USING AN ADAPTIVE LINEAR SUPPORT VECTOR MACHINE ALGORITHM FOR PREDICTING BREAST CANCER

A.S.K. AL-Hurdi [1,*]
A.M.A. Mohsen [1]

Received: 03/01/2025
Revised: 10/01/2025
Accepted: 18/01/2025

[1] Computer Information Systems, Arab Academy for Management, Banking, and Financial Science (AAMBFS), Aden, Yemen
* Corresponding Author's Email: aalhurdi@gmail.com

Using an Adaptive Linear Support Vector Machine Algorithm for Predicting Breast Cancer

AL-Hurdi A.S.K
College of Management, Banking, and Financial Science.Department: Computer Information Systems, Arab Academy for Management, Banking, and Financial Science (AAMBFS), Aden, Yemen
aalhurdi@gmail.com

Mohsen A.M.A
College of Management, Banking, and Financial Science.Department: Computer Information Systems, Arab Academy for Management, Banking, and Financial Science (AAMBFS), Aden, Yemen

Abstract— Breast cancer is the most common type of cancer and a significant contributor to the high death rates among women. The death rate increases when this condition is manually diagnosed since it takes several hours and specialists. Therefore, an automated breast cancer diagnosis has been suggested to speed up detection and stop the disease from spreading. Over the years, machine learning classification algorithms have been used to predict breast cancer. In the previous studies, one of the most used algorithms is the Support Vector Machine (SVM). However, these studies have inconsistent results. This work investigates the impact of the features' selection, hyperparameter parameters of SVM, and the mechanism of splitting data on the algorithm's performance. Thus, build an SVM, as a single machine learning model, that achieves a higher result. The Wisconsin dataset was used to train and test this model. The experimental results showed that the performance of the model was affected by the features' selection, hyperparameter parameters, and the mechanism of splitting data and random state values in terms of the best top one results and the average of the top three results. The comparison results revealed the superiority of the proposed method over the other state-of-the-art.

Keywords—Support vector machine, Wisconsin Breast Cancer Original Dataset, Machine learning, Accuracy, Random state.

I. INTRODUCTION

One type of cancer that occurs in the breast is breast cancer. It may begin in one breast or both. Anytime cells start to grow uncontrolled, cancer develops. An estimated 5% of all breast cancer patients are under 40. These individuals frequently receive alternatives since their illnesses are viewed as being more aggressive. The American College of Surgeons' Cancer database was examined from 1998 to 2005 for all breast cancer patients. Patients under 40 made up the study cohort [1]. Throughout their lifetimes, 12% of women in the United States have been diagnosed with breast cancer. In 2017, more than 250,000 new disease cases were discovered [2]. In the United States, breast cancer was the second-leading cause of cancer-related fatalities in women in 1993, accounting for around 46,300 deaths and posing serious public health concerns [3]. Various factors can impact the chances of developing breast cancer, and there is not just one factor, including DNA and surroundings, that contribute to it [4]. Business intelligence is the process of using information and analyzing it to support decision-making and using different methods to help organizations forecast the behavior of competitors, suppliers, customers, and environments to stay alive and survive in the global economy. Also, using tools such as data mining and data warehousing helps to make decisions and achieve a competitive advantage, and coordinating personal and organizational goals leads to creating synergy and improved performance. Early-stage diagnosis of breast cancer can help significantly increase patients' survival rates. Data mining techniques can be vital in the early-stage diagnosis of breast cancer [5]. Data mining techniques classification is the most widely used. It uses a set of pre-classified categories using Bayesian classification, decision tree induction, neural networks, and SVM [6]. A rising number of specialists are turning to SVM because of its excellent diagnostic abilities and useful categorization. The development of technologies that enable the early identification of breast cancer can be assisted by practitioners using machine learning techniques. SVM is a supervised machine learning algorithm employed for regression and classification [7].

II. LITERATURE REVIEW

A rough set-based feature selection for the prediction of breast cancer [8]. In this work, the researcher used a set of algorithms based on feature selection to predict breast cancer. It divided the data into training and testing data. The used algorithms are DT, KNN, BN, SVM, LR, RF, and Adaboost. The RF algorithm got a higher accuracy of 95.23%. [9] applied a method in an LDA–SVM machine learning model to breast cancer classification. After collecting and processing data, the researcher reduced, separated, trained, and tested the data using the LDA. The models used different techniques, including PCA-SVM, LDA-SVM, RF-LDA, and RF-PCA. The performance of the LDA-SVM model got higher results: 99.20% [10]. Used exploring machine learning algorithms to find the best features for predicting breast cancer and its recurrence. First, the researcher began to bring data on breast cancer patients from the site of the UCI WBC original dataset [11]. The researcher used multiple algorithms: RF, SVM, KNN, and BN, and split the data into two parts, 80-20, where 80 represents the training of the model and 20 percent is a test of the model. After the trial, the KNN algorithm got a higher accuracy of 95.90%. [12] demonstrated a breast cancer diagnosis and a survival prediction using JNN. In this

research, the JNN tool was used to analyze the data. The researcher used the ANN algorithm for a breast cancer diagnosis. However, this model has not achieved higher accuracy, 88.24%, and the results are not promising compared to previous studies [13]. Showed an artificial neural network method as an ensemble technique fused for improving classification accuracy. The researcher split the data into two parts: training and testing data. The researcher entered the data on a set of algorithms (ensemble algorithm) simultaneously: SVM, RF, and ANN. After scoring, the novel model got a higher accuracy of 98.50% [14]. Showed a system breast cancer prediction using the KE Sieve algorithm. In this study, the researcher divided the data for training and testing, as the data divider took several partitions. They are as follows: 10–90%, 20–80%, 30–70%, and 40–60%. The researcher used the KE-sieve algorithm. Where the values of K follow 1, 2, and 3, the KNN algorithm got a higher accuracy of 96.35% (when k=3, splitting data 90-10%). [15] Applied a novel intelligent classification model for a breast cancer diagnosis. This study used the Matlab tool the researcher. The researcher designed a new model: the work of a set of algorithms working together (ensemble algorithm) so that the data set is input into them, and then the outputs. These algorithms are KNN, CSSVM, and BP. The KNN=3 and the BP algorithm got a higher accuracy of 97.50%. [16] Illustrated a performance comparison of three classifiers for the classification of the breast cancer dataset. The researcher used three types of algorithms to compare three classifiers, and these algorithms, namely SVM, BN, and ANN. The SVM-linear kernel achieved a higher accuracy of 96.72%. [17] Demonstrated a method of prediction of benign and malignant breast cancer using data mining techniques. In this research, the researcher used J48, NB, and RPF networks in this model after testing the model and knowing that the tool used is Weka. The NB algorithm attained higher results: 97.30% [18]. Demonstrated a breast cancer prediction system, where they analyzed the data using the KNN algorithm. Then the KNN improved the accuracy of the classifier, which achieved a higher accuracy of 98.60% [19]. Illustrated an efficient breast cancer diagnosis and a survival prediction using the L-Perceptron. Research talks about the effectiveness of diagnosing breast cancer and predicting survival using the algorithms used for this purpose: L-perceptron, RPF network, BN, and J48. The researcher divided the data into specific proportions for training and testing, using Python tools to analyze the data. The L-perceptron algorithm obtained a higher accuracy of 97.42% [20]. Showed a method using feature selection techniques to improve the accuracy of breast cancer classification. In this research, the researcher started to make a technique for selecting features to enhance the performance of algorithms. The researcher used multiple models, which are as follows: NB, SVM, KNN, DT, LR, and ANN. The NB algorithm achieved a higher accuracy of 97.42% when researchers selected different features for every model. [21] Applied a performance evaluation method of machine learning methods to breast cancer prediction. The researcher divided the data into two parts for training and testing. In this study, the researcher used R programming to analyze the data and a set of algorithms to determine the performance and evaluate various algorithms to predict breast cancer. These algorithms are Decision Tree, Support Vector Machine, RF, LR, and NN. The Decision Tree and RF algorithm achieved better results at 96.1%. [22] Illustrated a performance analysis of a breast cancer classification using a decision tree classifier. In this work, the researcher uses the Weka tool. This environment contains all the supervised and non-supervised algorithms and divides the data into training and testing. The researcher used the algorithm of the decision tree for a breast cancer classification, where it is displayed as follows: REP Tree, Priority-based, J48, RF and RT are the RF algorithms that achieved a higher accuracy of 96.70% [23]. Applied an investigation of the effect of a correlation-based feature selection on a breast cancer diagnosis using artificial neural networks and support vector machines. In this research, the researcher set out to clarify the impact of the correlation of feature selection for predicting and diagnosing breast cancer, which the researcher used as a weak tool for the analysis process—he also divided the data into two parts for training and testing. Where the researcher used an algorithm, ANN, and SVM. In this research, the researcher selected 10, 5, 3, 2, and 1 features on the ANN and SVM algorithms. The SVM algorithm achieved a higher accuracy of 97.13% with the selected ten features [24]. Used a method performance comparison of machine learning techniques for breast cancer detection. In this work, the researcher compares different models of breast cancer detection using the Weka tool. The models used are as follows: RPF, SVM, SL, BN, KNN, AdaBoost, FUZZY, and J48. The SVM algorithm achieved the best accuracy at 97.07% [25]. Demonstrated a classification of breast cancer with improved self-organizing maps. In this research, the researcher used a set of algorithms to classify breast cancer: perceptron multilayer SVM, LR, SOM, and DSOM. The researcher divided the data into two parts for training and testing, 70-30%. The DSOM algorithm improves results and accuracy by 98.56%. [26] demonstrated a method using machine learning algorithms for breast cancer risk prediction and diagnosis. In this research, the researcher used a Weka tool to import a database from the site of UCI (WBC original dataset). Then the researcher divided the data into training and testing, and the algorithms chosen by the researcher are as follows: C45, SVM, BN, and KNN. The support vector machine algorithm achieved a higher accuracy of 97.13%. [27] showed a comparative study of machine learning algorithms for breast cancer detection and diagnosis. In this research, the researcher's Weka tool was used. Then the researcher divided the data into training and testing and used the support vector machine, RF, and NB algorithms. The NB algorithms got higher results: 97.20% [27]. [28] used an analysis of the Wisconsin breast cancer dataset and machine learning for breast cancer detection, where the researcher divided the data into two parts for training and testing. The researcher entered this data on two algorithms, NB and J48. The researcher selected 10, 9, 8, 7, 6, 5, and 4 features on the J48 and NB algorithms. After score models, the NB algorithm got better at 97.14% [28]. Al Islam and Imtiaz showed the most effective machine-learning algorithms for detecting breast cancer. The researcher entered the data after processing it into the Weka tool and then used four algorithms for analysis: DT, SVM, RF, and KNN, which are as follows after

testing the algorithm. The Decision Tree algorithm, the RF algorithm, is the best at 96.90%.

