https://dx.doi.org/10.4314/dujopas.v9i4b.8
ISSN (Print): 2476-8316
ISSN (Online): 2635-3490
Dutse Journal of Pure and Applied Sciences (DUJOPAS), *Vol. 9 No. 4b December 2023*

Comparative Analysis of Machine Learning Algorithms for Breast Cancer Prediction

Kene Tochukwu Anyachebelu[1], Sukkushe Hannah Hosea[1], Muhammad Umar Abdullahi[2], Maimuna Abdullahi Ibrahim[3]

[1]Department of Computer Science,
Nasarawa State University,
Keffi, Nigeria

[2]Department of Computer Science,
Federal University of Technology,
Owerri, Nigeria

[3]Department of Computer Science,
The Federal Polytechnic Nasarawa,
Nigeria

Email: anyachebelutk@nsuk.edu.ng

Abstract

Breast cancer is a global health concern, and early diagnosis is crucial for successful treatment. The objective of this paper is to conduct a comparative analysis of machine-learning algorithms for the prediction of breast cancer. This study used the Wisconsin Diagnostic Breast Cancer Dataset. Data preparation, technique selection, and performance evaluation are included in the study. The inquiry begins by comparing malignant and benign instances according to input factors and diagnostic outcomes. Finding components having an inverse relationship to the diagnosis is prioritized. Next, a careful approach is used to choose attributes to improve the dataset for model construction. The preprocessed data trains and optimizes four well-known machine learning algorithms: Random Forest, Support Vector Machine, K-Nearest Neighbor, and Logistic Regression. The models are evaluated for accuracy, precision, recall, F1-score, and ROC curve. This study aimed to evaluate numerous breast cancer prediction systems to determine their strengths and weaknesses. To provide openness and replicability, the study uses the Jupyter Notebook platform, Python, and data analytic tools. The logistic regression model has a test accuracy percentage of 99.26%, surpassing all other models examined in this study. Furthermore, it has a minimum false positive rate (FPR) of 1 and a false negative rate (FNR) of 4. The model exhibits a higher level of precision in comparison to the studies examined in the literature review. This study is crucial for early diagnosis and therapy development. The effects include lower healthcare expenses, better patient outcomes, and better diagnostics. Machine learning has shown promise in fighting breast cancer, boosting its relevance in healthcare.

Keyword: Breast Cancer, Prediction, Machine Learning, Algorithms, Comparative Analysis

**Author for Correspondence*

INTRODUCTION

Breast cancer, a disease with a history dating back to ancient Egypt, continues to pose a significant health challenge, particularly among women (Obaid, Mohammed, Khanapi, Ghani, Mostafa, & Taha, 2018). Recent research highlights its prevalence and underscores the importance of early detection. Breast cancer affects both men and women, but it is more common among the latter, with statistics suggesting that approximately one in eight women may face this diagnosis in their lifetime (Rufai, Muhammad, Garba, & Audu, 2020). In the United Kingdom, breast cancer is diagnosed in approximately 41,000 women annually, with significantly fewer cases among men (Islam, Haque, Iqbal, Hasan, Hasan, & Kabir, 2020). Globally, breast cancer remains a major public health concern, as evidenced by over 2.3 million cases reported in 2020, leading to approximately 685,000 deaths (Yee, Tzen, Yap, Goh, & Cher, 2022).

Breast cancer arises due to abnormal cell proliferation in the breast, which can lead to the formation of either benign or malignant tumors (Chaurasia & Pal, 2014). Benign tumors are non-cancerous, while malignant tumors are indicative of cancer (Fatima, Liu, Hong, & Ahmed, 2020). Survival rates in breast cancer vary significantly by stage, emphasizing the importance of early detection (Yee et al., 2022). To improve the accuracy and efficiency of breast cancer diagnosis, machine learning algorithms have been employed, often surpassing the accuracy of human physicians (Gupta & Garg, 2020).

