Performance analysis of machine learning algorithms in breast cancer diagnosis

Yihim Chan
School of Computing
Asia Pacific University of Technology and Innovation (APU)
Kuala Lumpur, Malaysia
tp050324@mail.apu.edu.my

Choy Yeng Chin
School of Computing
Asia Pacific University of Technology and Innovation (APU)
Kuala Lumpur, Malaysia
tp051475@mail.apu.edu.my

Kiang Xin Chen
School of Computing
Asia Pacific University of Technology and Innovation (APU)
Kuala Lumpur, Malaysia
tp050213@mail.apu.edu.my

Feng Zhan Kor Derick
School of Computing
Asia Pacific University of Technology and Innovation (APU)
Kuala Lumpur, Malaysia
tp051318@mail.apu.edu.my

Zailan Arabee Abdul Salam
School of Computing
Asia Pacific University of Technology and Innovation (APU)
Kuala Lumpur, Malaysia
zailan@apu.edu.my

***Abstract*— Breast cancer is recognized as one of the foremost causes of death among women in worldwide with more than one million of cases and nearly 600,000 deaths each year. It is extremely important to identify it at the early stage in considerably to increase the chances of survival. The breast cancer can be classified into two types of tumors which are benign and malignant. Benign tumors are undangerous tumors where they develop slowly in organ while malignant are dangerous tumors where they would spread to the other organ of body. In this paper, five machine learning algorithms are used to predict if the tumor is benign or malignant based on the Wisconsin Prognostic Breast Cancer dataset, while one of the algorithms is modified to achieve a better performance. The five algorithms used are Gaussian Naïve Bayes Classifier, Random Forest Classifier, Decision Tree Classifier, Kernel Support Vector Machine Classifier, and K-Nearest Neighbors Classifier while the modified algorithm is Kernel Support Vector Machine Classifier. The aim is to use Machine Learning algorithms to make prediction of breast cancer and improved the accuracy of the algorithm. 10-fold Cross Validation is implemented after compared it with Bootstrapping as a resampling method as it is more efficient. At the end, the comparison in results shown that the modified Kernel Support Vector Machine Classifier predicted the highest accuracy among these five machine learning models.**

Keywords—Breast cancer, machine learning, classification, diagnosis, UCI machine learning repository

I. INTRODUCTION

Breast Cancer is a type of cancer characterized by abnormal cell growth due to unregulated cell division, which results in the formation of breast tissue lumps known as breast tumors. While benign breast tumors are non-cancerous and do not metastasize, malignant breast tumors are cancerous and will potentially spread to other body parts. The most common types of breast cancer affecting women worldwide are Ductal Carcinoma In Situ (DCIS) and Invasive Ductal Carcinoma (IDC). In fact, the World Health Organization (WHO) suggested that breast cancer has accounted for 627,000 or 15% of women's cancer mortality rate. Therefore, early diagnosis plays a crucial role in elevating the survival rate of breast cancer patients.

The curve is shown to be flattening in recent years as a great number of machine learning approaches have been introduced in breast cancer diagnosis, especially in breast cancer tumor classification. However, greater number of simulations, larger data samples, and more testing using different ML algorithms are required before the classification models can be deployed for public clinical use. Hence, this paper aimed to study the performance of Gaussian Naïve Bayes, Random Forest, Decision Tree, and K-Nearest Neighbor Classifier and evaluate against the Support Vector Machine (SVM) to identify the best classification model.

In this experiment, the SVM algorithm will be implemented to examine its effectiveness and efficiency compared to other models. The purpose of implementing this algorithm is because it is known to be able to handle a huge amount of complex data and it is capable of precisely isolating the data based on the defined labels or outputs without performing complex transformations.

The experiment will be carried out on the Wisconsin breast cancer dataset which is open for the public as well as it is being widely researched. The following paper consists of several sections. Section 2 will be discussing the materials and approaches used. Section 3 will be evaluating the implemented algorithms. Section 4 will be visualizing the results produced by different algorithms. Finally, section 5 concludes the entire paper.