III. METHODOLOGY

This chapter presents the SVM-led mechanism. Presents the architecture of the proposed model. The SVM implementation mechanism is described. The algorithm Performance Investigation Flow Chart is presented. Dataset Collection is presented to demonstrate the data type of features. The data preprocessing is described. Demonstrates the appropriate hyperparameter. The proposed support vector machine model is presented. Present the evaluation. Presents the framework for predicting breast cancer in hospitals. Finally, a summary is presented.

A. Support Vector Machine Performance Mechanism (Flow Chart)

Figure 1 demonstrates the flowchart of the modeling process of this proposed model. The historical data, Preprocessing of Historical data, and loading of historical data are illustrated in the first and second steps. The third step is to define the initial SVM parameters and the kernel function that will be used from the proposed model. Then, the training process is demonstrated in the fourth step. The historical data are split into v parts. One subset is used as a validation part and the remaining is used to train the model in the fifth step. Then the trained model and unseen data are explained in the sixth step. A proposed model of the forecasting process is described in the seventh step.

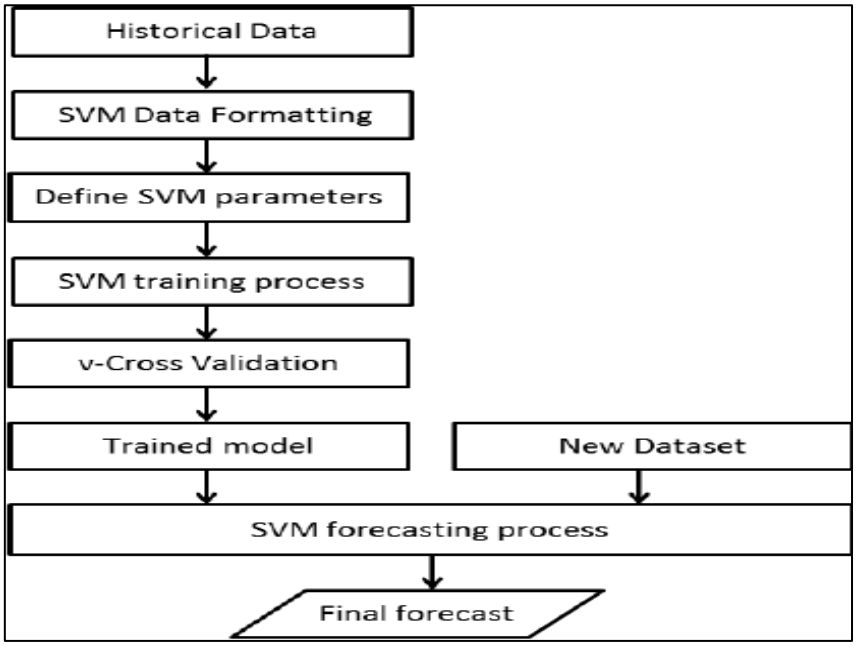

Fig. 1. SVM-led Mechanism

B. The Architecture of the Proposed Model

The general architecture of the proposed model consists of four phases, which are illustrated in Figure 2. These phases are data collection, data processing, hyperparameters, the proposed model (SVM), and evaluation of the applied model.

C. SVM Implement Mechanism

The SVM Implement Mechanism is based on various elements such as:

1) Import the libraries: the main libraries include Scikit-learn, Pandas, NumPy 1.17.4, and Matplotlib.
2) Upload the dataset: load the breast cancer dataset from the UCI machine learning repository.
3) Split the dataset into X and Y: X represents the breast cancer features (Sample code number, Clump Thickness, Uniformity of Cell Size, Uniformity of Cell Shape, Marginal Adhesion, Single Epithelial Cell Size, Bare Nuclei, Bland Chromatin, Normal Nucleoli, Mitoses, and Y represents a class label "class").
4) Split the X and Y dataset into the training set and test set: in this case using k-fold cross-validation to train and test k=4,8, and the given X 80%, 90%, 70%, and 60%.
5) Perform normalizations for features: in this case, transfer numbers over two between 0 and 1.
6) Fit the proposed model to the training set.
7) Predict the test set results: this case determines the performance of the model.
8) Make the confusion matrix: this case depended on TP, FP, FN, and TN.
9) Visualization of the test set results: in this research using an Excel tool to visualize the results.

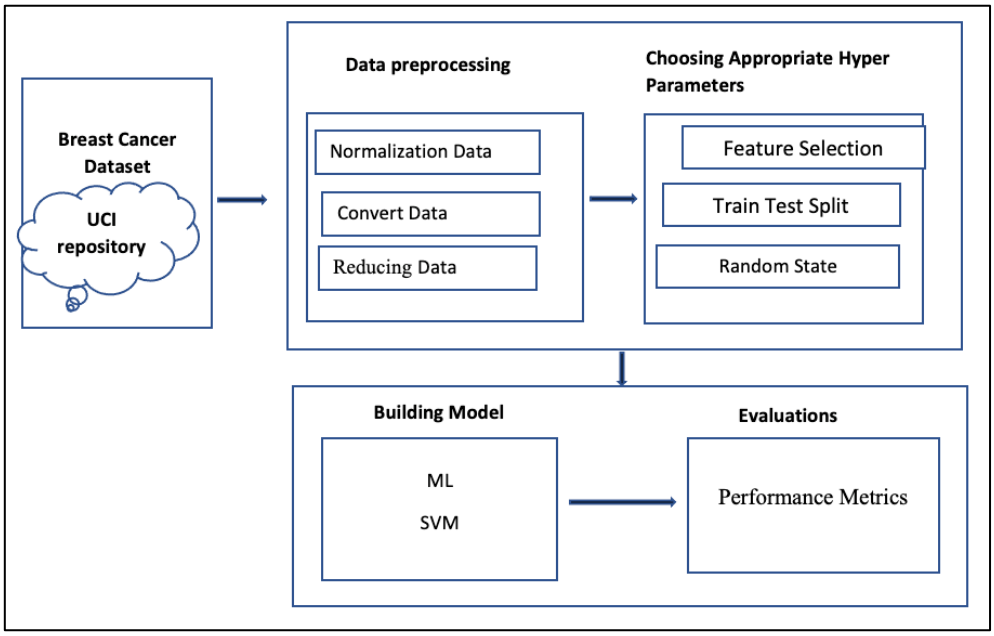

Fig. 2. Proposed Model for Predicting the Breast Cancer

D. Algorithm Performance Investigation Flow Chart

Figure 3 demonstrates the algorithm performance investigation flowchart including six steps, optimal dataset representation of the breast cancer dataset present in step One, features of a selection of nine features, eight features, seven features, five features present in step Two, split data 90-10, 80-20, 70-30, 60-40 present in step Three, the random state values (1-10) are presented in Step Four, record results based on the top one results present in step Five, and finally step Six records the results based on the average of the top three results. Will use these steps in chapter four.

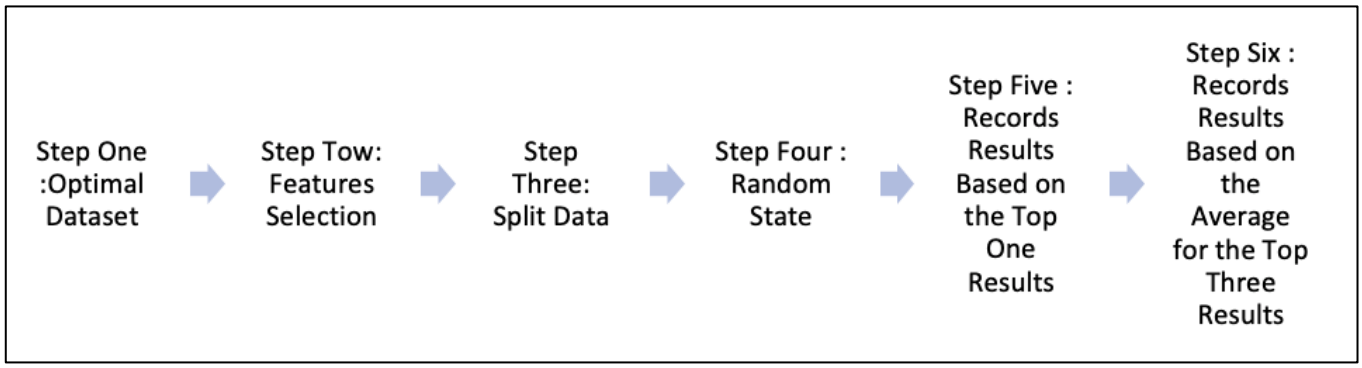

Fig. 3. Algorithm Performance Investigation Process

E. Dataset Collection

The UCI machine learning repository's Wisconsin breast cancer (original) datasets contain 699 incidences of breast cancer in Wisconsin (Benign, 458; Malignant, 241) and contain two classes (65.5% Malignant; 34.5% Benign) and 11 integer-valued features [11]. The WBC Original dataset features are as follows: (Sample code number, Clump Thickness, Uniformity of Cell Size, Uniformity of Cell Shape, Marginal Adhesion, Single Epithelial Cell Size, Bare Nuclei, Bland Chromatin, Normal Nucleoli, and Mitoses) as shown in Table 1.

