonlinear data.

3.12. AdaBoost M1. AdaBoost, also called Adaptive Boosting, is an ensemble-based algorithm that can be used for both binary and multiclass classification. In this boosting algorithm, classification is carried out by defining a threshold or weightage value for every classifier. The final output class is determined based on weighted votes from the individual classifiers. If there are any weak classifiers, then extra weights are given to the samples of these classifiers. This addition of extra weights to weak classifiers converts the weak learner into a strong learner [39, 40]. The weights for the individual classifier are determined using the following equation [41]:

$$\alpha_i = \frac{1}{2} \ln \left(\frac{1 - \varepsilon_i}{\varepsilon_i} \right), \quad (17)$$

where α_i is the weight for sample i and ε_i is the weighted error for sample i . The final output prediction is determined using the following equation [42]:

$$H(s) = \sum_{i=1}^n \alpha_i h_i(s), \quad (18)$$

where $H(s)$ is the final output prediction and $h_i(s)$ is the output prediction for each sample in a classifier.

3.13. LogitBoost. LogitBoost is also called an additive logistic regression algorithm [43]. LogitBoost is a weak classifier. LogitBoost takes different data samples while training and produces a weak output prediction. Boosting is applied to this weak model to improve performance and produce high output predictions. LogitBoost works well with both binary and multiclass classification. LogitBoost is similar to the AdaBoost algorithm. The only difference between the two algorithms is that the logit boost algorithm produces a weak classifier during the initial period of training the model. The output predictions are determined by using the following equation [44]:

$$P(x) = \frac{e^{f(x)}}{e^{f(x)} + e^{-f(x)}}, \quad (19)$$

where $P(x)$ are the probabilities of output class and $f(x)$ is the fitness function

3.14. Bagging. Bagging is also called Bootstrap Aggregation. It is a powerful ensemble algorithm that works well with small datasets. Bagging is used to reduce the variance in a trained model and prevents overfitting of data while training [45]. In this algorithm, the dataset is split into clusters. Classification algorithms like decision trees and random forest are applied to this cluster, and predictions are computed. The predictions from the individual clusters are averaged and through a voting basis, the final output prediction is determined [46, 47]. The pseudocode for bagging is given in Table 2.

4. Evaluation Metrics

The performance of the above-discussed 20 classification algorithms is evaluated using the parameters discussed below. A confusion matrix with TP, TN, FP, and FN for actual and predicted data samples is formed as seen in Table 3. A confusion matrix is a table that allows you to examine how an algorithm performs. Each column of the matrix contains examples from an actual class, whereas each row represents instances from a projected class [11].

The letter TP denotes that the actual value was positive and that the model projected a positive result. The letter TN denotes that the actual value was negative and that the model expected a negative result. In both of these examples, the projected value corresponds to the actual value. Whereas FP denotes a negative actual value but a positive prediction from the model, FN denotes a positive actual value but a negative prediction from the model. In both cases, the predicted value was falsely predicted. Based on these 4 values, accuracy, precision, recall, F1 measure, sensitivity, and specificity can be calculated.

Accuracy gives the proportion of properly identified samples to total samples. It is obtained from the formula [48, 49],

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{(\text{TP} + \text{TN} + \text{FP} + \text{FN})}. \quad (20)$$

Precision tells how many of the correctly anticipated samples were positive. It is calculated by dividing all positive cases by all positive examples projected to be positive [48].

$$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})}. \quad (21)$$

Similarly, *recall* indicates how many of the actual positive samples were properly predicted by the model. In medical circumstances, recall is critical since positive cases should not go undiscovered. In the case of this dataset, recall would be a preferable statistic because we do not want to mistake a malignant individual for a benign person. Precision and recall would decide whether or not the model is dependable [48].

$$\text{Recall} = \frac{\text{TP}}{(\text{TP} + \text{FN})}. \quad (22)$$

The *F1 metric* is a harmonic mean of recall and precision; hence it provides a comprehensive overview of these two measures. When we strive to improve our model's precision, the recall decreases, and vice versa. The F1-score incorporates both trends into a single number [48].

$$\text{F1 - measure} = \frac{2\text{TP}}{(2\text{TP} + \text{FP} + \text{FN})}. \quad (23)$$

Sensitivity is the ability of a test to correctly classify a sample as malignant/positive. It is defined as the rate of the predicted positive case to the total positive cases [48].

$$\text{Sensitivity} = \frac{\text{TP}}{(\text{TP} + \text{FN})}. \quad (24)$$

Specificity, on the other hand, is the ability of a test to correctly classify a sample as negative/benign. It gives the relationship of observed negative examples with all negative examples [48].

$$\text{Specificity} = \frac{\text{TN}}{(\text{TN} + \text{FP})}. \quad (25)$$

In addition to the parameters mentioned above, three model evaluation metrics in R are utilized to examine the performance of implementing algorithms (i.e., Kappa value, Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE)). The Kappa value, often known as Cohen's Kappa, is a measure of categorization accuracy. It is normalized to the dataset's random chance baseline. It is a statistic that compares the observed and expected accuracy. The Kappa statistic is used to evaluate not just a single classifier but also subclassifiers.

The MAE is the average of all absolute errors. It is also defined as the difference between the expected and actual values. References [50, 51] give the formula

TABLE 2: Pseudocode for bagging.

(1)	Start
(2)	Define the input features and the output label
(3)	Split data into training and testing sets
	Training the model
(4)	(i) Perform bootstrap process on the training data
	(ii) Choose the number of clusters to be formed based on the data
	(iii) Apply a classification algorithm like a decision tree
	Evaluate the model on the test set
(5)	(i) Compute the accuracy from each cluster in the model and average the accuracy
	(ii) Determine the output prediction by a voting process
(6)	Stop

TABLE 3: Confusion matrix representation.