II. LITERATURE REVIEW

In machine learning as well as data mining, the classifying technique is known as one of the most crucial and critical tasks that each model should be emphasized to produce optimal results. There was various good research conducted and implemented in several medical datasets in order to assist in classifying breast cancer and many of them have demonstrated great accuracy of classification.

Hiba Asri, Hajar Mousannif, Hassan AI Moatassime and Thomas Noel [2] have conducted research regarding the comparison among SVM, NB, k-NN and C4.5 algorithms in terms of effectiveness and efficiency. All of the models were utilized with the libraries from the WEKA machine learning

environment and implemented in the dataset of Wisconsin Breast Cancer. The experimental results were reported as SVM reached the best outcome with the accuracy of 97.13% which proved that SVM has outperformed NB (95.99%), k-NN (95.27%) and C4.5 (95.13) models with respect to the sensitivity, specification and precision.

The study carried out by Sudhir D. Sawarkar, Ashok A. Ghatol and Amol P. Pande [11] was to prove that the using SVM can cater to the needs in the medical area. In this study, the kernel Adatron model was used with SVM in order to efficiently examine the accuracy of the outcome in the Wisconsin Breast Cancer dataset. The result was shown that SVM achieved a precision rate of 97% that has proven using this model has a higher accuracy rate compared to human manual detection which was 85%.

Another research performed by Alaa M. Elsayad and H. A. Elsalamony [1] compared the performance of decision tree classifiers against RBF-SVM on Wisconsin Breast Cancer Dataset using SPSS Clementine. It has reported C5.0 classifier as the best DT classification model while RBF-SVM classifier as the overall best classification model.

Based on the experimental results, it is found that RBF-SVM has achieved the highest accuracy and perfect sensitivity among other classifiers at 99.98% and 100.00% in the training dataset. On the other hand, RBF-SVM has recorded 98.20% sensitivity while tying the accuracy score with C5.0 at 96.64% in the validation dataset. But RBF-SVM still outperform C5.0 with 99.32% specificity as compared to C5.0 that only managed to achieve specificity score of 95.36%.

With regards to the aforementioned related works, there are a lot of comprehensive studies have been conducted onto breast cancer diagnosis via ML approaches. To date, there are still a lot of undergoing researches that aimed to pinpoint the ideal ML algorithm to be implemented onto actual clinical practices in hope that one day, it can fully support the entire healthcare industry for the benefit of modern society.

III. MATERIALS AND METHOD

A) TOOLS

The building and training of classification models using different ML algorithms is conducted using the CMD.exe Prompt of Anaconda Navigator using Intel ® Core (TM) i7 – 7700HQ with CPU of 2.80GHz. The open source Anaconda Navigator is a distribution of Python and R. It serves a platform for professionals to conduct data science and machine learning tasks by providing wide and easy access to a multitude of packages and libraries.

B) RESOURCES

In this research, the diagnostic version of Wisconsin Breast Cancer Dataset (WBCD) is acquired from the official website of University of California (UCI), Machine Learning Repository. The dataset is created by Dr. William H. Wolberg at the University of Wisconsin Hospital in Madison, Wisconsin. This dataset consisted of 569 instances along with 30 cytological attributes and 1 target attribute. In this research, this dataset is used to evaluate the performance of different classifiers against SVM classifiers.

TABLE I. ATTRIBUTES OF WISCONSIN BREAST CANCER (DIAGNOSTIC) DATASET.

Cytological Attribute	Data Type
id	Numeric
diagnosis	String
radius_mean	Numeric
texture_mean	Numeric
perimeter_mean	Numeric
area_mean	Numeric
smoothness_mean	Numeric
compactness_mean	Numeric
concavity_mean	Numeric
Concave_points_mean	Numeric
symmetry_mean	Numeric
fractal_dimension_mean	Numeric
radius_se	Numeric
texture_se	Numeric
perimeter_se	Numeric
area_se	Numeric
smoothness_se	Numeric
compactness_se	Numeric
concavity_se	Numeric
Concave_points_se	Numeric
symmetry_se	Numeric
fractal_dimension_se	Numeric
radius_worst	Numeric
texture_worst	Numeric
perimeter_worst	Numeric
area_worst	Numeric
smoothness_worst	Numeric
compactness_worst	Numeric
concavity_worst	Numeric
concave_points_worst	Numeric
symmetry_worst	Numeric
fractal_dimension_worst	Numeric
Unnamed:32	Numeric (NaN)