Table 1. The Breast Cancer Dataset Features

Attribute	**Values**
Sample code number	1-10
Clump Thickness	1-10
Uniformity of Cell Size	1-10
Uniformity of Cell Shape	1-10
Marginal Adhesion	1-10
Single Epithelial Cell Size	1-10
Bare Nuclei	1-10
Bland Chromatin	1-10
Normal Nucleoli	1-10
Mitoses	1-10
Class	2-4

F. Data Types of Features

Table 2. Data Type of Features

Id	integer
clump thic	integer
uniformity of size	integer
uniformity of cell shape	integer
marginal	integer
single ep cell size	integer
bar nuclei	integer
bland chromatin	integer
normal nucleoli	integer
mitoses	integer
class	integer

Table 2 illustrates data types of features, all features include values between 0-10, 0-5 which represents a negative value, and 6-10 representing positive values.

G. Data Preprocessing

The preprocessing is the second phase in the proposed model and consists of the following elements: normalization data, converting data, and reducing data.

H. Reducing Data

Table 3 shows all breast cancer data with ten features without the class label feature Table 4 demonstrates all breast cancer data without the ID number feature and class label feature. The objective of data reduction is the process of lowering the amount of necessary storage space.

Table 3. The Breast Cancer Data Before Reducing

ID_no	clump thic	uniformity of size	uniformity of cell shape	marginal	single ep cell size	bar nuclei	bland chromatin	normal nucleoli	mitoses
0	8	10	10	8	7	10	9	7	1
1	5	3	3	3	2	3	4	4	1

Table 4. Breast Cancer After Reducing Data

clump thic	uniformity of size	uniformity of cell shape	marginal	single ep cell size	bar nuclei	bland chromatin	normal nucleoli	mitoses
8	10	10	8	7	10	9	7	1
5	3	3	3	2	3	4	4	1

I. Normalization Data

Table 5 demonstrates breast cancer data before normalization. The features include different values between (1-10). However, Table 6 illustrates breast cancer data after normalization. The objective of normalizing, any redundant or unstructured data is removed to obtain a high-accuracy data output.

Table 5. Breast Cancer Data Before Normalization

clump thic	uniformity of size	uniformity of cell shape	marginal	single ep cell size	bar nuclei	bland chromatin	normal nucleoli	mitoses
8	10	10	8	7	10	9	7	1
5	3	3	3	2	3	4	4	1

Table 6. Breast Cancer Data After Normalization

clump thic	uniformity of size	uniformity of cell shape	marginal	single ep cell size	bar nuclei	bland chromatin	normal nucleoli	mitoses
0.19790469	0.70221201	0.74177362	0	0	0	0.5556085	0.69885309	0.18182716
0.19790469	0.27725185	0.26278299	0	0	0	1.69516613	1.77286724	0.18182716

J. Converting Data

The process of converting data from one format to another is known as data conversion.

Table 7 illustrates class label feature results before transformation, however, Table 8 shows class label feature results after transformation. Due to 2 transfer to 0 and 4 transfer to 1. This stage allows reading, updating, and using the data, as the updated data are suitable for analysis or for working with the algorithm.

Table 7. The class labels before the transfer

Class
2
4
2

Table 8. The class labels after the transfer

Class
0
1
0

K. Appropriate Hyper Parameter

The third case is considered in the proposed model and it consists of the following elements: train test split function, feature selection, and random state.

L. Feature Selection

This research used different feature selections for the algorithm performance investigations such as nine feature selections as shown in Table 9, eight feature selections as demonstrated in Table 10, seven feature selections as illustrated in Table 11, and finally, five feature selections as shown in Table 12.

Table 9. The Nine Features Selected for Investigation

Features name	**Random state**
clump thic	Serial
uniformity of size	1
Un of cell shape	2
marginal	3
single ep cell size	4
normal nucleoli	5
bland chromatin	6
bar nuclei	7
mitoses	8
	9
	10

Table 10. The Eight Features Selected for Investigation

Features name	**Random state**
clump thic	Serial
uniformity of size	1
Un of cell shape	2
marginal	3
single ep cell size	4
normal nucleoli	5
bland chromatin	6
mitoses	7
	8
	9
	10

Table 11. The Seven Features Selected for Investigation

Features name	**Random state**
clump thic	Serial
uniformity of size	1
Un of cell shape	2
marginal	3
single ep cell size	4
normal nucleoli	5
mitoses	6
	7
	8
	9
	10

Table 12. The Five Features Selected for Investigation

Features name	**Random state**
clump thic	Serial
uniformity of size	1
Un of cell shape	2
marginal	3
mitoses	4
	5
	6
	7
	8
	9
	10

M. Train Test Split Function
In this research, using various functions to execute train, and test for the algorithm namely the splitting data 80-20, 90-10, 70-30, and 60-40.

N. Random State
This variable is used to shuffle the data and then enter the data on the model for examination to ensure the performance of the model, and this operation continues until obtaining the best accuracy as shown in Table 9, Table 10, Table 11, and Table 12.

O. The Proposed Support Vector Machine Model
This research used the linear and the kernel support vector machine for separating the negative and positive patient data. The SVM mechanism depends on a set of factors, and it is represented by the data that have been processed as well as the hyperparameters setting, and it is represented by the random state of the data, feature selection (i.e., five features, seven features, eight features, and nine features), and finally, to the mechanism of splitting the data for training and testing into 80-20, 90-10, 70-30, and 60-40.

P. Evaluation
The evaluation of the proposed model performance is conducted using the confusion matrix. The confusion matrix consists of four basic measures: true positive, false positive, false negative, and true negative.

Q. Framework for Predicting Breast Cancer
Using the proposed model, a framework is designed to predict breast cancer in hospitals as depicted in Figure 4. Two assumptions for the framework should be made to perform the prediction. The first assumption is the preparation of the data, in which the preprocessing process should be applied to make the data suitable for the proposed model. The equality of the preprocessed data is the second assumption. When using the model for different patients, the data after the preprocessing process for each patient must be equally input into the model to avoid biased results. Based on the proposed model, the polarity of each breast cancer screening has to be predicted as positive or negative.

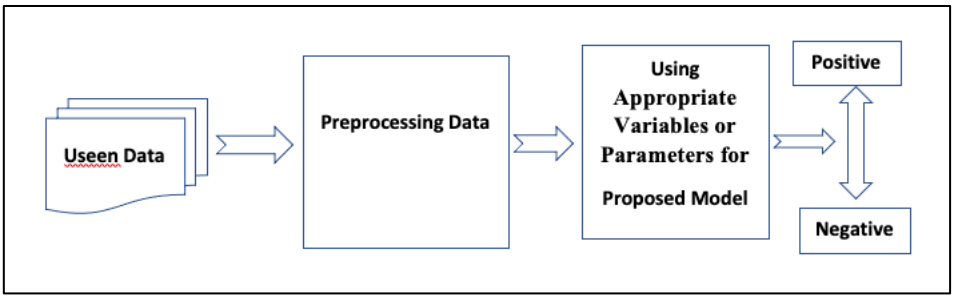

Fig. 4. Framework for Predicting Breast Cancer in the Hospitals

R. Summary
In the methodology section, the flow chart of the support vector machine model and the methodology of the proposed model are demonstrated, including serious phases: breast cancer dataset, pre-processing data, hyperparameters, and performance evaluation. Finally, illustrated a framework for predicting breast cancer in the medical field. The next sections illustrate the algorithm performance investigation, experimental environment, evaluation measures, and result.

IV. Experimental Environment

This section describes the experimental environment, tools used in experiments, measures of performance evaluation of classifiers, and ML classifiers. Applied experiments on a machine with properties that are Intel (R) Core (TM) i7-4702MQ CPU @ 2.20 GHz (8 CPUs), 8.00 GB RAM, 1.0 TB hard disk drive, and Windows 10 operating system. Using Python 3.7.0 to build a single algorithm model. Besides, deal with some libraries in Python for predictive data analysis. The main libraries include Scikit-learn, Pandas, NumPy 1.17.4, and Matplotlib. Microsoft Excel was used to organize and store datasets in tables, do some simple preprocessing, and analyze the results.

A. Libraries Used

1. NumPy: Numerical Python, sometimes known as NumPy, is a Python library created in 2000 for use in scientific computation. It is frequently used for processing multidimensional arrays. Additionally, it has several other characteristics, such as 1) broadcasting (advanced) operations and 2) it has tools that can also combine C/C++ and FORTRAN code.

2. Pandas: Pandas is a 2008-released open-source Python module that takes its name from panel data. Pandas work for data modification and analysis.

3. Scikit-Learn: It is a free and open-source library developed on top of SciPy, NumPy, and Matplotlib. Different classifications, clustering, regression, and other methods are helpful for data analysis and mining.

4. Matplotlib: With just a few lines of code, 2-D diagrams like bar charts, histograms, plots, scatter plots, etc., may be created using the Matplotlib charting tool.

B. Evaluation Measures

The confusion matrix is a performance measure frequently used in classification problems with two or more class-label outputs. Accuracy, precision, F1-score, and recall work using a confusion matrix.

1. Accuracy: The percentage of the correctly classified objects used to calculate the classifier's accuracy is calculating accuracy is explained by Equation (1) as follows:

$$=\left(\frac{True\ Positive+True\ Negative}{True\ Positive+True\ Negative+False\ Negative+False\ Positive}\right)(1)$$

where: TN and TP denote true negative and true positive, respectively. They are used to examine the correctness of the identified records as either positive or negative class. At the same time, FN and FP denote false negatives and false positives, respectively. They are used to test the incorrectness of the identified records for the opposite class.