	Benign	Malignant
Benign	TP	FP
Malignant	FN	TN

$$MAE = \frac{\sum_{i=1}^n |y_i - x_i|}{n}, \quad (26)$$

where n is the total number of samples, y_i is the predicted value, and x_i is the actual value.

Root mean square error (RMSE). It is equal to the square root of the mean of the squares of all mistakes. RMSE is widely used and regarded as an excellent error metric for various sorts of predictions [51, 52].

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - x_i)^2}{n}}. \quad (27)$$

5. Results and Discussions

In this study, 20 various classification algorithms were discussed in the previous section. To analyze its performance, all the algorithms were applied to Wisconsin’s Breast Cancer dataset to predict whether the cancer is benign or malignant. The analysis was justified based on various parameters such as precision, F1 measure, confusion matrix, recall, accuracy metrics, and run time discussed in Section 4. As discussed in Section 2, the dataset contains 569 samples, each with 33 attributes. The dataset is preprocessed by removing the unnecessary attributes which do not contribute to the output and checking for any incomplete data. After that, the dataset is divided into 426 training samples and 143 testing samples. All algorithms are trained on training samples before being verified on test data. The evaluation parameters given in this section are obtained from the validation of test samples. The current method employs a data driven approach. Based on historical data, the model will be able to detect breast cancer. Also, this method is faster in breast cancer detection when compared to traditional diagnostic methods on basis of performance and time. The novelty of this method is attributed to the fact that the ensemble model developed for the current study consists of combination of boosting and optimization algorithms which will dynamically reduce the errors within few iterations.

5.1. Experimental Platform. In this study, an Intel Core i5 powered 2.30 GHz/8.00 GB system running under the Windows operating system is used for processing purposes. Out of the 20 discussed algorithms, 9 of them (Logistic regression; KNN; Naive Bayes–Complement, Bernoulli, Gaussian; Decision tree; Random forest; Gradient Booster; Support Vector Machine) are run on Google Colab notebooks and coded using Python language. The Scikit-learn package is used to train and test the samples. The remaining 11 algorithms (SGD, SMO, K star, Decision Stump, random tree, REPTree, BayesNet, classification via regression, AdaBoost M1, LogitBoost, and bagging) are run using WEKA software which is a workbench for machine learning.

5.2. Experimental Results. Train and test data are split in the ratio of 80:20. The TP, TN, FP, and FN values of the implemented algorithms are given in Table 4. Based on these 4 values, accuracy, precision, recall, F1 measure, sensitivity, and specificity are calculated using equations (20)–(25) and tabulated in Table 5. Apart from the above-discussed parameters, three model evaluation metrics in “ R ” are also calculated using equations (26) and (27) and tabulated in Table 6. Also, the accuracy and run time of the algorithms are graphically depicted in Figures 8 and 9.

5.3. Discussions. The hyperparameters used to implement the algorithms are summarized in Table 7. From Table 5 it is clearly understood that SGD yields a higher accuracy of 98%, followed by random forest, SMO, and gradient booster with 97%. BNB yields the least accuracy of 63%. This is due to the fact that BNB is used for discrete data and operates on the Bernoulli distribution. The key characteristic of BNB is that it only takes binary values such as true or false. The Breast Cancer dataset has a continuous range of values for all the features. This is why BNB is not able to produce higher accuracy. All other algorithms yield accuracy ranging from 90% to 96%. KNN was able to provide an accuracy of 95% with its K value set to 4. The K value was decided using a search method where, for each K value from 1 to 10, the error obtained was plotted and visualized in Figure 10.

Smaller K values can be noisy and have a greater effect on the outcome, but bigger K values have smoother decision boundaries, which means reduced variance but increased bias and is computationally intensive. Hence an optimal K value of 4 was selected. The type of solver and learning rate

TABLE 4: Confusion matrix (TP, TN, FP, FN) of the implemented classifier.

Classifier	TP	TN	FP	FN
KNN	89	47	1	6
Logistic regression	85	51	5	2
CNB	89	44	1	9
BNB	93	0	0	53
GNB	88	48	2	5
Decision tree	86	50	4	3
Decision stump	80	47	6	9
Random tree	80	55	6	1
REPTree	77	51	9	5
Random forest	88	51	2	2
SVM	89	47	1	6
SGD	85	54	1	2
SMO	84	54	2	2
K star	84	50	2	6
Bayes net	81	52	5	4
Classification via regression	82	53	4	3
Gradient booster	88	51	2	2
AdaBoost M1	82	53	4	3
LogitBoost	84	52	2	4
Bagging	83	52	3	4

TABLE 5: Performance of implemented classifiers.