C) DATA SET

The database that used in this research is Wisconsin Prognostic Breast Cancer (WPBC) dataset from UCI machine learning repository. It contains 569 records, where 357 records are Benign tumors and 212 records are Malignant tumors as shown in Table II.

TABLE II. FREQUENCIES PERCENTAGE

Tumor type	Number of instances	Percentage
Benign	357	62.74
Malignant	212	37.26

Even though the distribution of class is unfair, where Benign occupied 62.74% while Malignant occupied 37.26% of dataset, somehow the dataset is rebalanced as data were distributed in wide-ranging and the features were standardized with a mean of '0' and standard deviation of '1'.

D) DATA PRE-PROCESSING

Data pre-processing is conducted onto the dataset. But since only one dataset is used, and all its attributes are having the same data type with no missing values, data cleaning process is less emphasized but is still compulsory to perform. On the other hand, data reduction is placed greater emphasis and is performed via filter approach of feature selection techniques.

This is done in order to reduce the dimensionality of the data by removing all insignificant attributes that are irrelevant in the classification process. Through this approach, 'Unnamed:32' attribute is removed while the 'ID' column is now set to index the dataset. After that, the dataset is divided into independent variables, x that is having all 30 cytological attributes while dependent variable, y consists of the target attribute 'Diagnosis'.

Next, data transformation is also performed to encode all the values of the 'Diagnosis' attribute from 'M' and 'B' into '1' and '0' respectively. By doing this, we transformed all the string values into their binary format for binary classification purpose. Based on the encoded values, it is found that there are a total of 357 instances of '0' (benign cases) and 212 instances of '1' (malignant case). Then, the dataset is partitioned into training and validation datasets following a 80:20 ratio. As for training the dataset, attributes are normalized individually to ensure a more stable performance of classification models. Then, a stratified 10-fold cross validation is also performed by splitting the dataset into 10 folds that are having the same class distribution. Among them, 9 folds are used for training purposes and remaining 1-fold for testing purpose.

Other than encoding the string value into binary format, this research also applies feature scaling in data preprocessing processes. The dataset is undergoing standardization by importing the StandardScaler function in python. This function standardizes features by eliminating the mean and scaling to unit variance. The reason performs standardization is due to the different scale of variables in datasets. The perimeter_mean has values on scale 43.79 - 188.5 while symmetry_mean has values on scale 0.106-0.304. Therefore, standardization is needed to provide common scale and shorten the range of the values of the dataset. Below shows the equation of standardization.

Standardization:

$$z = \frac{x - \mu}{\sigma} \quad (1)$$

with mean:

$$\mu = \frac{1}{N} \sum_{i=1}^{N} (X_1) \quad (2)$$

And standard deviation:

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (X_1 - \mu)^2} \quad (3)$$

Min-Max Scaling:

$$X_{norm} = \frac{x - X_{min}}{X_{max} - X_{min}} \quad (4)$$

IV. ALGORITHM IMPLEMENTATION

In order to classify the observations into benign or malignant categories, a classification approach of supervised learning technique is adopted. During the training phase, models are trained using observations labelled with known target attributes. However, during the validation phase, trained models are used to predict the target attribute of unlabeled observations. As such, the main purpose of using this approach is to build classification models with highly accurate target attribute prediction but also within a short period of time.

A) GAUSSIAN NAÏVE BAYES CLASSIFIER

A Naïve Bayes algorithm defines a simple approach to implement Bayes theorem for classification. It applies probability distribution to allocate class labels to test data by processing of numeric features, which is an effective way as it purely implies the probability theory. Based on Fig. 1, each node has its own parent nodes where all variables are conditionally independent. Hence, a set of variables of joint probability can be calculated by disintegrating it into product of conditional probability distributions on each variable given its parents in the graph.