2. Precision: Precision computed using Equation (2). Precision sometimes referred to as confidence, is the percentage of positive and natural negative occurrences that are unmistakably positive. Demonstrates the classifier's capacity to deal with favorable findings while minimizing commentary on adverse ones [29].

$$=\left(\frac{\text{True Positive}}{\text{True Positive+FalsePositive}}\right) \quad (2)$$

3. F1-score: The weighted harmonic mean of precision and recall is named the F1-Score which is explained by Equation (3) below. This score accounts for false positives and negatives.

$$\mathbf{F1-score} = \left(2 * \frac{\text{rcall} * \text{precision}}{\text{rcall+precision}}\right) \quad (3)$$

4. Recall: Recall, commonly referred to as sensitivity, is the frequency at which favorable forecasts are expected to be positive. This measure is interesting, particularly, in the clinical setting. For example, during the examination, correctly identifying a hazardous tumor is more crucial than wrongly identifying a benign one [29]. The recall is calculated using the following Equation (4).

$$\textbf{Recall}: = \left(\frac{\text{True Positive}}{\text{True Positive+False Negative}}\right) \text{[29] (4)}$$

C. Important Hyper Parameters for Investigation

Table 13 below shows the number of features selected for different investigation experiments. The number of features varies between five to nine.

Table 13. Number of Selected Features Setting

ID	Features selections
1	Five selections
2	Seven selections
3	Eight selections
4	Nine selections

Table 14. The splitting of data 80-20, 90-10, 70-30, and 60-40 and 4-fold-cross validation

ID	Splitting data
1	80-20%
2	90-10%
3	70-30%
4	60-40%
5	4-fold-cross validation

In the experiments, the data used for training and testing is split according to a different setting to show its effect on the performance of the model. Table 14 above shows the splitting settings including 80-20%, 90-10%, 90-10%, 70-30%, 60-40%, and 4-fold-cross validation. The function of the random state variable is to shuffle the data to get the best accuracy of the algorithm. Accordingly, shuffling the data should have different values starting from 0 to n to record the best value. It is worth noting that in this study only ten options for random state hyperparameters are used as shown in Table 15 below.

Table 15. Random state setting 1-10

ID	Random state
1	1 to 10

D. Experiments

For this research, 17 experiments aimed to evaluate and analyze the proposed combination of preprocessing steps and feature sets. In each trial split the data 80–20, 90–10, 70–30, and 60–40, and random state (1–10) with five feature selections, seven feature selections, eight feature selections, and nine feature selections. Also, four-fold validation, six-fold cross-validation, and eight-fold cross-validation were used to evaluate the performance of each.

1. Experiment 1: Algorithm Performance Investigation with Splitting 80-20, Nine Features, and Random State (1-10): Experiment 1, examined different combinations and split data with 80,20 for training and testing respectively. The selected features were nine and random state values varying between 1-10. The purpose is to choose the appropriate hyperparameters that achieve the high performance of the proposed model.

Table 16. Experiment 1 results of splitting data 80-20 % with Nine (9) features and random state 1-10

Features name	Random state	Confusion matrix					
clump thic uniformity of size Un of cell shape marginal single ep cell size normal nucleoli	serial	TP	FP	FN	TN	Accuracy	F1_Score
	1	90	0	2	45	0.9854	0.99
	2	79	4	3	51	0.9489	0.95
	3	75	3	2	57	0.9635	0.96
	4	85	5	0	47	0.9635	0.96
	5	88	2	1	46	0.9781	0.98

bland chromatin bar nuclei mitoses	6	80	1	2	54	0. 9781	0.98
	7	86	3	1	47	0.9708	0.97
	8	84	0	2	51	0.9854	0.99
	9	84	0	2	51	0.9854	0.99
	10	87	2	1	47	0.9781	0.98

2. *Experiment 2: Algorithm Performance Investigation with Splitting 90-10, Nine Features, and Random State (1-10):* Experiment 2, examined different combinations and split data with 90,10 for training and testing respectively. The selected features were nine and the random state values varied between 1-10. The purpose was to choose the appropriate values of the hyperparameters that achieved the high performance of the proposed model.

Table 17. Experiment 2 splitting data 90-10 % with Nine (9) feature random state -1-10

Features name	**Random state**	**Confusion matrix**					
clump thic uniformity of size Un of cell shape marginal single ep cell size normal nucleoli bland chromatin bar nuclei mitoses	Serial	TP	FP	FN	TN	Accuracy	F1_Score
	1	43	0	2	24	0.9710	0.97
	2	41	1	2	25	0.9565	0.96
	3	30	0	1	29	0.9855	0.99
	4	43	3	0	23	0.9565	0.96
	5	46	1	0	22	0.9855	0.99
	6	42	2	0	25	0.9710	0.97
	7	42	3	0	24	0.9565	0.96
	8	43	0	2	24	0.9710	0.97
	9	37	0	2	30	0.9710	0.97
	10	40	1	1	27	0.9710	0.97

3. Experiment 3: Algorithm Performance Investigation with Splitting 70-30, Nine Features, and Random State (1-10): Experiment 3, examined different combinations and split data with 70,30 for training and testing respectively. The selected features were nine features and the random state values varied between 1-10. The purpose was to choose the appropriate values of the hyperparameters that achieved the high performance of the proposed model.

Table 18. Experiment 3 splitting data 70-30 % with nine (9) features and random state 1-10

Features name	**Random state**	**Confusion matrix**					
clump thic uniformity of size Un of cell shape marginal single ep cell size normal nucleoli bland chromatin bar nuclei mitoses	serial	TP	FP	FN	TN	Accuracy	F1_Score
	1	129	4	4	68	0.9609	0.96
	2	116	9	3	77	0.9414	0.94
	3	124	4	4	73	0.9609	0.96
	4	122	10	3	70	0.9365	0.94
	5	129	4	4	68	0.9609	0.96
	6	128	3	3	71	0.9707	0.97
	7	126	5	1	73	0.9707	0.97
	8	121	2	4	70	0.9707	0.97
	9	126	1	2	76	0.9853	0.99
	10	128	3	1	73	0.9804	0.98

4. *Experiment 4: Algorithm Performance Investigation with Splitting 60-40, Nine Features, and Random State (1-10):* Experiment 4, examined different combinations and split data with 60,40 for training and testing respectively. The selected features were nine and the random state values varied between 1-10. The purpose was to choose the appropriate values of the hyperparameters that achieved the high performance of the proposed model.

Table 19. Experiment 4 splitting data 60-40 % with nine (9) features and random state 1-10

Features name	Random state	Confusion matrix					
clump thic uniformity of size Un of cell shape marginal single ep cell size normal nucleoli bland chromatin bar nuclei mitoses	serial	TP	FP	FN	TN	Accuracy	F1_Score
	1	174	4	4	92	0.9708	0.97
	2	164	8	4	98	0.9562	0.96
	3	171	5	3	95	0.9708	0.97
	4	165	10	3	96	0.9525	0.95
	5	172	8	4	90	0.9562	0.96
	6	180	3	2	89	0.9817	0.98
	7	167	5	2	100	0.9744	0.98
	8	162	6	3	103	0.9671	0.97
	9	173	1	3	97	0.9854	0.99
	10	171	4	4	95	0.9708	0.97

Discussion: Comparing the obtained results in Tables 16, 17, 18, and 19, and first, based on the top one results, the ranking of the obtained higher accuracies is 0.99. These results were achieved when the splitting data was 80% and 20% for training and testing, respectively. The number of features was nine, and the value of the random state was one. In the same vein, these results were achieved when the splitting data was 90% and 10% for training and testing, respectively. The number of features was nine, and the value of the random state was three. In addition, when the splitting data was 70% and 30% for training and testing, respectively. The number of features was nine, and the value of the random state was nine, achieving the best result. Finally, these results were obtained when the splitting data was 60% and 40% for training and testing, respectively. The number of features was nine, and the value of the random state was nine. Second, based on the average for the top three results, the ranking of the obtained higher accuracies is 0.99, 0.98, and 0.9766. The first-rank accuracy (i.e., 0.99%) was achieved when the splitting data was 80% and 20% for training and testing, respectively. The number of features was nine, and the values of the random state were one, eight, and nine. The second-rank accuracy (i.e., 0.98%) was achieved when the splitting data was 90% and 10% for training and testing, respectively. The number of features was nine, and the values of the random state were one, three, and five. In addition, this accuracy (i.e., 0.98%) was achieved when the splitting data was 70% and 30% for training and testing, respectively. The number of features was nine, and the values of the random state were six, nine, and ten, achieving the best result. Finally, (i.e., 0.9766%) was obtained when the splitting data was 60% and 40% for training and testing, respectively. The number of features was nine features, and the value of the random state was six, seven, and nine.

5. Experiment 5: Algorithm Performance Investigation with Splitting 80-20, Seven Features, and Random State (1-10): Experiment 5 examined different combinations and split the data with 80/20 for training and testing, respectively. The selected features were seven, and the random state values varied between 1 and 10. The purpose was to choose the appropriate values of the hyperparameters that achieved the high performance of the proposed model.

Table 20. Experiment 5 splitting data 80-20 % with seven (7) features and random state 1-10

Features name	Random state	Confusion matrix					
clump thic uniformity of size Un of cell shape marginal single ep cell size normal nucleoli mitoses	serial	TP	FP	FN	TN	Accuracy	F1_Score
	1	82	2	1	51	0.9785	0.98
	2	92	5	2	41	0.95	0.95
	3	78	2	5	55	0.95	0.95
	4	84	2	5	49	0.95	0.95
	5	89	3	3	45	0.9571	0.96
	6	89	6	2	43	0.9428	0.94
	7	82	6	4	48	0.9285	0.93
	8	83	6	5	46	0.9214	0.92
	9	90	4	0	46	0.9714	0.97
	10	94	3	1	42	0.9714	0.97

6. Experiment 6: Algorithm Performance Investigation with Splitting 90-10, Seven Features, and Random State (1-10): Experiment 6, examined different combinations and split data with 90,10 for training and testing respectively. The selected features were seven and the random state values varied between 1-10. The purpose was to choose the appropriate values of the hyperparameters that achieved the high performance of the proposed model.