Classifier	Precision	Recall	F1 measure	Accuracy	Sensitivity	Specificity	Roc area
KNN	0.97	0.88	0.93	0.95	0.93	0.98	0.94
Logistic regression	0.91	0.96	0.93	0.95	0.97	0.91	0.95
CNB	0.97	0.83	0.89	0.93	0.90	0.97	0.90
BNB	0.00	0.00	0.00	0.63	0.63	0.00	0.50
GNB	0.96	0.90	0.93	0.95	0.94	0.96	0.94
Decision tree	0.92	0.94	0.93	0.95	0.96	0.93	0.95
Decision stump	0.88	0.83	0.86	0.89	0.89	0.89	0.85
Random tree	0.90	0.98	0.94	0.95	0.98	0.90	0.96
REPTree	0.85	0.91	0.87	0.90	0.93	0.85	0.93
Random forest	0.96	0.96	0.96	0.97	0.98	0.96	0.97
SVM	0.98	0.89	0.93	0.95	0.93	0.97	0.93
SGD	0.98	0.96	0.97	0.98	0.97	0.98	0.97
SMO	0.96	0.96	0.96	0.97	0.97	0.96	0.97
K star	0.96	0.89	0.92	0.94	0.93	0.96	0.98
Bayes net	0.91	0.93	0.92	0.94	0.95	0.91	0.99
Classification via regression	0.93	0.94	0.93	0.95	0.96	0.93	0.98
Gradient booster	0.96	0.96	0.96	0.97	0.98	0.96	0.97
AdaBoost M1	0.93	0.95	0.94	0.95	0.96	0.93	0.99
LogitBoost	0.96	0.92	0.95	0.96	0.95	0.96	0.99
Bagging	0.95	0.93	0.94	0.95	0.95	0.95	0.99

used for logistic regression was selected based on the grid search method. Three solvers “newton-cg,” “liblinear,” and “saga” were used and the learning rate was set in a range from 0.001 to 10. Logistic regression was able to yield an accuracy of 95% with a “Liblinear” solver and a learning rate of 10.

Figure 9 shows the run time of all implemented algorithms. The gradient booster algorithm was found to take more time than 7.74 seconds, followed by logistic regression with 4.3 seconds. This is because the gradient booster tends to overfit the model. Hence, setting a large number of boosting results increases the performance of the model. In this case, the boosting stages were set to

2000 which caused the algorithm to take more time. The same is in the case of a random forest. As discussed before, the grid search method was used in logistic regression to find the optimal solver and learning rate which resulted in more time. Decision tree was able to provide an accuracy of only 92%. The tree diagram for the dataset is given in Figure 11.

To increase the accuracy, pruning was executed. Pruning reduces the size of decision trees by removing nodes of the tree that do not contribute to model training. Pruning can also improve classification accuracy. The pruned decision was able to provide an accuracy of 95% and the same tree is shown in Figure 12.

TABLE 6: Kappa value, MAE, RMSE, and run time of implemented algorithms.

Classifier	Kappa value	MAE	RMSE
KNN	0.895	0.048	0.221
Logistic regression	0.892	0.048	0.221
CNB	0.849	0.069	0.264
BNB	0.006	0.370	0.608
GNB	0.897	0.048	0.221
Decision tree	0.891	0.048	0.221
Decision stump	0.773	0.168	0.310
Random tree	0.897	0.049	0.222
REPTree	0.799	0.128	0.287
Random forest	0.946	0.027	0.167
SVM	0.897	0.050	0.221
SGD	0.962	0.021	0.145
SMO	0.946	0.028	0.167
K star	0.885	0.057	0.228
Bayes net	0.878	0.059	0.236
Classification via regression	0.894	0.085	0.207
Gradient booster	0.942	0.030	0.170
AdaBoost M1	0.896	0.058	0.189
LogitBoost	0.917	0.054	0.174
Bagging	0.895	0.083	0.188

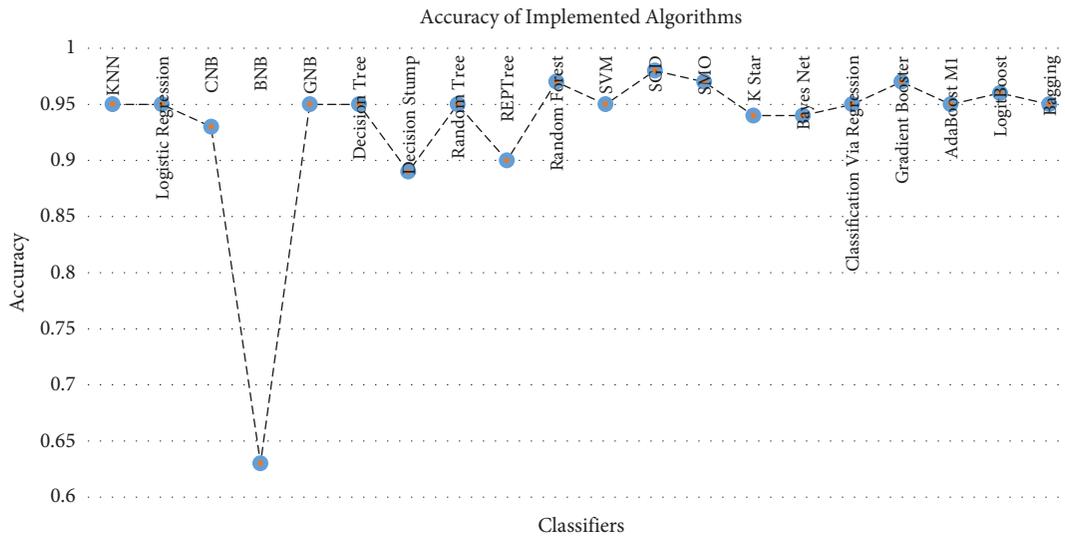


FIGURE 8: Accuracy of implemented algorithms.

5.4. *Development of Novel Ensemble Model.* The ultimate objective of every ML challenge is to find a model that can predict the desired outcome with high accuracy. Rather than creating a single model and hoping it is the best, we can use ensemble to take a variety of diverse models into consideration and average those models to get a single final model. In other words, the ensemble approach is a machine learning strategy that integrates numerous basic models to build a single highly accurate model. Common ensemble algorithms such as bagging, AdaBoost, LogitBoost, and gradient boosting are discussed above and implemented. In addition to that, a novel ensemble model has been developed combining logistic regression, stochastic gradient descent, and random forest. The hyperparameters of the developed

ensemble model are given in Table 8. The performance of this ensemble model is investigated by implementing it on both platforms (i.e., Colab notebook and WEKA software). The results obtained are compared and discussed further in the next section.