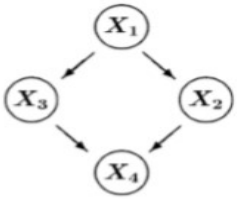

Fig. 1. Simple structure of Bayes network

Equation of joint probability is as follows:

$$P(Y_1, Y_2, \ldots, Y_n) = \prod_{i=1}^{n} P(Y_i, | Parents(Y_i)) \quad (5)$$

Where Parent (Y_i) is the set of parent variables.

Naïve Bayes can be extended to real-values attributes is called Gaussian Naive Bayes which is commonly assume a Gaussian distribution. Gaussian Naïve Bayes uses estimated mean and standard deviation from training data to estimate the distribution of data.

The mean can be calculated as the following equation:

$$Mean(x) = \frac{1}{N} \sum_{i=1}^{N} (X_1) \quad (6)$$

Where n is the number of instance and x are the values of input variable in training data. Standard deviation can be calculated as the following equation:

$$Standard\ Deviation\ (x) = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (X_1 - \mu)^2} \quad (7)$$

Where sqrt() is the function square root, sum() is the function sum, xi is the specific value of x variable, mean(x) is described as above and ^2 is the function square.

To calculate the probabilities of new x value, Gaussian Probability Density Function (PDF) is used.

$$f(x) \triangleq \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (8)$$

Where pdf is the Gaussian PDF, sqrt() is the function square root, mean is mean, std is standard deviation, PI is numeric constant, exp() is numerical constant Euler's number raised to the power and x is the value of input variable.

B) RANDOM FOREST CLASSIFIER

Random Forest Classifier is a supervised ensemble machine learning algorithm developed by Leo Breiman [4] that fits a range of decision tree classifiers on randomly selected data samples, uses averaging to enhance accuracy results and selects the optimal solution via voting. By averaging the result, it can reduce the over-fitting issue. Basically, random forests applicable for image classification and feature selection. Random Forest applies Bootstrap Aggregating (Bagging) training algorithm, which combines multiple predictions together to produce more accurate predictions. For instance, 3 mean values 2.3, 3.4 and 4.5 obtained from resamples, Bagging would take the average mean value which is 3.4 as the estimated mean.

Following steps briefly describe the flow of Random Forest algorithm:

Step 1 - First, select random samples from a given dataset

Step 2 - Construction of decision tree for each sample happens in this stage. Prediction result acquired from each decision tree.

Step 3 - Current step will perform voting for every predicted result.

Step 4 - Lastly, the most votes is selected as the final prediction

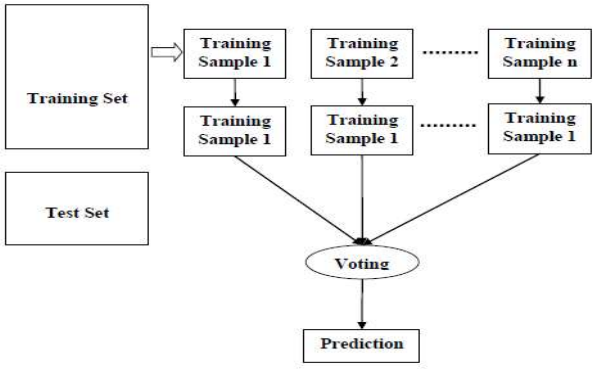

Fig. 2. Random forest algorithm flows

There are some hyperparameters of sklearn built-in random forest function that are strongly impacting the prediction results. It allows the model to be built faster and boosts the predictive power of the model.

n_estimators: Represent the number of trees in the algorithm, default is 10.

Higher number of trees indicates more splitting and taking maximum of voting increases the performance, but it consumes more time for computing.

max_features: Maximum number of features to consider when looking for the best split.

By default, max_features = 'auto'. If integer, then consider max_features. If float, then int(max_features * n_features). If 'auto' and 'sqrt', then max_features=sqrt(n_features). If 'log2', then max_features=log2(n_features). If none, then max_features=n_features.

min_samples_leaf: It determines the minimum number of samples required to be at a leaf node, default is 1.