Several studies have utilized machine learning to predict breast cancer. Shubham and Kamalraj (2022) used K-Nearest Neighbors (KNN), Support Vector Machine (SVM), and Decision Tree Classifier (DT) algorithms to predict breast cancer, with SVM achieving an accuracy rate of 97%. Tiwari et al. (2020) applied SVM, KNN, DT, Naïve Bayes (NB), Logistic Regression (LR), and Random Forest (RF) on a dataset with the highest accuracy of 96.5% achieved by SVM and RF. Singh (2020) employed KNN, SVM, LR, and NB, with KNN achieving exceptional performance with an accuracy rate of 98%. Rawal (2020) found that SVM and the C4.5 decision tree method had high true positive rates for benign and malignant classes, with SVM demonstrating lower false positives.

Khan et al. (2022) used Logistic Regression to achieve an accuracy rate of 98% in their breast cancer prediction model. Obaid et al. (2018) employed Support Vector Machine, K-Nearest Neighbors, and Decision Tree, with SVM achieving an impressive accuracy rate of 98.1%. Rufai et al. (2020) utilized Support Vector Machine and achieved an accuracy rate of 94.3%. Ganggayah et al. (2019) compared several machine learning models, with Random Forest achieving the highest accuracy of 82.7%. Shravya et al. (2019) applied KNN, SVM, and LR, with SVM achieving an accuracy rate of 92.7%. Yee et al. (2022) used LR, RF, SVM, and Multilayer Perceptron (MLP), with RF achieving an accuracy rate of 82%.

Rana et al. (2015) employed Support Vector Machines, Logistic Regression, K-Nearest Neighbors, and Naive Bayes, with KNN achieving an accuracy rate of 95.68%. These studies often did not specify feature selection procedures, but it is important to note that feature selection can enhance the accuracy of machine learning predictions by removing irrelevant or negatively associated input features (Rana et al., 2015).

The use of machine learning algorithms, including Support Vector Machine, K-Nearest Neighbor, Logistic Regression, and Random Forest, has shown promise in breast cancer prediction. These algorithms have been utilized in various studies, achieving high accuracy rates and outperforming human physicians. The choice of these algorithms was based on their

effectiveness and extensive use in the empirical literature. Early detection of breast cancer through machine learning holds the potential to improve survival rates and reduce the burden of this disease on individuals and healthcare systems.

METHODOLOGY

The methodology for this paper is known as Machine Learning Pipeline or Data Science Workflow. The methodology has eight stages. As shown in Figure 1. which includes dataset collection, data exploration, and pre-processing, splitting dataset, training the models, testing the models, performance evaluation, performance comparison, conclusion.

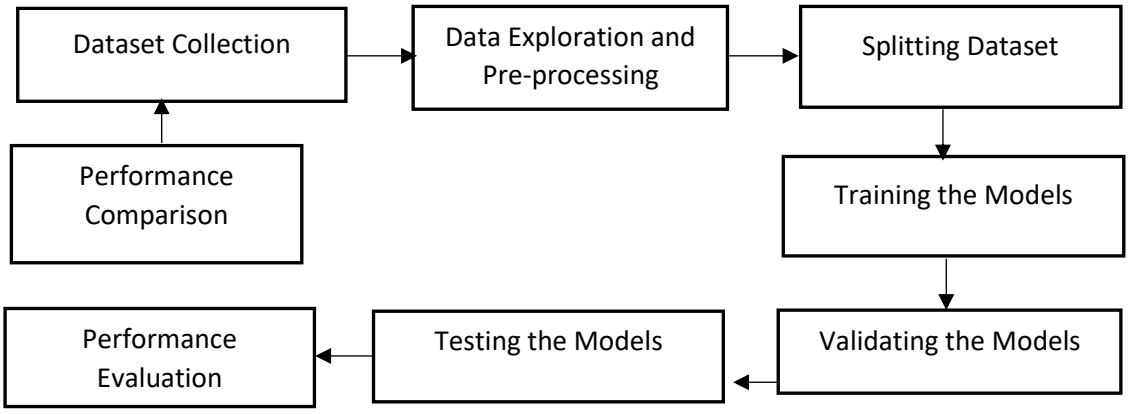

Figure 1: Research Design for the Study

Figure 1 shown in this study was modified from the original work of Rana et al. (2015) to align with the research approach utilized in this report. The primary aim of this research is to predict the malignancy or benignancy of a tumor in a patient. The attainment of this target was achieved by the systematic implementation of a well-structured series of procedures.

Dataset Collection: The initial step involved the collection of the dataset in CSV format, followed by its importation into Jupyter Notebook for further analysis.