5.5. *Comparative Study between WEKA and Python Coded Results.* A comparative study has been performed between WEKA results and Python coded results. The ensemble model developed above has been used to perform this comparative study. Table 9 summarizes the results obtained through WEKA and Python by ensemble of the three algorithms.

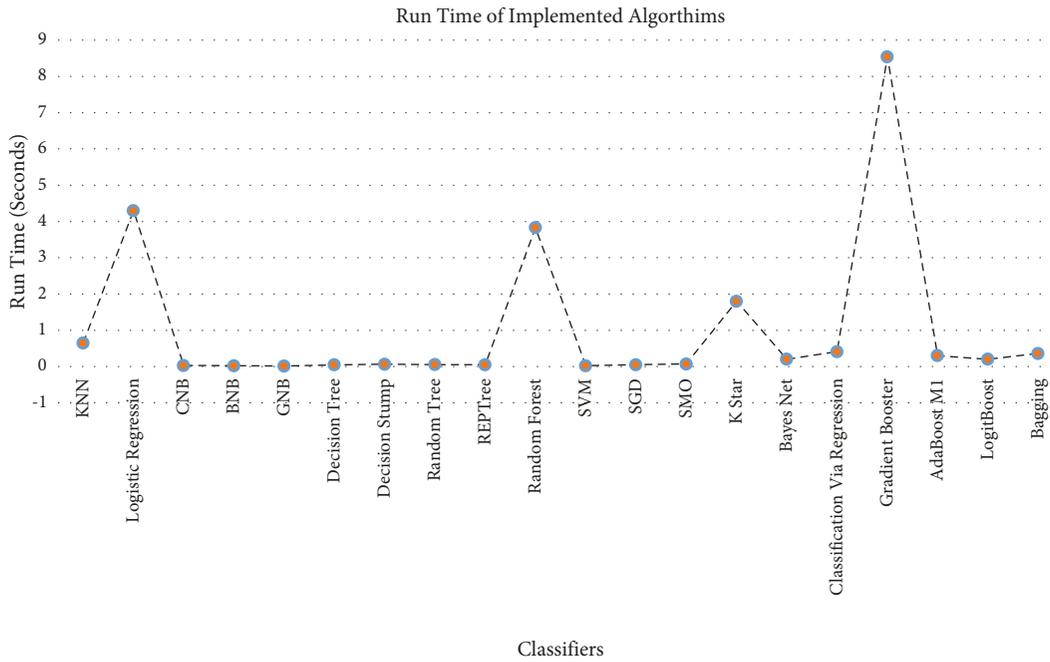


FIGURE 9: Run time of implemented algorithms.

TABLE 7: Hyperparameters of the implemented 20 classifiers.

Classifiers	Hyperparameters
KNN	Nearest neighbors—4; brute algorithm
Logistic regression	Solvers—Newton-cg, liblinear, saga; learning rate—0.001, 0.01, 0.1, 1; penalty—l2, max. iterations—2000
CNB, BNB, GNB	Alpha (smoothing parameter)—1.0 (default) var_smoothing—1.00e-09
Decision tree	Criterion—“gini”; ccp_alpha (minimal cost-complexity pruning)—0.1; splitter—“best”
Decision stump	Batch size—100
Random tree	K value—1; batch size—100; max. depth—2
REPTree	Min variance prop—0.001; batch size—100; num folds—3
Random forest	n_estimators—2000; criterion—“gini”; bootstrap—true; n_jobs—-1; ccp_alpha—0.1
SVM	C (regularization parameter)—1.0; kernel—“sigmoid”
SGD	Loss—“log”; penalty—l2; alpha—0.01
SMO	Calibrator—“logistic”; kernel—“poly Kernel”; C—0.1; epsilon—1.00e-12
K star	Batch size—100; global blend—20; entropic auto blend—true
Bayes net	Batch size—100; estimator—“simple”; search algorithm—“hill climber”
Classification via regression	Batch size—100; regression type—“additive”
Gradient boosting classifier	n_estimators—2000; loss —“deviance”; learning rate—0.01
AdaBoost M1	Batch size—100; weight threshold—100; seed—1; iterations—0
LogitBoost	Batch size—100; weight threshold—100; seed—1; iterations—50; z max—5; shrinkage—1; pool size—1
Bagging	Seed—1; iterations—50; batch size—100; bag size percent—50

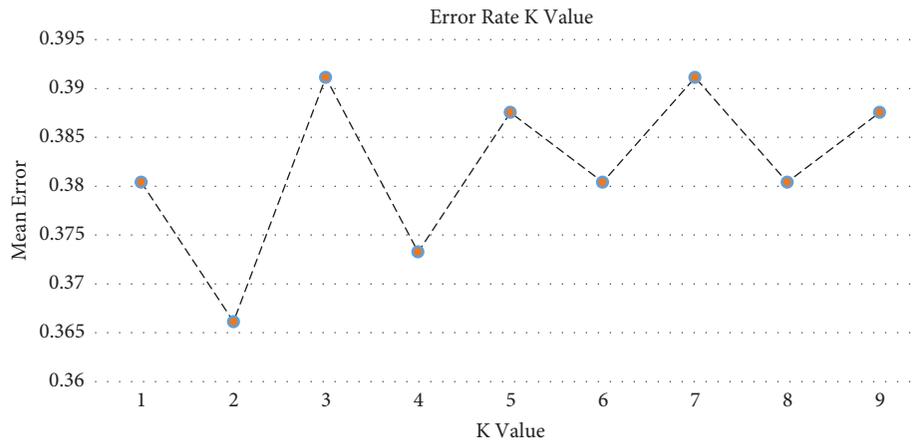


FIGURE 10: Error rate vs. K value for KNN.