This hyperparameter helps to smooth the model, exclusively in regression.

criterion: To measure the quality of a split, default is 'Gini'.

Gini are standard metrics to compute "impurity" while Entropy computes "information gain".

$$Gini = 1 - \sum_{j=1}^{c} p_j^2 \quad (9)$$

$$Entropy = -\sum_{j=1}^{c} p_j \log p_j \quad (10)$$

As shown in Fig. 2, Gini uses squared proportion of classes in the equation while Entropy involves logarithmic function. Thus, Gini impurity is calculated with less computation. Gini is usually used for CART (Classification and Regression Tree) and produces small values indicates less impurity. Same goes to Entropy, smaller value is better. That makes the difference between the parent node's entropy larger.

random_state: Manipulate both the randomness of the Bagging of the samples used when constructing trees. Feeding a value like 0, 1 and 42 into random_state to ensure the splits that are generated are reproducible. Without fixed value, the random values of the train and test datasets would be different each time.

n_jobs: Represent the number of jobs to run in parallel. This hyperparameter informs the engine of the usage of processors that are allowed to be used. If the value = 1, it can only allow one processor. However, the value of "-1" indicates no limit.

C) DECISION TREE CLASSIFIER

Decision Tree (DT) serves as one of the most powerful and widely used ML algorithms in terms of classification and prediction problems. The main concept of DT is to split the data using binary recursive partitioning based on a specific splitting criterion. This mainly results in a tree-like structure that consists of decision nodes, branches, and leaf nodes.

Firstly, decision nodes including the root node are having splits, which they will further branch out into multiple decision nodes to test for a particular attribute. Secondly, branches that interconnect the nodes in turn denotes different outcomes of an attribute test. Thirdly, leaf node that act as the terminal node will represent the final predicted outcome of a test.

For classification problems, decision tree classifier utilized a divide-and-conquer approach to subset the data samples continuously. This is done until the data has achieved a certain degree of homogeneity or a stopping criterion is met. The classification mechanism of a decision tree classifier is described as follow:

Take the entire training dataset as the root node of the tree.

1) Create branches for each possible outcome of the corresponding attribute test.
2) If: All observations in the current node are having the same class label, terminate the recursion process, and make this node as the leaf node.
3) Else: Continue to split data samples of the attribute with the highest information gain or lowest Gini index and make this node as a decision node.

4) Repeat Step 4 until the condition mentioned in Step 3 is satisfied.
5) But if the predetermined maximum depth of the tree is reached before Step 3 is achieved, force terminate the recursion process to prevent overfitting issues.

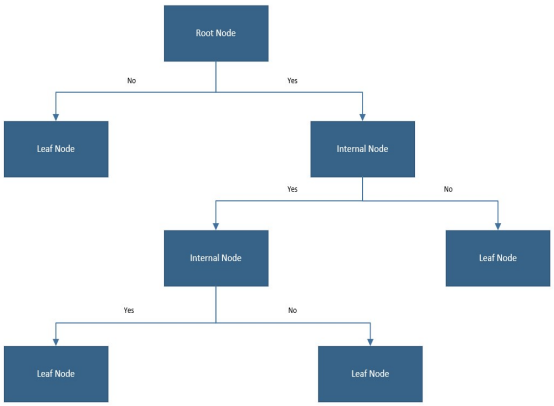

Fig. 3. Generic decision Tree Algorithm.

To determine the splitting criterion of the decision tree classifier, Gini Index and Entropy of Information Gain are calculated. They served as the standard splitting measures that can help to identify the best attribute to be used as the root node to ensure best split at every subsequent node that in turn lead to a better classification.