Data Exploration and Pre-processing: This phase encompassed four key stages. First, a thorough exploration of the data was conducted to understand its characteristics. Subsequently, features with negative correlations to the target variable were identified and removed to enhance model performance. To facilitate modeling, labels representing malignancy (M) and benignity (B) were transformed into binary values, specifically 1 and 0, respectively. Finally, feature scaling was applied to ensure that all features were on a consistent scale.

Splitting the Dataset: Following the importation of the dataset, it was divided into two distinct subsets. The first subset was utilized for training the machine learning models, while the second subset was employed to evaluate the performance of these models. This step was undertaken to assess the performance of the models on an independent dataset, therefore mitigating the risk of overfitting.

Training the Models: During the training phase, four separate machine learning models were utilized, specifically Random Forest, K-Nearest Neighbour, Support Vector Machine, and Logistic Regression. The aforementioned models were employed to construct prediction algorithms.

Validation of Models: To improve the prediction performance of the machine learning models, a validation set was utilized. This particular collection played a crucial role in the process of adjusting model parameters to maximize their performance.

Testing the Models: The models, having undergone training and validation, were rigorously tested using the testing dataset to assess their accuracy and predictive power.

Performance Evaluation: To gauge the effectiveness of the models, a thorough performance evaluation was conducted. This evaluation employed metrics such as the scikit-learn accuracy score and confusion matrix.

Performance Comparison: The next phase was a thorough comparison of the four machine learning models. The goal of this comparison research was to determine which model outperformed the others in the essential duty of predicting breast cancer, therefore adding vital insights to the area of medical diagnostics.

Source of Dataset

The dataset utilized in this research, referred to as the Wisconsin Breast Cancer Diagnostic Dataset (WBCDD), comprises secondary data. The medical dataset was acquired from the publicly accessible Kaggle database found at https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data. The dataset's attributes are obtained from a digital image of a breast cancer sample that was taken by the process of fine-needle aspirate (FNA) (Gupta & Garg, 2020). The characteristics of the cell nuclei observed in the snapshot are employed to determine their attributes. The dataset referred to as WDBCD has 32 unique attributes and a cumulative count of 3414 instances. Out of the given cases, a total of 2142 instances have been categorized as benign, whereas 1272 instances have been classed as malignant.

Performance Metrics for Classification

The evaluation criteria utilized for gauging the effectiveness of this analysis are as follows:

Accuracy

The efficacy of a model is assessed by the proportion of accurate predictions produced across all sorts of forecasts. The evaluation process involves assessing the accuracy of classification by comparing the count of corrType equation here.ectly categorized instances to the overall count of occurrences. The measure of accuracy is particularly valuable in cases when the distribution of classes in the target variable is uniformly spread throughout the dataset. This is expressed in Equation 1. $Accuracy = \frac{TP+TN}{TP+FP+FN+TN}$ … (1)

Sensitivity or Recall

The sensitivity, also known as recall, is a measure of the true positive rate in the context of a software defect system. In this particular context, it denotes the number of occurrences classified as faulty software that were accurately forecasted by the model. Equation 2 represents the proportion of problematic software instances accurately detected by the model.

$Sensitivity = \frac{TP}{TP+FN}$ … (2)

Specificity

Specificity, known as the genuine negative rate, holds relevance within the software defect domain. Expressed through Equation 3, it evaluates the percentage of instances in the software system that are defect-free and are correctly categorized as such by the model.

$$Specifity = \frac{TN}{TN+FP} \quad \ldots (3)$$

Detection Rate

The detection rate refers to the proportion of the entire sample in which events were accurately identified. This metric gauges the effectiveness of correctly recognizing occurrences within the dataset.

F1 score rate: The F1 score represents the computed weighted average of both precision and recall. As such, this score takes into account the balance between false positives and false negatives.

Precision: Precision is a metric that measures the accuracy of positive predictions made by a model. It is defined as the ratio of correctly predicted positive samples to the total number of samples predicted as positive.

Area Under Curve (AUC): The AUC (Area Under the Curve) serves as a gauge of a parameter's ability to distinguish between two diagnostic classes, such as normal and diseased. Ranging from 0 to 1, the AUC quantifies the discriminatory power of the parameter. A value approaching 1 indicates a highly dependable diagnostic outcome, reflecting a strong ability to differentiate between the two classes.