Table 21. Experiment 6 splitting data 90-10 % with seven (7) features and random state 1-10

Features name	**Random state**	**Confusion matrix**					
clump thic uniformity of size Un of cell shape marginal single ep cell size normal nucleoli mitoses	serial	TP	FP	FN	TN	Accuracy	F1_Score
	1	42	0	0	28	100	100
	2	44	2	2	22	0.9428	0.94
	3	36	2	4	28	0.9142	0.91
	4	43	0	2	25	0.9714	0.97
	5	44	2	2	22	0.9428	0.94
	6	42	2	3	23	0.9285	0.93
	7	40	3	3	24	0.9142	0.91
	8	41	3	2	24	0.9285	0.93
	9	42	3	0	25	0.9571	0.96
	10	42	1	1	26	0.9714	0.97

7. Experiment 7: Algorithm Performance Investigation with Splitting 70-30, Seven Features, and Random State (1-10): Experiment 7, examined different combinations and split data with 70,30 for training and testing respectively. The selected features were seven and the random state values varied between 1-10. The purpose was to choose the appropriate values of the hyperparameters that achieved the high performance of the proposed model.

Table 22. Experiment 7 splitting data 70-30 % with seven (7) features and random state 1-10

Features name	**Random state**	**Confusion matrix**					
clump thic uniformity of size Un of cell shape marginal single ep cell size normal nucleoli mitoses	Serial	TP	FP	FN	TN	Accuracy	F1_Score
	1	138	4	2	65	0.9712	0.97
	2	134	6	7	62	0.9377	0.94
	3	126	4	7	72	0.9473	0.95
	4	125	4	8	72	0.9425	0.94
	5	129	4	4	72	0.9617	0.96
	6	134	7	4	64	0.9473	0.95
	7	126	6	10	67	0.9234	0.92
	8	123	6	7	73	0.9377	0.94
	9	128	3	5	73	0.9617	0.96
	10	135	6	3	65	0.9569	0.96

8. Experiment 8: Algorithm Performance Investigation with Splitting 60-40, Seven Features, and Random State (1-10): Experiment 8 examined different combinations and split data with 60,40 for training and testing respectively. The selected features were seven and the random state values varied between 1-10. The purpose was to choose the appropriate values of the hyperparameters that achieved the high performance of the proposed model.

Table 23. Experiment 8 splitting data 60-40 % with seven (7) features and random state 1-10

Features name	**Random state**	**Confusion matrix**					
clump thic uniformity of size Un of cell shape marginal single ep cell size	Serial	TP	FP	FN	TN	Accuracy	F1_Score
	1	181	5	6	87	0.9605	0.96
	2	174	7	10	88	0.9390	0.94
	3	164	5	11	99	0.9426	0.94

normal nucleoli mitoses	4	166	5	12	96	0.9390	0.94
	5	175	7	5	92	0.9569	0.96
	6	186	7	4	82	0.9605	0.96
	7	170	9	9	91	0.9354	0.94
	8	170	7	7	95	0.9498	0.95
	9	169	5	4	101	0.9677	0.97
	10	177	8	5	89	0.9534	0.95

Discussion: Comparing the obtained results there are in Table 20 Table 21, Table 22, and Table 23 deal with breast cancer datasets, and first, based on the top result, the ranking of the obtained higher accuracies is 100, 0.98, and 0.97. The first-rank accuracy (i.e. 100 %) was achieved when the splitting data was 90%, and 10% for training and testing respectively. The number of features was seven, and the value of the random state was one. The second-rank accuracy (i.e. 98 %) was achieved when the splitting data was 80%, and 20% for training and testing respectively. The number of features was seven, and the value of the random state was one. The third rank accuracy (i.e. 97 %) was achieved when the splitting data was 70%, and 30% for training and testing respectively. The number of features was seven, and the value of the random state was one. In addition, this accuracy (i.e. 97 %) was achieved when the splitting data was 60%, and 40% for training and testing respectively. The number of features was seven, and the value of the random state was 9. Second, based on the average for the top three results the ranking of the obtained higher accuracies is 0.98, 0.9733, and 0.9633. The first-rank accuracy (i.e. 0.98 %) was achieved when the splitting data was 90%, and 10% for training and testing respectively. The number of features was seven, and the values of the random state were one, four, and ten. The second-rank accuracy (i.e. 0.9733 %) was achieved when the splitting data was 80%, and 20% for training and testing respectively. The number of features was seven, and the values of random state were one, nine, and ten. The third-rank accuracy (i.e. 0.9633 %) was achieved when the splitting data was 70%, and 30% for training and testing respectively. The number of features was seven, and the values of the random state were one, five, and nine achieving the best result. As well, this accuracy (i.e. 0.9633 %) was obtained when the splitting data was 60%, and 40% for training and testing respectively. The number of features was seven, and the values of the random state were one, six, and nine.

9. Experiment 9: Algorithm Performance Investigation with Splitting 80-20, Eight Features, and Random State (1-10): Experiment 9, examined different combinations and split data with 80,20 for training and testing respectively. The selected features were eight and the random state values varied between 1-10. The purpose was to choose the appropriate values of the hyperparameters that achieved the high performance of the proposed model.

Table 24. Experiment 9 splitting data 80-20 % with eight (8) features and random state 1-10

Features name	**Random state**	**Confusion matrix**					
clump thic uniformity of size Un of cell shape marginal single ep cell size normal nucleoli bland chromatin mitoses	Serial	TP	FP	FN	TN	Accuracy	F1_Score
	1	82	2	2	50	0.9714	0.97
	2	93	4	2	41	0.9571	0.96
	3	78	2	4	56	0.9571	0.96
	4	84	2	3	51	0.9642	0.96
	5	86	6	2	46	0.9428	0.94
	6	89	6	2	43	0.9428	0.94
	7	82	6	4	48	0.9285	0.93
	8	84	6	2	49	0.95	0.95
	9	90	4	0	46	0.9714	0.97
	10	94	3	1	42	0.9714	0.97

10. Experiment 10: Algorithm Performance Investigation with Splitting 90-10, Eight Features, and Random State (1-10): Experiment 10, examined different combinations and split data with 90,10 for training and testing respectively. The selected features were eight and the random state values varied between 1-10. The purpose was to choose the appropriate values of the hyperparameters that achieved the high performance of the proposed model.

Table 25. Experiment 10 splitting data 90-10 % with eight (8) features and random state 1-10

Features name	Random state	Confusion matrix					
clump thic uniformity of size Un of cell shape marginal single ep cell size normal nucleoli bland chromatin mitoses	serial	TP	FP	FN	TN	Accuracy	F1_Score
	1	42	0	0	28	100	100
	2	44	2	2	22	0.9428	0.94
	3	36	2	4	28	0.9142	0.91
	4	43	0	1	25	0.9714	0.97
	5	42	4	1	25	0.9285	0.93
	6	42	2	2	24	0.9428	0.94
	7	40	3	3	24	0.9142	0.91
	8	41	3	1	25	0.9428	0.94
	9	41	4	0	25	0.9428	0.94
	10	42	1	1	26	0.9714	0.97

11. Experiment 11: Algorithm Performance Investigation with Splitting 70-30, Eight Features and Random State (1-10): Experiment 11, examined different combinations and split data with 70,30 for training and testing respectively. The selected features were eight and the random state values varied between 1-10. The purpose was to choose the appropriate values of the hyperparameters that achieved the high performance of the proposed model.

Table 26. Experiment 11 splitting data 70-30 % with eight (8) features and random state 1-10

Features name	Random state	Confusion matrix					
clump thic uniformity of size Un of cell shape marginal single ep cell size normal nucleoli bland chromatin mitoses	serial	TP	FP	FN	TN	Accuracy	F1_Score
	1	137	5	3	64	0.9617	0.96
	2	134	6	4	65	0.9521	0.95
	3	124	6	5	74	0.9473	0.95
	4	126	3	4	76	0.9665	0.97
	5	126	7	3	73	0.9521	0.95
	6	133	8	3	65	0.9473	0.95
	7	126	6	9	68	0.9282	0.93
	8	123	6	6	74	0.9425	0.94
	9	127	4	3	75	0.9665	0.97
	10	135	65	1	67	0.9665	0.97

12. Experiment 12: Algorithm Performance Investigation with Splitting 60-40, Eight Features and Random State (1-10): Experiment 12, examined different combinations and split data with 60,40 for training and testing respectively. The selected features were eight and the random state values varied between 1-10. The purpose was to choose the appropriate values of the hyperparameters that achieved the high performance of the proposed model.

Table 27. Experiment 12 splitting data 60-40 % with eight (8) features and random state 1-10

Features name	Random state	Confusion matrix					
clump thic uniformity of size Un of cell shape marginal single ep cell size normal nucleoli bland chromatin mitoses	serial	TP	FP	FN	TN	Accuracy	F1_Score
	1	179	7	6	87	0.9534	0.95
	2	173	8	7	91	0.9462	0.95
	3	164	5	6	104	0.9605	0.96
	4	165	6	9	99	0.9462	0.95
	5	174	8	5	92	0.9534	0.95
	6	186	7	5	81	0.9569	0.96
	7	171	8	9	91	0.9390	0.94

	8	171	6	8	94	0.9498	0.95
	9	170	4	3	102	0.9749	0.98
	10	179	6	4	90	0.9641	0.96

Discussion: Comparing the obtained results there are in Table 24, Table 25, Table 26, and Table 27 dealing with breast cancer datasets, and first, based on the top one result, the ranking of the obtained higher accuracies 100, 0.97. The first rank accuracy (i.e. 100 %) was achieved when the splitting data was 90%, and 10% for training and testing respectively. The number of features was eight, and the value of the random state was one, nine, and ten. The second-rank accuracy (i.e. 97 %) was achieved when the splitting data was 80%, and 20% for training and testing respectively. The number of features was eight, and the value of the random state was one. In addition, this accuracy (i.e. 97 %) was achieved when the splitting data was 70%, and 30% for training and testing respectively. The number of features was eight, and the value of the random state was one and nine. As well, this accuracy (i.e. 97 %) was achieved when the splitting data was 60%, and 40% for training and testing respectively. The number of features was eight features, and the value of the random state was nine.