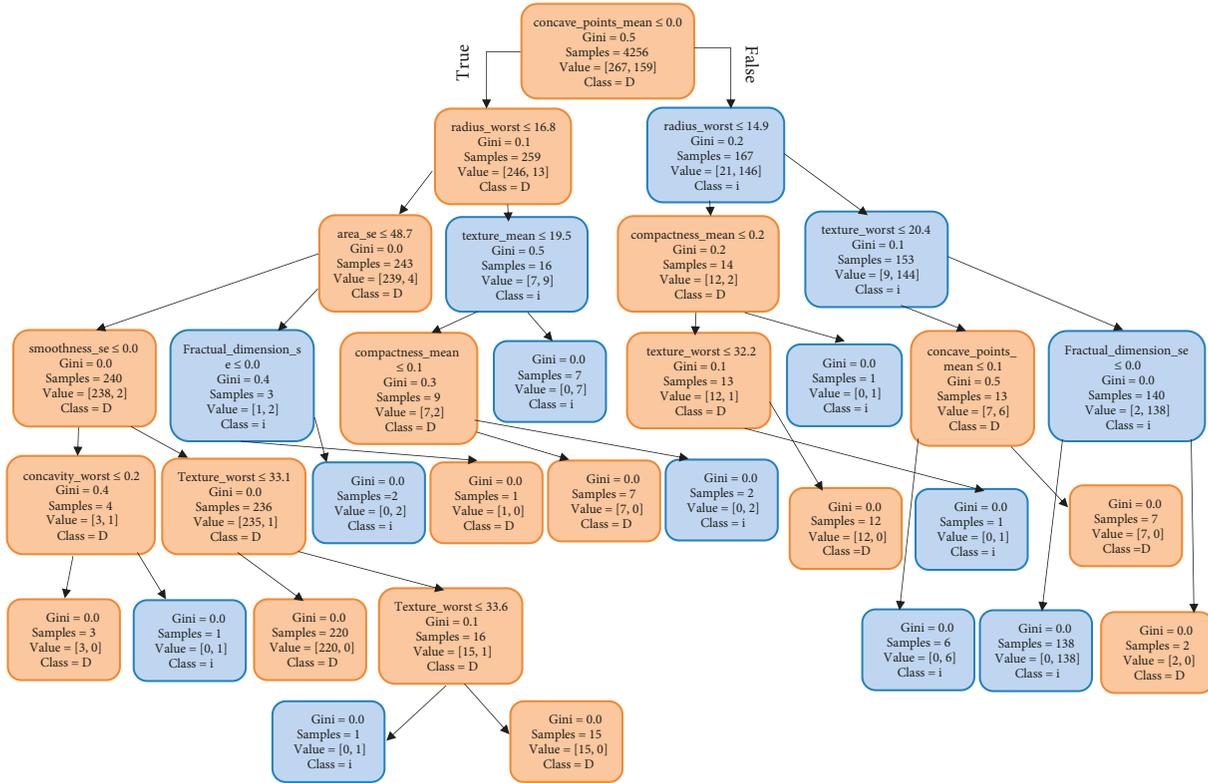


FIGURE 11: Tree diagram before pruning.

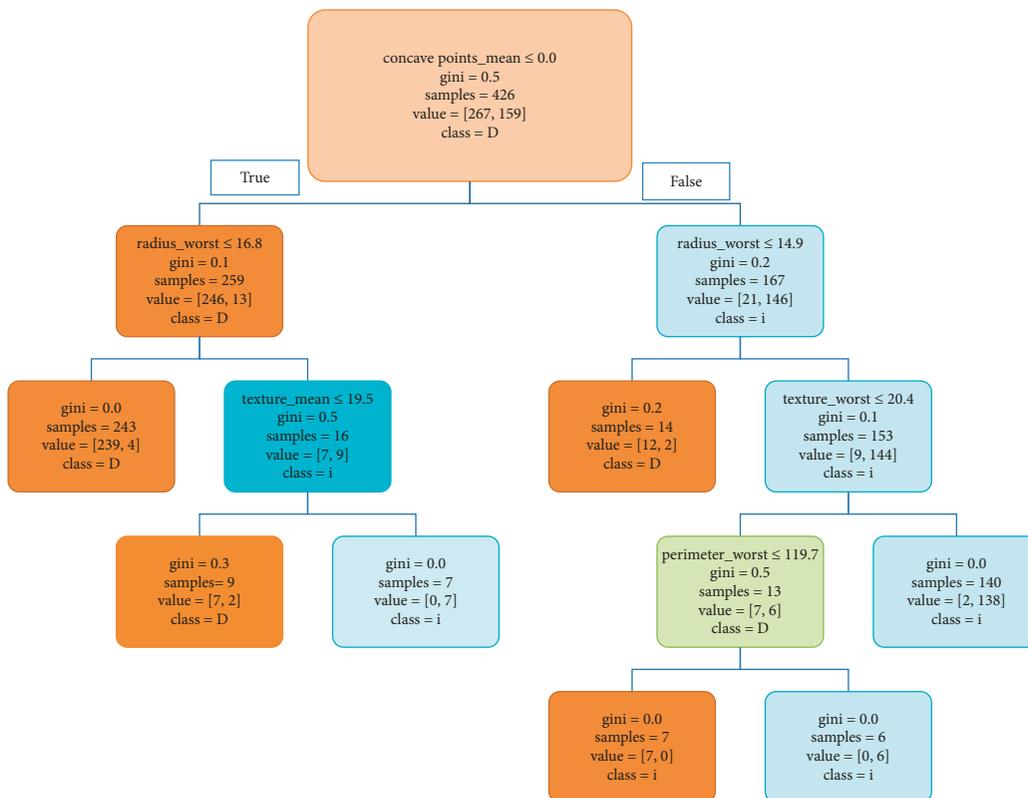


FIGURE 12: Tree diagram after pruning.