RESULTS AND DISCUSSION

Data preprocessing

Importing the Libraries

The Jupyter Notebook was set up with the necessary Python libraries, including Numpy, Pandas, Matplotlib, and Seaborn. Numpy is a powerful library that allows for efficient processing and broadcasting of n-dimensional arrays (Stanin & Jovi, 2019). Panda is an open-source tool for data analysis and manipulation built on the Python programming language (Subasi, 2020). Matplotlib and Seaborn are popular packages used for data visualization. The platform provides a user-friendly interface that makes it easy to create visually appealing and informative graphs. Seaborn, a data visualization library, is an extension of Matplotlib, offering a slightly reduced set of functionalities (Pintor et al., 2019). Figure 2 depicts the process of importing Numpy, Pandas, Matplotlib, and Seaborn Python libraries into the Jupyter Notebook.

```
In [1]: # import libraries
        import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        import seaborn as sns
```

Figure 2: Importing Python Libraries

Loading the Dataset

The dataset was imported into the Jupyter Notebook environment using the pd.read_csv function. Figure 3 depicts the process of importing the dataset into Jupyter Notebook.

```
In [2]: # Load Breast Cancer Dataset
        df = pd.read_csv("data1.csv")
```

Figure 3: Loading the Dataset into Jupyter Notebook
Source: Authors

The Shape of the Dataset

The shape() function is employed to get and display the dimensions of a dataset, namely the count of rows and columns. Figure 4 depicts the presence of 3414 rows and 32 columns.

```
In [3]: df.shape

Out[3]: (3414, 32)
```

Figure 4: Number of Rows and Columns in the Dataset
Source: Authors

Check for Duplicates in the Dataset

The duplicate () function in a data frame returns a series of true and false values indicating which rows are duplicates. Table 1 shows that there are no duplicates in the dataset.

Table 1: Number of Duplicates

```
In [4]: df.duplicated()

Out[4]: 0       False
        1       False
        2       False
        3       False
        4       False
                ...
        3409    False
        3410    False
        3411    False
        3412    False
        3413    False
        Length: 3414, dtype: bool
```

The Info of the Dataset

The info() function is utilized to present the number of columns, their respective labels, data kinds, and the count of non-null cells within each column. According to the data presented in Table 2, there is a lack of empty values within the dataset. However, it is necessary to convert the diagnostic column from a string format to numerical values to provide more effective analysis utilizing machine learning methodologies.

Table 2: Info of the Dataset

```
In [5]: df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3414 entries, 0 to 3413
Data columns (total 32 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   id                       3414 non-null   int64
 1   diagnosis                3414 non-null   object
 2   radius_mean              3414 non-null   float64
 3   texture_mean             3414 non-null   float64
 4   perimeter_mean           3414 non-null   float64
 5   area_mean                3414 non-null   float64
 6   smoothness_mean          3414 non-null   float64
 7   compactness_mean         3414 non-null   float64
 8   concavity_mean           3414 non-null   float64
 9   concave points_mean      3414 non-null   float64
 10  symmetry_mean            3414 non-null   float64
 11  fractal_dimension_mean   3414 non-null   float64
 12  radius_se                3414 non-null   float64
 13  texture_se               3414 non-null   float64
 14  perimeter_se             3414 non-null   float64
 15  area_se                  3414 non-null   float64
 16  smoothness_se            3414 non-null   float64
 17  compactness_se           3414 non-null   float64
 18  concavity_se             3414 non-null   float64
 19  concave points_se        3414 non-null   float64
 20  symmetry_se              3414 non-null   float64
 21  fractal_dimension_se     3414 non-null   float64
 22  radius_worst             3414 non-null   float64
 23  texture_worst            3414 non-null   float64
 24  perimeter_worst          3414 non-null   float64
 25  area_worst               3414 non-null   float64
 26  smoothness_worst         3414 non-null   float64
 27  compactness_worst        3414 non-null   float64
 28  concavity_worst          3414 non-null   float64
 29  concave points_worst     3414 non-null   float64
 30  symmetry_worst           3414 non-null   float64
 31  fractal_dimension_worst  3414 non-null   float64
dtypes: float64(30), int64(1), object(1)
```

Data Count

Data count indicates the number of benign (B) and malignant (M) instances. In the dataset, there are 2142 benign and 1272 malignant cases, shown in Figure 5.

```
In [8]: #Get the count of malignant<1> and Benign<0> cells
        df['diagnosis'].value_counts()

Out[8]: B    2142
        M    1272
        Name: diagnosis, dtype: int64
```

Figure 5: Data Count

Data Visualization

Data Visualization is the representation of the data count using histogram and pie chart. Figure 6 shows 2142 instances of benign and 1272 instances of malignant using histogram which is equivalent to 62.7% and 37.3% respectively as can be seen on the pie chart.