Second, based on the average for the top three results, the ranking of the obtained higher accuracies is 0.98, 0.9714, 0.97, and 0.9666. The first-rank accuracy (i.e. 0.98 %) was achieved when the splitting data was 90%, and 10% for training and testing respectively. The number of features was eight, and the values of the random state were one, four, and ten. The second-rank accuracy (i.e. 0.9714 %) was achieved when the splitting data was 80%, and 20% for training and testing respectively. The number of features was eight, and the values of the random state were one, nine, and ten. The third-rank accuracy (i.e. 0.97 %) was achieved when the splitting data was 70%, and 30% for training and testing respectively. The number of features was eight, and the values of the random state were four, nine, and ten achieving the best result. The fourth-rank accuracy (i.e. 0.9666 %) was obtained when the splitting data was 60%, and 40% for training and testing respectively. The number of features was eight, and the values of the random state were three, nine, and ten.

13. Experiment 13: Algorithm Performance Investigation with Splitting 80-20, Five Features, and Random State (1-10): Experiment 13, examined different combinations and split data with 80,20 for training and testing respectively. The selected features were five and the random state values varied between 1-10. The purpose was to choose the appropriate values of the hyperparameters that achieved the high performance of the proposed model.

Table 28. Experiment 13 splitting data 80-20 % with five (5) features and random state 1-10

Features name	**Random state**	**Confusion matrix**					
clump thic uniformity of size Un of cell shape marginal mitoses	serial	TP	FP	FN	TN	Accuracy	F1_Score
	1	88	0	1	51	0.9928	0.99
	2	92	5	2	41	0.95	0.95
	3	78	2	5	55	0.95	0.95
	4	84	2	5	49	0.95	0.95
	5	89	3	5	43	0.9428	0.94
	6	89	6	3	42	0.9357	0.94
	7	82	6	3	49	0.9357	0.94
	8	85	4	6	45	0.9285	0.93
	9	90	4	2	44	0.9571	0.96
	10	94	3	2	41	0.9642	0.96

14. Experiment 14: Algorithm Performance Investigation with Splitting 90-10, Five Features and Random State (1-10): Experiment 14, examined different combinations and split data with 90,10 for training and testing respectively. The selected features were five and the random state values varied between 1-10. The purpose was to choose the appropriate values of the hyperparameters that achieve the high performance of the proposed model.

Table 29. Experiment 14 splitting data 90-10 % with five (5) features and random state 1-10

Features name	**Random state**	**Confusion matrix**					
clump thic uniformity of size Un of cell shape	serial	TP	FP	FN	TN	Accuracy	F1_Score
	1	42	0	0	28	100	100
	2	44	2	2	22	0.9428	0.94

marginal mitoses	3	36	2	3	29	0.9285	0.93
	4	43	0	1	26	0.9857	0.99
	5	44	2	4	20	0.9142	0.91
	6	44	0	2	24	0.9714	0.97
	7	40	3	2	25	0.9285	0.93
	8	42	2	3	23	0.9285	0.93
	9	42	3	2	23	0.9285	0.93
	10	42	1	2	25	0.9571	0.96

15. Experiment 15: Algorithm Performance Investigation with Splitting 70-30, Five Features, and Random State (1-10): Experiment 15, examined different combinations and split data with 70,30 for training and testing respectively. The selected features were five and the random state values varied between 1-10. The purpose was to choose the appropriate values of the hyperparameters that achieved the high performance of the proposed model.

Table 30. Experiment 15 splitting data 70-30 % with five (5) features and random state 1-10

Features name	**Random state**	**Confusion matrix**					
clump thic uniformity of size Un of cell shape marginal mitoses	serial	TP	FP	FN	TN	Accuracy	F1_score
	1	139	3	3	64	0.9712	0.97
	2	134	6	9	60	0.9282	0.93
	3	126	4	8	71	0.9425	0.94
	4	126	3	10	70	0.9377	0.94
	5	129	4	6	70	0.9521	0.95
	6	134	7	6	62	0.9377	0.94
	7	124	8	9	68	0.9186	0.92
	8	123	6	8	72	0.9330	0.93
	9	127	4	7	71	0.9473	0.95
	10	135	6	3	65	0.9569	0.96

16. Experiment 16: Algorithm Performance Investigation with Splitting 60-40, Five Features, and Random State (1-10): Experiment 16, examined different combinations and split data with 60,40 for training and testing respectively. The selected features were five and the random state values varied between 1-10. The purpose was to choose the appropriate values of the hyperparameters that achieved the high performance of the proposed model.

Table 31. Experiment 16 splitting data 60-40 % with five (5) features and random state 1-10

Features name	**Random state**	**Confusion matrix**					
clump thic uniformity of size Un of cell shape marginal mitoses	serial	TP	FP	FN	TN	Accuracy	F1_Score
	1	183	3	10	83	0.9534	0.95
	2	175	6	12	86	0.9354	0.94
	3	166	3	12	98	0.9462	0.95
	4	164	7	14	94	0.9247	0.93
	5	177	5	10	87	0.9462	0.95
	6	186	7	8	78	0.9462	0.95
	7	169	10	11	89	0.9247	0.93
	8	170	7	8	94	0.9462	0.95
	9	169	5	8	97	0.9534	0.95
	10	177	8	6	88	0.9498	0.95

Discussion:

Comparing the obtained results, there are in Table 28, Table 29, Table 30, and Table 31 dealing with breast cancer datasets, and first, based on the top one results, the ranking of the obtained higher accuracy is 100, 0.99, 0.97, and 0.95. The first rank accuracy (i.e., 100%) was achieved when the splitting data was 90% and 10% for training and testing, respectively. The number of features was five, and the value of the random state was one. The second-rank accuracy (i.e., 0.99%) was achieved when the splitting data was 80% and 20% for training and testing, respectively. The number of features was five, and the value of the random state was one. The third rank accuracy (i.e., 97%) was achieved when the splitting data was 70% and 30% for training and testing, respectively. The number of features was five, and the value of the random state was one. The fourth-rank accuracy (i.e., 95%) was achieved when the splitting data was 60% and 40% for training and testing, respectively. The number of features was five, and the value of the random state was one.

Second, based on the average for the top three results, the ranking of the obtained higher accuracy is 0.9766, 0.97, 0.96, and 0.95. The first-rank accuracy (i.e., 0.9766%) was achieved when the splitting data was 90% and 10% for training and testing, respectively. The number of features was five, and the values of the random state were one, six, and ten. The second-rank accuracy (i.e., 0.97%) was achieved when the splitting data was 80% and 20% for training and testing, respectively. The number of features was five, and the values of the random state were one, nine, and ten. The third-rank accuracy (i.e., 0.96%) was achieved when the splitting data was 70% and 30% for training and testing, respectively. The number of features was five, and the values of the random state were one, five, and ten, achieving the best result. The fourth-rank accuracy (i.e., 0.95%) was obtained when the splitting data was 60% and 40% for training and testing, respectively. The number of features was five, and the values of the random state were one, three, and nine.

V. RESULT

Table 32 below shows the best-obtained results based on the top one results.

Table 32. The Best Hyperparameter to Achieve Higher Results Based on the Top One Results

Features selection	Splitting data	Random state	Accuracy
Five features	80-20%	1	99.28 %
Five features	90-10%	1	100 %
Seven features	90-10%	1	100 %
Eight features	90-10%	1	100 %
Nine features	90-10	3	99

The results shown in Figure 5 below and Table 32 above illustrate the higher accuracy obtained based on the top one results. It is clear that the algorithm reached the best accuracy of 100% when the splitting data was 90%and 10%, the value of the random state was one and the number of selected features was five, seven, and eight. The second-best results (i.e. 99.28%) were obtained when the splitting data was 80%and 20%, the value of the random state was one and the number of features was five. In addition, splitting data 90-10 with nine features and the random states three achieved an accuracy of 99%. It is worth noting that five, seven, and eight features with the splitting data as 90%-10% and a random state value of one attained a higher accuracy. The random state value of one and the large distance between training and testing data is the optimal hyperparameter to achieve higher accuracy. The features between 5-9 affected the model performance.