TABLE 8: Hyperparameters of the ensemble model.

Classifiers in ensemble model	Hyperparameters
Logistic regression	Solvers—Newton-cg, liblinear, saga; learning rate—0.001, 0.01, 0.1, 1; penalty—l2, max. iterations—2000 (grid search)
Random forests	$n_{\text{estimators}}$ —2000; criterion—“gini”; bootstrap—true; min. sample split—2
Stochastic gradient descent (SGD)	Loss—“log”; penalty—l2; alpha—0.01

TABLE 9: Comparison between WEKA and Python coded results.

Evaluation parameters	WEKA	Python coded
Accuracy	0.970	0.985
Precision	0.974	0.970
Recall	0.974	0.990
F1 score	0.974	0.980
True positive rate (TPR)	0.970	0.986
False positive rate (FPR)	0.030	0.015
Run time (seconds)	0.780	4.940
Kappa value	0.946	0.940
Mean absolute error (MAE)	0.045	0.026
Root mean square error (RMSE)	0.138	0.162

From Table 9, it is evident that the algorithm implemented in the Python programming language yielded an accuracy of 98.5%, while the model implemented in WEKA provided an accuracy of 97%. But the run time is more when executed in Python. This is because logistic regression uses a grid search technique to find the optimal solver and learning rate while the other 2 algorithms use a high value of boosting stages to yield better results which together accounts for the more computational time when compared to WEKA. Now, from Tables 5 and 9, it is evident that the ensemble model developed yields an accuracy of 98.5%, while the highest accuracy yielded by individual classifiers is 98% (SGD). Conventional ensembles like gradient booster, AdaBoost, LogitBoost, and bagging were able to achieve an accuracy of 97%, 95%, 96%, and 95% respectively. Hence, it is concluded that the performance of the developed ensemble model using Python is better than the traditional ML algorithms.

6. Conclusion

Breast cancer is a very common type of cancer and it is predominantly found in women. In this paper, 20 different ML classification algorithms were trained on a Breast Cancer dataset from the University of Wisconsin to detect the same. From the results obtained it was found that SGD has yielded the highest accuracy of 98%, while BNB provided the least accuracy of 63%. Apart from accuracy, various other evaluation parameters like confusion matrix, precision, recall, F1 measure, sensitivity, specificity, Kappa value, MAE, RMSE, and run time were calculated and tabulated. Further, a comparative study was carried out to analyze the performance of the algorithm when coded in Python and when run on WEKA. An ensemble model has been developed combining logistic regression, random forest, and SGD algorithms and deployed in the comparative study. The performance of the ensemble model in both platforms has

been compared and investigated in detail. It was found that the ensemble model developed using Python was able to achieve an accuracy of 98.5% and the model developed in the WEKA data mining tool yielded an accuracy of 97%. We believe that development and implementation of ensemble models can improve the quality of diagnosis and brings a great advantage compared to the recent works. Future directions of this work include the development of a real-time AI-based hardware prototype that can be deployed in hospitals around the world to assist doctors in detecting breast cancer. The use of AI and ML technology is limited not only to the detection of breast cancer, but also to the detection of other deadly diseases. The same technology can automate society and one such example is described in this paper. Thus, the work in this paper is an attempt to improve and automate society, establish the concept of smart hospitals, and contribute to the development of these technologies.

Data Availability

Data will be available upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] X. Liang, X. Yu, and T. Gao, “Machine learning with magnetic resonance imaging for prediction of response to neoadjuvant chemotherapy in breast cancer: a systematic review and meta-analysis,” *European Journal of Radiology*, vol. 150, Article ID 110247, 2022.
- [2] W. Dong, W. P. Bensken, U. Kim, J. Rose, N. A. Berger, and S. M. Koroukian, “Phenotype discovery and geographic disparities of late-stage breast cancer diagnosis across U.S. Counties: a machine learning approach,” *Cancer Epidemiology Biomarkers & Prevention*, vol. 31, no. 1, pp. 66–76, 2022.
- [3] A. Pfob, B. J. Mehrara, J. A. Nelson, E. G. Wilkins, A. L. Pusic, and C. Sidey-Gibbons, “Towards patient-centered decision-making in breast cancer surgery: machine learning to predict individual patient-reported outcomes at 1-year follow-up,” *Annals of Surgery*, vol. 275, no. 4, 2022.
- [4] A. Rasool, C. Bunternghit, L. Tiejian, M. R. Islam, Q. Qu, and Q. Jiang, “Improved machine learning-based predictive models for breast cancer diagnosis,” *International Journal of Environmental Research and Public Health*, vol. 19, no. 6, p. 3211, 2022.
- [5] J. Y. Lee, K.-s. Lee, B. K. Seo et al., “Radiomic machine learning for predicting prognostic biomarkers and molecular subtypes of breast cancer using tumor heterogeneity and