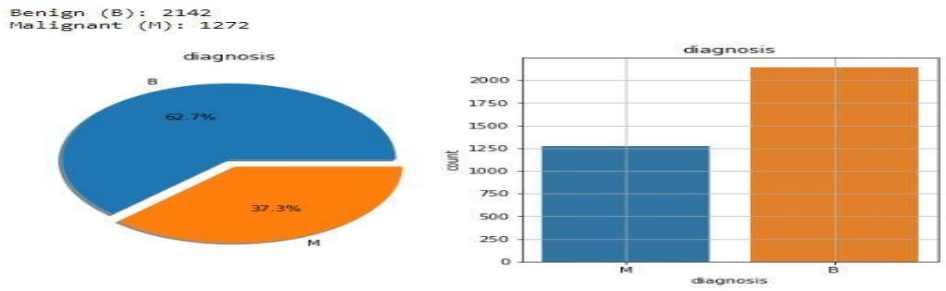

Figure 6: Data Visualization

RESULTS

The results extracted from the findings of the data analysis are presented here. The results are presented using figures and tables.

Data Encoding

In the dataset, the diagnosis column consists of strings of characters that represent either malignant (M) or benign (B) conditions. To convert this feature into numerical values, M was replaced with 1 and B was replaced with 0. The encoded diagnosis feature can be seen in Table 3, where M's and B's are shown as 1's and 0's respectively.

Table 3: Data Encoding

```
In [10]: # Label encoding (convert the value of M and B into 1 and 0)
         from sklearn.preprocessing import LabelEncoder
         labelEncoder_Y = LabelEncoder()
         df.iloc[:,1]=labelEncoder_Y.fit_transform(df.iloc[:,1].values)

In [11]: df.head()
Out[11]:
```

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean
0	842302	1	17.99	10.38	122.80	1001.0	0.11840
1	842517	1	20.57	17.77	132.90	1326.0	0.08474
2	84300903	1	19.69	21.25	130.00	1203.0	0.10960
3	84348301	1	11.42	20.38	77.58	386.1	0.14250
4	84358402	1	20.29	14.34	135.10	1297.0	0.10030

5 rows × 32 columns

```
In [12]: df.tail()
Out[12]:
```

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean
3409	926424	1	21.56	22.39	142.00	1480.0	0.11100
3410	926682	1	20.13	28.25	131.20	1262.0	0.09780
3411	926954	1	16.60	28.08	108.30	858.1	0.08455
3412	927241	1	20.60	29.33	140.10	1265.0	0.11780
3413	92751	0	7.76	24.54	47.92	181.0	0.05263

5 rows × 32 columns

Getting the Correlation

To ascertain the features that exhibit a negative association with the diagnosis, an analysis was conducted to examine the correlation between each feature. The analysis of Table 4 reveals that the variables fractal_dimension_mean, texture_se, smoothness_se, and symmetry_se exhibit a negative connection with the diagnosis. Consequently, these characteristics will be excluded from the dataset.

Table 4: Getting the Correlation

```
In [13]: #Get correlation of the columns
         df.iloc[:,1:32].corr()
Out[13]:
```