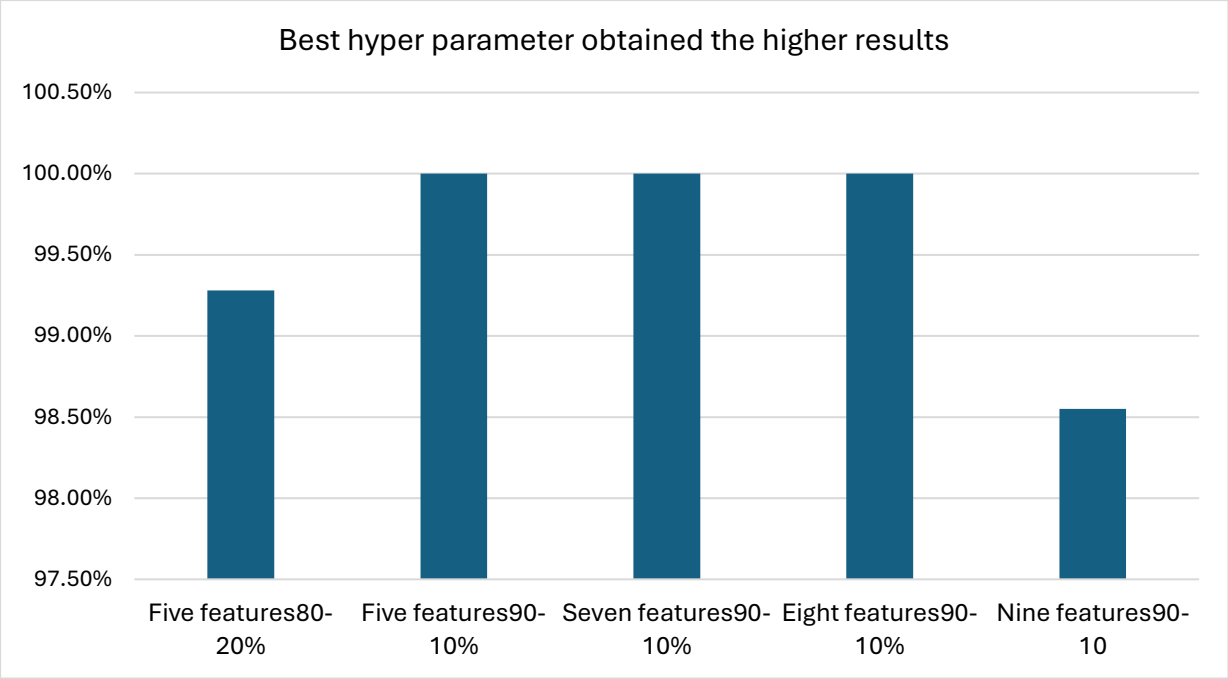

Fig. 5. Best hyperparameters obtained the higher results based on the top one results

Finally, Table 33 shows the best-obtained results based on the average for the top three results, which are combined in one table and compared with the results of previous studies shown in Section 5.3.

Table 33. The Best Hyperparameter to Achieve Higher Results Based on the Average for the Top Three Results

Features selection	**Splitting data**	**Random state**	**Accuracy**
Seven features	90-10%	1,4,10	0.98
Eight features	90-10%	1,4,10	0.98
Nine features	80-20	1,8,9	0.99
Nine features	70-30	6,9,10	0.98
Nine features	90-10	1,3,5	0.98

The results shown in Figure 6 below and Table 33 above illustrated a higher accuracy obtained based on the average for the top three results. The algorithm reached the best accuracy ranking of 0.99, and 0.98. The first rank (i.e. 0.99%) was achieved when the splitting data 80-10, nine features, and the random state values were one, eight, and nine. The second rank (i.e. 0.98%) was achieved when the splitting data was 90-10, seven features, and the random state values were one, four, and ten. Besides, this accuracy (i.e. 0.98%) was achieved when the splitting data was 90-10, eight features, and the random state values were one, four, and ten. In addition, this accuracy (i.e. 0.98%) was achieved when the splitting data 70-30, nine features, and the random state values were six, nine, and ten. As well, this accuracy (i.e. 0.98%) was achieved when the splitting data 90-10, nine features, and the random state values were one, three, and five. This may be attributed to the large distance between training and testing data and random state values (1,10) are the optimal hyperparameter to achieve higher accuracy. Aside from the features between 5-9 affected the model performance.

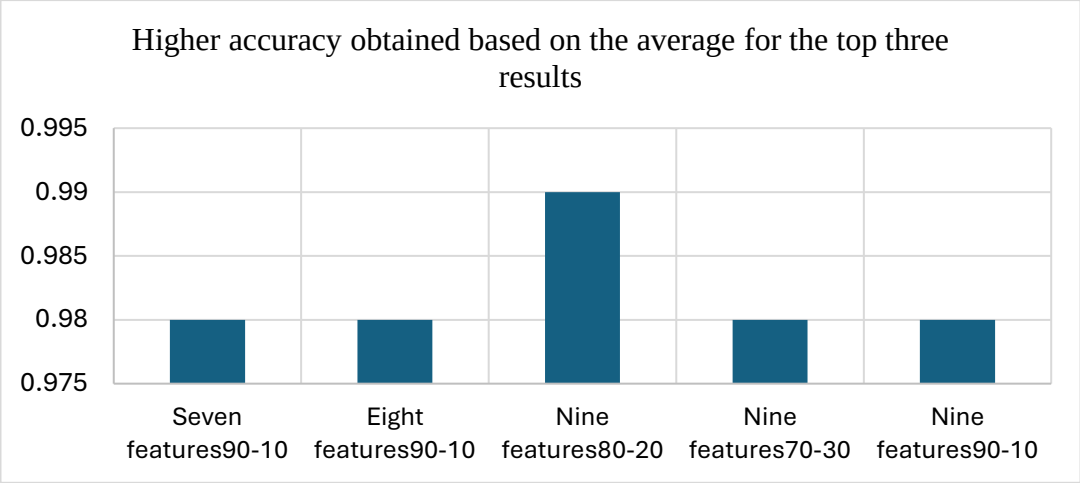

Fig. 6. Best hyperparameter to achieve higher results based on the average for the top three results.

A. A Comparison Between Related Works and Proposed Model Based on the Top One Results

Table 34 Compares previous study results with proposed model results, based on the top result.

Author	Dataset	Features	Model	Acc
Our work with five features splitting data (90-10)	WBCO	5-Features	SVM	100 %
Our work with seven features splitting data (90-10)	WBCO	7-Features	SVM	100 %
Our work with eight features splitting data (90-10)	WBCO	8-Features	SVM	100 %
Our work with five features % splitting data (80-20)	WBCO	5-Features	SVM	99.28 %
Hiba Asria et al	WBCO	9-Features	SVM	97.13
Madhu Kumaria et al.	WBCO	9-Features	SVM	97.38%
Na Liua, et.al	WBCO	9-Features	SVM	95.7%
Anika Islam Aishwarja1 et al.	WBCO	9-Features	SVM	94.50%
Dana Bazazeh1 et al	WBCO	9-Features	SVM	97%
Showrov, Islam, et al (2019)	WBCO	9-Features	SVM	96.72%
Hajar Saoud1 et al	WBCO	7-Features	SVM	97.2818
Dada Emmanuel Gbenga et al	WBCO	9-Features	SVM	97.7%
Hanumanthu Bhukya [8]	WBCO	9-Features	SVM	94.23%
Reem Alyami1 et al.	WBCO	9-Features	SVM	97.138%
Hicham Omara et al	WBCO	9-Features	SVM	95.70%
Yixuan Li1 et al.	WBCO	9-Features	SVM	97.7%
Mohammed Elnahas et al	WBCO	9-Features	SVM	97.2%
Onyinyechi Jessica Egwom 1 et al.	WBCO	9-Features	SVM	97.8%

B. A Discussion of Related Works with Proposed Model Based on the Top One Results

Some studies tackled the breast cancer prediction problem using the breast cancer dataset and several classification methods. Table 34 above illustrates a comparison with existing methods in terms of accuracy on the breast cancer dataset. [26] Experimented SVM on breast cancer dataset resulted in 97.13% in terms of accuracy, as well [10] examined SVM on breast cancer dataset resulted in 94.50% in terms of accuracy, [27] experimented SVM on breast cancer dataset resulted in 97% in terms of accuracy, in

additional [16] used SVM on breast cancer dataset resulted in 96.72% in terms of accuracy, besides [20] applied SVM on breast cancer dataset resulted in 97.2818% in terms of accuracy, as well [24] examined SVM on breast cancer dataset resulted in 97.7% in terms of accuracy, , in additional [9] used SVM on breast cancer dataset resulted in 97.8% in terms of accuracy. The results are outperformed by our model using the same algorithms on the breast cancer dataset, this may be attributed to the appropriate partition data, random state values, and appropriate feature selection, which allowed the model to significantly improve its performance.

C. A Comparison Between Related Works and Proposed Model Based on the Average for the Top Three Results

Table 35 Compare previous study results with proposed model results based on the average for the top three results

Author	Dataset	Features	Model	Acc
Our work with Seven features splitting data 90-10	WBCO	7-Features	SVM	98%
Our work with Eight features splitting data 90-10	WBCO	8-Features	SVM	98%
Our work with Nine features splitting data 80-20	WBCO	9-Features	SVM	99%
Our work with Nine features splitting data 90-10	WBCO	9-Features	SVM	98%
Our work with Nine features splitting data 70-30	WBCO	9-Features	SVM	98%
Hiba Asria et al	WBCO	9-Features	SVM	97.13%
Madhu Kumaria et al.	WBCO	9-Features	SVM	97.38%
Na Liua, et.al	WBCO	9-Features	SVM	95.7%
Anika Islam Aishwarja1 et al.	WBCO	9-Features	SVM	94.50%
Dana Bazazeh1 et al	WBCO	9-Features	SVM	97%
Showrov, Islam, et al (2019)	WBCO	9-Features	SVM	96.72%
Hajar Saoud1 et al	WBCO	7-Features	SVM	97.2818%
Dada Emmanuel Gbenga et al	WBCO	9-Features	SVM	97.7%
Hanumanthu Bhukya	WBCO	9-Features	SVM	94.23%
Reem Alyami1 et al.	WBCO	9-Features	SVM	97.138%
Carson K. Leung∗ et al.	WBCO	9-Features	SVM	49%
Hicham Omara et al	WBCO	9-Features	SVM	95.70%
Yixuan Li1 et al.	WBCO	9-Features	SVM	97.7%
Mohammed Elnahas et al	WBCO	9-Features	SVM	97.2%
Onyinyechi Jessica Egwom 1 et al.	WBCO	9-Features	SVM	97.8%

D. A Discussion of Related Works with Proposed Model Based on the Average for the Top Three Results

Table 35 above illustrates a comparison. With existing methods in terms of accuracy on the breast cancer dataset. [26] Experimented SVM on breast cancer dataset resulted in 97.13% in terms of accuracy, as well [10] examined SVM on breast cancer dataset resulted in 94.50% in terms of accuracy, [27] experimented SVM on breast cancer dataset resulted in 97% in terms of accuracy, in additional [16] used SVM on breast cancer dataset resulted in 96.72% in terms of accuracy, besides [20] applied SVM on breast cancer dataset resulted in 97.2818% in terms of accuracy, as well [24] examined SVM on breast cancer dataset resulted in 97.7% in terms of accuracy, , in additional [9] used SVM on breast cancer dataset resulted in 97.8% in terms of accuracy. The results are outperformed by our model using the same algorithms on the breast cancer dataset, this may be attributed to the appropriate partition data, random state values, and appropriate feature selection, that allowed the model to significantly improve its performance.