- angiogenesis properties on MRI," *European Radiology*, vol. 32, no. 1, pp. 650–660, 2022.
- [6] T. Chandrasekaran, S. Chandrasekaran, R. Hua, I. Banerjee, and A. Sanyal, "A fully-integrated analog machine learning classifier for breast cancer classification," *Electronics*, vol. 9, no. 3, p. 515, 2020.
- [7] A. R. Vaka, B. Soni, and S. Reddy, "Breast Cancer Detection by Leveraging Machine Learning," *ICT Express*, vol. 6, 2020.
- [8] S. Mojriani, G. Pinter, J. H. Joloudari et al., "Hybrid machine learning model of extreme learning machine radial basis function for breast cancer detection and diagnosis; A multilayer fuzzy expert system," in *Proceedings of the RIVF International Conference on Computing and Communication Technologies (RIVF)*, pp. 1–7, Ho Chi Minh, Vietnam, Vietnam, July 2020.
- [9] V. Chaurasia and S. Pal, "Applications of machine learning techniques to predict diagnostic breast cancer," *SN Computer Science*, vol. 1, no. 5, pp. 270–311, 2020.
- [10] M. F. Ak, "A comparative analysis of breast cancer detection and diagnosis using data visualization and machine learning applications Healthcare," *Multidisciplinary Digital Publishing Institute*, vol. 8, no. 2, p. 111, 2020.
- [11] Y. Mahmood, N. Kama, A. Azmi, A. S. Khan, and M. Ali, "Software effort estimation accuracy prediction of machine learning techniques: a systematic performance evaluation," *Software: Practice and Experience*, vol. 52, no. 1, pp. 39–65, 2022.
- [12] Q. Kuang and L. Zhao, "A practical GPU based KNN algorithm," in *Proceedings of the Second Symposium International Computer Science and Computational Technology (ISCSCST '09)*, pp. 151–155, P. R. China, December 2020.
- [13] Z. Deng, X. Zhu, D. Cheng, M. Zong, and S. Zhang, "Efficient KNN Classification Algorithm for Big Data," *Neurocomputing*, vol. 195, pp. 1–6, 2016.
- [14] B. O'Shea, "K-nearest Neighbors Algorithm (KNN) and Artificial Neural Network (ANN) Accurately Predicting Malignancy of Breast Cancer (BC) Tissue Solely Based of Features Acquired from Imaging Modalities," 2020, <http://hdl.handle.net/1920/11759>.
- [15] X. Zhang, Y. Li, R. Kotagiri, L. Wu, Z. Tari, and M. Cheriet, "KRNN: k rare-class nearest neighbour classification," *Pattern Recognition*, vol. 62, pp. 33–44, 2017.
- [16] S. Agarwal, B. Jha, T. Kumar, M. Kumar, and P. Ranjan, "Hybrid of naive bayes and Gaussian naive bayes for classification: a map reduce approach," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 6S3, pp. 266–268, 2019.
- [17] M. M. Saritas and A. Yasar, "Performance analysis of ANN and naive bayes classification algorithm for data classification," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 7, no. 2, pp. 88–91, 2019.
- [18] S. M. Piryonesi and T. E. El-Diraby, "Role of data analytics in infrastructure asset management: overcoming data size and quality problems," *Journal of Transportation Engineering, Part B: Pavements*, vol. 146, no. 2, Article ID 04020022, 2020.
- [19] J. Tanha, H. Salarabadi, M. Aznab, A. Farahi, and M. Zoberi, "Relationship among prognostic indices of breast cancer using classification techniques," *Informatics in Medicine Unlocked*, vol. 18, Article ID 100265, 2020.
- [20] R. R. Putra and H. W. Dhany, "Determination of accuracy value in id3 algorithm with gini index and gain ratio with minimum size for split, minimum leaf size, and minimum gain," *IOP Conference Series: Materials Science and Engineering*, vol. 725, no. 1, Article ID 012088, 2020.
- [21] N. Yang, T. Li, and J. Song, "Construction of decision trees based entropy and rough sets under tolerance relation," in *International Conference on Intelligent Systems and Knowledge Engineering*, pp. 1515–1519, Atlantis Press, Amsterdam, Netherlands, 2007.
- [22] T. N. Shah, M. Z. Khan, M. Ali, B. Khan, and N. Idress, "CART, J-48 Graft, J48, ID3, Decision Stump and Random Forest: A comparative study," 2020.
- [23] Y. Wang, H. Zhang, H. Chen, D. Boning, and C.-J. Hsieh, "On lp-norm robustness of ensemble decision stumps and trees," in *International Conference on Machine Learning* vol. 119, PMLR, Article ID 10104, 2020.
- [24] J. H. Joloudari, E. Hassannataj Joloudari, H. Saadatfar et al., "Coronary artery disease diagnosis; ranking the significant features using a random trees model," *International Journal of Environmental Research and Public Health*, vol. 17, no. 3, p. 731, 2020.
- [25] J. S.K. and S. Sasikala, "S. S. Reptree classifier for identifying link SPAM in web search engines," *ICTACT Journal on Soft Computing*, vol. 03, no. 02, pp. 498–505, 2013.
- [26] Y. Zhao and Y. Zhang, "Comparison of decision tree methods for finding active objects," *Advances in Space Research*, vol. 41, no. 12, pp. 1955–1959, 2008.
- [27] C. Aroef, Y. Rivan, and Z. Rustam, "Comparing random forest and Support vector machines for breast cancer classification," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 18, no. 2, p. 815, 2020.
- [28] R. S. Thakare, S. M. Deshmukh, and V. R. Raut, "Automatic breast segmentation and cancer detection using svm," *Open Access International Journal of Science and Engineering*, vol. 5, no. 9, 2020.
- [29] S. V. M. Vishwanathan and M. N. Murty, "SSVM: a simple SVM algorithm," in *Proceedings of the 2002 International Joint Conference on Neural Networks*, vol. 3, pp. 2393–2398, IEEE, 2002.
- [30] A. S. Assiri, S. Nazir, and S. A. Velastin, "Breast tumor classification using an ensemble machine learning method," *Journal of Imaging*, vol. 6, no. 6, p. 39, 2020.
- [31] M. Kiknadze and A. Gürhanlı, "Prediction of breast cancer using artificial neural networks," *International Journal of Engineering Research in Africa*, vol. 10, no. 2, pp. 6–15, 2020.
- [32] Z. Sajjadnia, R. Khayami, and M. R. Moosavi, "Preprocessing Breast cancer data to improve the data quality, diagnosis procedure, and Medical Care Services," *Cancer Informatics*, vol. 19, Article ID 117693512091795, 2020.
- [33] A. N. Oo and K. T. Win, "Feature selection based Sequential Minimal Optimization (SMO) for heart disease classification," in *Proceedings of the 2009 Fifth International Joint Conference on INC, IMS and IDC, Seoul, Korea (South)*, August 2009.
- [34] S. B. Ud Din Tahir, A. Jalal, and M. Batool, "Wearable sensors for activity analysis using SMO-based random forest over smart home and sports datasets," in *Proceedings of the 2020 3rd International Conference on Advancements in Computational Sciences (ICACS)*, pp. 1–6, IEEE, Lahore, Pakistan, February 2020.
- [35] D. Y. Mahmood and M. A. Hussein, "Intrusion detection system based on K-Star classifier and feature set detection," *IOSR Journal of Computer Engineering*, vol. 15, no. 5, pp. 107–112, 2013.
- [36] T. Kavzoglu and I. Colkesen, "Entropic distance-based K-Star algorithm for remote sensing image classification," *Fresenius Environmental Bulletin*, vol. 20, no. 5, pp. 1200–1207, 2011.
- [37] S. Hartmann, "Bayes Net and Rationality," 2020, Preprint.