	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean
diagnosis	1.000000	0.719760	0.413564	0.742560	0.708986
radius_mean	0.719760	1.000000	0.318575	0.983519	0.973364
texture_mean	0.413564	0.318575	1.000000	0.327990	0.319794
perimeter_mean	0.742560	0.983519	0.327990	1.000000	0.986386
area_mean	0.708986	0.973364	0.319794	0.986386	1.000000
smoothness_mean	0.358560	0.166638	-0.023328	0.207230	0.177026
compactness_mean	0.596534	0.498083	0.235784	0.556857	0.498501
concavity_mean	0.696360	0.666850	0.301241	0.716055	0.685982
concave points_mean	0.776614	0.810231	0.292313	0.850869	0.823268
symmetry_mean	0.330499	0.144819	0.071105	0.182985	0.151293
fractal_dimension_mean	-0.012838	-0.307991	-0.076141	-0.261458	-0.283111
radius_se	0.567134	0.670167	0.274749	0.691681	0.732561
texture_se	-0.008303	-0.095632	0.384843	-0.086764	-0.066284
perimeter_se	0.556141	0.685309	0.280578	0.693048	0.726628
area_se	0.548236	0.726239	0.258791	0.744890	0.800085
smoothness_se	-0.067016	-0.219308	0.006566	-0.202685	-0.166782
compactness_se	0.292999	0.203305	0.191207	0.250708	0.212581
concavity_se	0.253730	0.191658	0.142716	0.228058	0.207658
concave points_se	0.408042	0.370479	0.163174	0.407151	0.372318
symmetry_se	-0.006522	-0.102476	0.009066	-0.081643	-0.072497
fractal_dimension_se	0.077972	-0.041770	0.054238	-0.005535	-0.019888
radius_worst	0.776454	0.955535	0.351181	0.969370	0.962746
texture_worst	0.456903	0.292609	0.908619	0.303053	0.287489
perimeter_worst	0.782914	0.951227	0.356641	0.970278	0.959120
area_worst	0.733825	0.927668	0.342191	0.941444	0.959213
smoothness_worst	0.421465	0.116413	0.077197	0.150531	0.123520
compactness_worst	0.590998	0.406736	0.276764	0.455733	0.390411
concavity_worst	0.659610	0.519196	0.299875	0.563828	0.512606
concave points_worst	0.793566	0.732601	0.294167	0.771159	0.722017
symmetry_worst	0.416294	0.160196	0.104609	0.189090	0.143572
fractal_dimension_worst	0.323872	0.006707	0.118768	0.051016	0.003738

31 rows × 31 columns

Dropping Features with Negative Correlation

To enhance the classification accuracy of the model, it was concluded that specific modifications should be implemented. In particular, four specific qualities, namely fractal_dimension_mean, texture_se, smoothness_se, and symmetry_se, were selected for exclusion. The aforementioned traits demonstrated a negative correlation with the diagnosis, as seen in Figure 7. As a result, the number of columns in the model has been reduced to 28.

```
In [14]: df.drop(columns=['fractal_dimension_mean','texture_se','smoothness_se','symmetry_se'], axis=1, inplace=True)

In [15]: df.shape

Out[15]: (3414, 28)
```

Figure 7: Dropping Features with Negative Correlation

Splitting the Dataset into two

The dataset has been partitioned into three distinct subsets, namely the test set, validation set, and training set. The data partitioning into these sets is seen in Figure 8. The training set comprises 60% of the dataset, and the validation and test set each encompass 20% of the dataset.

```
In [16]: # split the dataset into dependent(X) and Independent(Y) database
         x=df.iloc[:,2:28].values
         y=df.iloc[:,1].values
```

Figure 8: Splitting the dataset into x and y

Train, Validation and Test Split

The data has been divided into three sets - test set, validation set, and training set. Figure 9 illustrates the split of data into these sets. The training set contains 60% of the data, while the validation and test sets contain 20% of the data each.

```
In [17]: # splitting the data into training, validation and test dataset
         from sklearn.model_selection import train_test_split
         x_main,x_test,y_main,y_test=train_test_split(x,y,test_size=0.20,random_state=101)
         x_train,x_val,y_train,y_val=train_test_split(x_main,y_main,test_size=0.20,random_state=101)
```

Figure 9: Train, Validation, and Test Split

Feature Scaling

Feature scaling was applied using Standard Scaler to standardize the dataset to optimize the performance of the models. Figure 10 shows the scaling and standardizing of the training, validation, and test data using a standard scaler.

```
In [21]: # feature scaling
         from sklearn.preprocessing import StandardScaler
         x_train=StandardScaler().fit_transform(x_train)
         x_test=StandardScaler().fit_transform(x_test)
         x_val=StandardScaler().fit_transform(x_val)
```