Overall, the results of the proposed model showed that the use of appropriate hyperparameters with the classifier improves the classification of accuracy performance. The results of the proposed model showed its superiority in terms of classifier accuracy. This may be attributed to the appropriate split data, random state values, and appropriate feature selection, which allowed the proposed model to significantly improve its performance, however, previous studies achieved less accuracy This may be attributed to the not used suitable hyperparameters.

VI. CONCLUSIONS

First, record results based on the top one result: When split, the data was 90-10% for training and testing, respectively, with applied five, seven, and eight features, and the random state was one; the model achieved a higher accuracy of 100%. However, splitting the data 80-20% for training and testing, respectively, with five features and the random state was one, the model achieved a higher accuracy of 0.9928%. First research question: Do the hyperparameters impact the algorithm's performance? The hyperparameter affected the algorithm's performance. If realized that the hyperparameters affect the algorithm's performance, a second research question was, what are the best values for the hyperparameters that make the algorithm achieve a higher result? So, splitting the data 90-10 with five, seven, and eight features, the random state was one; splitting the data 80-20 with five features and the random state was one represents the best hyperparameter values to achieve a higher result.

Second, record results based on the average for the top three results: When split, the data was 90-10% for training and

testing, respectively, with applied seven features, and the random state values were one, four, and ten. As well, eight features split the data 90-10% for training and testing, respectively, and the random state values were one, four, and ten. Besides, nine features split the data 90-10% for training and testing, respectively, and the random state values were one, three, and five. Aside from nine features, split the data 70-30% for training and testing, respectively, and the random state values were six, nine, and ten; the model achieved a higher accuracy of 0.98%. However, splitting the data 80-20% for training and testing, respectively, used nine features, and the random state values were one, eight, and nine; the model achieved a higher accuracy of 0.99%. First research question: Do the hyperparameters impact the algorithm's performance? The hyperparameter affected the algorithm's performance. If realized that the hyperparameters affect the algorithm's performance, a second research question was, what are the best values for the hyperparameters that make the algorithm achieve a higher result? So, split data 90-10% with applied seven features and random state values were one, four, and ten. As were eight features, split data 90-10%, and random state values were one, four, and ten. Aside from that, nine features with split data of 90-10% and random state values were one, three, and five. In addition, nine features, split data 70-30%, and random state values were six, nine, and ten. Besides, split data 80-20% used nine features, and random state values were one, eight, and nine, representing the best hyperparameter values to achieve a higher result.

REFERENCES

[1] Sariego, J., "Breast cancer in the young patient," *Am Surg*, vol. 76, no. 12, pp. 1397-1400, 2010.

[2] A. G. Waks and E. P. Winer, "Breast Cancer Treatment: A Review," *JAMA*, vol. 321, no. 3, pp. 288-300, 2019, doi: 10.1001/jama.2018.19323.

[3] L. S. Caplan, K. J. Helzlsouer, S. Shapiro, M. N. Wesley, and B. K. Edwards, "Reasons for delay in breast cancer diagnosis," *Preventive Medicine*, vol. 25, no. 2, pp. 218-224, 1996.

[4] B. S. Hulka and A. T. Stark, "Breast cancer: cause and prevention," *The Lancet*, vol. 346, no. 8979, pp. 883-887, 1995.

[5] R. PadmaPriya and P. S. Vadivu, "A Review on Data Mining Techniques for Prediction of Breast Cancer Recurrence," *International Journal of Engineering and Management Research*, e-ISSN: 2250-0758, 2019.

[6] C. Oprea and Ș. Ti, "Performance evaluation of the data mining classification methods," *Information Society and Sustainable Development*, vol. 1, pp. 249-253, 2014.

[7] Y. I. Rejani and S. Thamarai, "Early Detection of Breast Cancer Using SVM Classifier Technique," *International Journal on Computer Science and Engineering*, vol. 1, 2009.

[8] H. Bhukya and M. Sadanandam, "RoughSet based feature selection for prediction of breast cancer," Wireless Pers. Commun., vol. 130, no. 3, pp. 2197–2214, 2023.

[9] O. J. Egwom, M. Hassan, J. J. Tanimu, M. Hamada, and O. M. Ogar, "An LDA–SVM Machine Learning Model for Breast Cancer Classification," *BioMed Informatics*, vol. 2, no. 3, pp. 345-358, 2022.

[10] A. I. Aishwarja, N. J. Eva, S. Mushtary, Z. Tasnim, N. I. Khan, and M. N. Islam, "Exploring the machine learning algorithms to find the best features for predicting breast cancer and its recurrence," presented at the *3rd International Conference on Intelligent Computing and Optimization (ICO 2020)*, 2021.

[11] Dheeru and C. Graff, "UC Irvine Machine Learning Repository," *UCI Machine Learning Repository*, 2017. [Online]. Available: https://archive.ics.uci.edu/ml/index.php

[12] M. Z. A. Shawarib, A. E. A. Latif, B. E. E.-D. Al-Zatmah, and S. S. Abu-Naser, "Breast cancer diagnosis and survival prediction using JNN," *International Journal of Engineering and Information System*, vol. 23-30, 2020, ISSN: 2643-640X.

[13] M. Elnahas, M. Hussein, and A. Keshk, "An artificial neural network as an ensemble technique fuser for improving classification accuracy," presented at the *2019 Ninth International Conference on Intelligent Computing and Information Systems (ICICIS)*, 2019.

[14] G. Priyanka, P. K. Sahoo, V. Rohith, and K. Eswaran, "Breast cancer prediction system using KE Sieve algorithm," *International Journal of Scientific & Engineering Research*, vol. 10, no. 1, pp. 19-21, 2019.

[15] N. Liu, E.-S. Qi, M. Xu, B. Gao, and G.-Q. Liu, "A novel intelligent classification model for breast cancer diagnosis," *Information Processing & Management*, vol. 56, no. 3, pp. 609-623, 2019.

[16] M. I. H. Showrov, M. T. Islam, M. D. Hossain, and M. S. Ahmed, "Performance comparison of three classifiers for the classification of breast cancer dataset," presented at the *2019 4th International Conference on Electrical Information and Communication Technology (EICT)*, 2019.

[17] V. Chaurasia, S. Pal, and B. Tiwari, "Prediction of benign and malignant breast cancer using data mining techniques," *Journal of Algorithms & Computational Technology*, vol. 12, no. 2, pp. 119-126, 2018.

[18] M. Kumari and V. Singh, "Breast cancer prediction system," *Procedia Computer Science: International Conferences on Computational Intelligence and Data Science*, vol. 132, pp. 371-376, 2018.

[19] H. Mansourifar and W. Shi, "Toward efficient breast cancer diagnosis and survival prediction using L-perceptron," *arXiv preprint arXiv:1811.03016*, 2018.

[20] H. Saoud, A. Ghadi, M. Ghailani, and B. A. Abdelhakim, "Using feature selection techniques to improve the accuracy of breast cancer classification," presented at *Innovations in Smart Cities Applications Edition 2: The Proceedings of*

the Third International Conference on Smart City Applications, 2019.

[21] Y. Li and Z. Chen, "Performance evaluation of machine learning methods for breast cancer prediction," *Applied Computational Mathematics*, vol. 7, no. 4, pp. 212-216, 2018.

[22] P. Hamsagayathri and P. Sampath, "Performance analysis of breast cancer classification using decision tree classifiers," *International Journal of Current Pharm Research*, vol. 9, no. 2, pp. 19-25, 2017.

[23] R. Alyami, J. Alhajjaj, B. Alnajrani, I. Elaalami, A. Alqahtani, N. Aldhafferi, T. O. Owolabi, and S. O. Olatunji, "Investigating the effect of correlation-based feature selection on breast cancer diagnosis using artificial neural network and support vector machines," presented at the *2017 International Conference on Informatics, Health & Technology (ICIHT)*, 2017.

[24] D. E. Gbenga, N. Christopher, D. C. Yetunde, and N. Maiduguri, "Performance comparison of machine learning techniques for breast cancer detection," *Nova Journal of Engineering and Applied Sciences*, vol. 6, no. 1, pp. 1-8, 2017.

[25] H. Omara, M. Lazaar, and Y. Tabii, "Classification of breast cancer with improved self-organizing maps," presented at the *Proceedings of the 2nd International Conference on Big Data, Cloud and Applications*, 2017.

[26] H. Asri, H. Mousannif, H. Al Moatassime, and T. Noel, "Using machine learning algorithms for breast cancer risk prediction and diagnosis," *Procedia Computer Science*, vol. 83, pp. 1064-1069, 2016.

[27] D. Bazazeh and R. Shubair, "Comparative study of machine learning algorithms for breast cancer detection and diagnosis," presented at the *2016 5th International Conference on Electronic Devices, Systems, and Applications (ICEDSA)*, 2016.

[28] L. R. Borges, "Analysis of the Wisconsin breast cancer dataset and machine learning for breast cancer detection," *Proceedings of XI Workshop de Visao Computacional*, vol. 1, no. 369, pp. 15-19, 1989.

[29] H. G. V. K. Donga, "Comparing Machine Learning Models: For Diagnosis of Breast Cancer," *Diva-Portal*, 2022. [Online]. Available: https://www.diva-portal.org/smash/get/diva2:1679145/FULLTEXT02