- [38] S. Dutta and S. K. Bandyopadhyay, "Early breast cancer prediction using artificial intelligence methods," *Journal of Engineering Research and Reports*, vol. 13, no. 2, pp. 48–54, 2020.
- [39] M. M. Rahman, Y. Ghasemi, E. Suley, Y. Zhou, S. Wang, and J. Rogers, "Machine Learning Based Computer Aided Diagnosis of Breast Cancer Utilizing Anthropometric and Clinical Features," *IRBM*, vol. 42, 2020.
- [40] E. Sevinç, "An empowered AdaBoost algorithm implementation: a COVID-19 dataset study," *Computers & Industrial Engineering*, vol. 165, Article ID 107912, 2022.
- [41] Y. Wang and L. Feng, "Improved Adaboost algorithm for classification based on noise confidence degree and weighted feature selection," *IEEE Access*, vol. 8, Article ID 153011, 2020.
- [42] D. G. Ganakwar and V. K. Kadam, "Face detection using logit boost algorithm with YCbCr color space," *International Journal for Research in Applied Science and Engineering Technology*, vol. 8, no. 1, pp. 184–189, 2020.
- [43] H. Jain, A. Khunteta, and S. Srivastava, "Churn prediction in telecommunication using logistic regression and logit boost," *Procedia Computer Science*, vol. 167, pp. 101–112, 2020.
- [44] A. M. Widodo, N. Anwar, B. Irawan, and L. Meria, "Data Mining Classification for Breast Cancer Prediction," *Procedia Computer Science*, vol. 169, 2019.
- [45] I. K. Omurlu, F. Cantas, M. Ture, and H. Ozturk, "An empirical study on performances of multilayer perceptron, logistic regression, ANFIS, KNN and bagging CART," *Journal of Statistics & Management Systems*, vol. 23, no. 4, pp. 827–841, 2020.
- [46] R. H. Saputra and B. Prasetyo, "Improve the accuracy of C4.5 algorithm using Particle Swarm Optimization (PSO) feature selection and bagging technique in breast cancer diagnosis," *Journal of Soft Computing Exploration*, vol. 1, no. 1, pp. 47–55, 2020.
- [47] R. Cristin, P. N. Jeipratha, T. Daniya, and N. Kanagavalli, "Logistic regression model with gradient descent parameter estimation," *Journal of Critical Reviews*, vol. 7, no. 12, pp. 2048–52, 2020.
- [48] A. F. Psaros, X. Meng, Z. Zou, L. Guo, and G. E. Karniadakis, "Uncertainty quantification in scientific machine learning: methods, metrics, and comparisons," 2022, <https://arxiv.org/abs/2201.07766>.
- [49] A. Suragala, P. Venkateswarlu, and M. China Raju, "A comparative study of performance metrics of data mining algorithms on medical data," in *ICCCE 2020*, pp. 1549–1556, Springer, Singapore, 2021.
- [50] M. E. Pérez-Pons, J. Parra-Dominguez, G. Hernández, E. Herrera-Viedma, and J. M. Corchado, "Evaluation metrics and dimensional reduction for binary classification algorithms: a case study on bankruptcy prediction," *The Knowledge Engineering Review*, vol. 37, p. e1, 2022.
- [51] M. Steurer, J. Hill, and P. Robert, "Norbert. Metrics for evaluating the performance of machine learning based automated valuation moders," *Journal of Property Research*, vol. 38, pp. 1–31, 2021.
- [52] C. Miller, Andrew, A. Gatys, Leon, J. Futoma, and B. Fox, "Emily. Model-Based metrics: sample-Efficient estimates of predictive model subpopulation performance," 2021, <https://arxiv.org/abs/2104.12231>.