Figure 10: Feature scaling

Modeling with the Selected Algorithm

The process involves the integration of the dataset into the algorithms to facilitate the training, validation, and testing of the models. Figure 11 illustrates the utilization of Random Forest, Support Vector Machine, K-Nearest Neighbour, and Logistic Regression Algorithms to fit the training and validation data for training and validation purposes.

```
In [22]: from sklearn.ensemble import RandomForestClassifier
         rf=RandomForestClassifier(random_state=0,criterion="entropy",n_estimators=10)
         rf.fit(x_train,y_train)
         rf.fit(x_val,y_val)
Out[22]: RandomForestClassifier(criterion='entropy', n_estimators=10, random_state=0)

In [23]: from sklearn.svm import SVC
         sv=SVC()
         sv.fit(x_train, y_train)
         sv.fit(x_val, y_val)
Out[23]: SVC()

In [24]: from sklearn.neighbors import KNeighborsClassifier
         knn=KNeighborsClassifier()
         knn.fit(x_train, y_train)
         knn.fit(x_val, y_val)
Out[24]: KNeighborsClassifier()

In [25]: from sklearn.linear_model import LogisticRegression
         lr=LogisticRegression(solver='liblinear', multi_class='ovr')
         lr.fit(x_train,y_train)
         lr.fit(x_val,y_val)
Out[25]: LogisticRegression(multi_class='ovr', solver='liblinear')
```

Figure 11: Modeling with the Selected Algorithm

Train Score Accuracy Evaluation

This is the train score accuracy of the models. Figure 12 shows the train score accuracy of the models. The accuracy score of Random Forest = 96.42%, Support Vector Machine = 97.84%, K-Nearest Neighbour = 96.74% and Logistic Regression = 98.76%.

```
In [26]: from sklearn.metrics import accuracy_score
         print("Train Accuracy of Random Forest", rf.score(x_train,y_train)*100)
         Train Accuracy of Random Forest 96.42857142857143

In [27]: print("Train Accuracy of Support Vector Machine", sv.score(x_train,y_train)*100)
         Train Accuracy of Support Vector Machine 97.84798534798534

In [28]: print("Train Accuracy of K-Nearest Neigbour", knn.score(x_train,y_train)*100)
         Train Accuracy of K-Nearest Neigbour 96.74908424908425

In [29]: print("Train Accuracy of Logistic Regression", lr.score(x_train,y_train)*100)
         Train Accuracy of Logistic Regression 98.76373626373626
```

Figure 12: Train Score Accuracy Evaluation

DISCUSSION

Table 5 presents accuracy measures, namely the False Positive Rate (FPR) and False Negative Rate (FNR), for several machine learning models used in the prediction of breast cancer. Among the models under evaluation, it is worth noting that the logistic regression model has a remarkable test accuracy rate of 99.26%. The level of accuracy shown by the model surpasses that of all other models and is especially remarkable within the unique setting of the research. The improved efficacy of the logistic regression model may be ascribed to its systematic approach to selecting features. BeforeFactors that showed a negative correlation with the diagnostic (output) were carefully excluded prior to data analysis. The technique that was

previously mentioned has made a substantial contribution to the model's extraordinary level of accuracy. Furthermore, it is crucial to emphasize that the logistic regression model exhibits a minimum False Positive Rate (FPR) of 1 and a minimum False Negative Rate (FNR) of 4. This highlights the efficacy of the method in accurately detecting instances of breast cancer. The indicated degree of precision is particularly noteworthy when compared with the results reported in the previously examined academic literature. The work incorporates references to many significant studies conducted by Shubham and Kamalraj (2022), Tiwari et al. (2020), Khan et al. (2022), Obaid et al. (2018), and Rufai et al. (2020).

Table 5: Accuracy Table of the Models.

Machine Learning Models	Test Score of the Models	False Positive Rate (FPR)	False Negative Rate (FNR)
Random Forest	96.63%	7	16
Support Vector Machine	98.24%	2	10
K-Nearest Neighbor	96.92 %	3	18
Logistic Regression	99.26%	1	4

CONCLUSION

The Logistic Regression model had a high level of accuracy, with a rate of 99.26%. It also revealed a low False Positive Rate (FPR) of 1% and a False Negative Rate (FNR) of 4%. Based on a thorough examination of three different models and an exhaustive review of relevant literature, it has been concluded that Logistic Regression is the most suitable choice for the early detection of breast cancer. Hence, a compelling argument can be made in favor of Logistic Regression as the optimal model for the early detection of breast cancer.

Competing interests: The authors declare that they have no conflict of interest.

REFERENCES

Chaurasia, V., & Pal, S. (2014). A Novel Approach for Breast Cancer Detection Using Data Mining Techniques. *International Journal of Innovative Research in Computer and Communication Engineering, 2(1), 2456–2465.*

Fatima, N., Liu, L., Hong, S., & Ahmed, H. (2020). Prediction of Breast Cancer, Comparative Review of Machine Learning Techniques, and Their Analysis. *IEEE Access*, *8*, 150360–150376.

Gupta, P., & Garg, S. (2020). Breast Cancer Prediction Using Varying Parameters of Machine Learning Models. *Procedia Computer Science*, *171*, 593–601.

Islam, M., Haque, R., Iqbal, H., Hasan, M., Hasan, M., & Kabir, M. N. (2020). Breast Cancer Prediction: A Comparative Study Using Machine Learning Techniques. *SN Computer Science*, *1*(5), 1–14.

Mahmood, T., Li, J., Pei, Y., Akhtar, F., Imran, A., & Rehman, U. K. (2020). A Brief Survey on Breast Cancer Diagnostic With Deep Learning Schemes Using Multi-Image Modalities. *IEEE Access*, *8*, 165779–165809.

Monirujjaman Khan, M., Islam, S., Sarkar, S., Ayaz, F. I., Kabir, M., Tazin, T., Albraikan, A. A., & Almalki, F. A. (2022). Machine Learning Based Comparative Analysis for Breast Cancer Prediction. *Journal of Healthcare Engineering*, *2022*.

Naji, M. A., El Filali, S., Aarika, K., Benlarmar, E. H., Abdelouhahid, R. A., & Debauche, O. (2021). Machine Learning Algorithms For Breast Cancer Prediction And Diagnosis. *Procedia Computer Science*, *191*, 487–492.

Obaid, O. I., Mohammed, M. A., Khanapi, M., Ghani, A., Mostafa, S. A., & Taha, F. (2018). Evaluating the Performance of Machine Learning Techniques in the Classification of

Wisconsin Breast Cancer. *International Journal of Engineering & Technology*, *7*(4.36), 160–166.

Pintor, M., Demetrio, L., Sotgiu, A., Melis, M., Demontis, A., & Biggio, B. (2019). *Cecil: A Python Library for Secure and Explainable Machine Learning*. *arXiv preprint arXiv:1912.10013*.

Rana, M., Chandorkar, P., Dsouza, A., & Kazi, N. (2015). Breast Cancer Diagnosis and Recurrence Prediction Using Machine Learning Techniques. *International Journal of Research in Engineering and Technology*, *4*(4), 372–376.

Rawal, R. (2020). Breast Cancer Prediction Using Machine Learning. *Journal of Emerging Technologies and Innovative Research (JETIR)*, *7*(5), 13–24.

Rufai, M. A., Muhammad, A. S., Garba, S., & Audu, L. (2020). Machine Learning Model for Breast Cancer Detection. *FUDMA Journal of Sciences (FJS)*, *4*(1), 55–61.

Shravya, C., Pravalika, K., & Subhani, S. (2019). Prediction of Breast Cancer Using Supervised Machine Learning Techniques. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, *8*(6), 1106–1110.

Shubham, K., & Kamalraj, R. (2022). Breast Cancer Detection Using Machine Learning Algorithms. *International Journal of Advances in Engineering and Management (IJAEM)*, *4*(3), 987–994.

Singh, G. (2020). Breast Cancer Prediction Using Machine Learning. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, *8*(4),

Subasi, A. (2020). *Practical Machine Learning for Data Analysis*. Academic press.

Tiwari, M., Lokare, R., Shah, P., & Bharuka, R. (2020). Breast Cancer Prediction Using Deep Learning and Machine Learning Techniques. *SSRN 3558786*.

Yee, W. S., Ng, H., Tzen, T., Yap, V., Goh, V. T., Ng, K. H., & Cher, D. T. (2022). An Evaluation Study on the Predictive Models of Breast Cancer Risk Factor Classification. *Journal of Logistics, Informatics and Service Science*, *9*(3), 129–145.