Gini Index is a metric that is used to measure the probability of a wrongly classified attribute when it is chosen at random. Its range of value is between 0 and 1, where 0 indicates perfect purity, where all data samples belong to the same class whereas 1 denotes random distribution of data samples throughout all available classes. In binary classification, Gini Index of 0.5 also represents maximum impurity. Thus, the lowest value of the Gini Index is often chosen as the root node when building a decision tree. To calculate the Gini Index, take 1 minus the sum of squared probabilities of each class. The formula for Gini Index is shown below:

$$I_G(n) = 1 - \sum_{i=1}^{J}(pi)^2 \quad (11)$$

On the other hand, entropy of Information Gain is the measurement of the degree of uncertainty in the data points. Similar to Gini Index, value of 0 in entropy specifies perfect purity. When entropy is frequently expressed in between 0 and 1, the value of maximum impurity can be higher than 1, depending on the number of existing classes in the dataset. In terms of Information Gain, it provides an approach to quantify the degree of uncertainty in attributes. With this, Information Gain can minimize the level of entropy from the root node to the leaf nodes of a decision tree. The entropy of Information Gain is in turn obtained by multiplying the class' probabilities with the log base 2 of the same probabilities. The formula for entropy is shown below:

$$E(S) = \sum_{i=1}^{c} p_i \log_2 p_i \quad (12)$$

While both the impurity measures often yield the similar results, entropy is slower to be computed as compared to Gini Index. This is because the calculation of entropy involved the logarithmic function which is computationally heavier. Furthermore, Gini Index regulates misclassification in large distribution better while entropy works well in reducing the uncertainty of a smaller distribution.

Nevertheless, decision tree algorithm still plays an important role in addressing the classification problems especially in the context of breast cancer diagnosis. This is because it still possessed several advantages over other classification algorithms as listed below:

1) Simple & inexpensive to build.
2) Quick classification of unknown records.
3) High accuracy relative to other algorithms.
4) Easy-to-understand algorithm structure of small-sized trees.
5) Ignore and eliminate insignificant features of the dataset.

D) K-NEAREST NEIGHBORS CLASSIFER

In machine learning, a k-Nearest Algorithm is known as one of the most utilized algorithms with respect to efficiency and accuracy. Since the training phase is not needed in this algorithm, therefore, its learning approach is based on the input instances and the efficiency of the result will not be affected if there is additional data being added into the experiment. The neighbors are decided by the type of distance that divides the data to new elements in order to begin classifying.

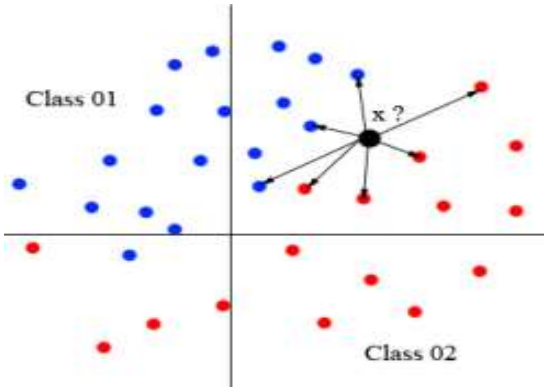

Fig. 4. K-Nearest neighbors method

This algorithm normally functions depending on the parameter of k value which denotes the number of neighbors to be chosen to distribute the class for the new element as well as the form of distance utilized. [8]

1) Distance

From the perspective of mathematics, the distance aims to formalize the concept of distance that is the length among two different points and it is able to provide assistance in grouping the related data and isolate data that do not resemble.

Distance of Cityblock (One-Distance)

$$\boldsymbol{d}(X_i, X_j) = \sum_{r=1}^{n} |X_{ir} - X_{jr}| \quad (13)$$

This type of distance is also known as the Manhattan distance and it is equated to the one – norm, for two different vectors (xir, xjr). The formula is described as the sum of absolute differences.

Distance of Euclidean (Two-Distance)

$$\boldsymbol{d}(X_i, X_j) = \sqrt{\sum_{r=1}^{n}(X_{ir} - X_{jr})} \quad (14)$$

This distance is known as the most universal distance which is placed between two vectors (xir, xjr). In addition, the Euclidean distance is a special case with the metric of Minkowski when p equals to two.

Distance of Minkowski (P-Distance)

$$\boldsymbol{d}(X_i, X_j) = \sqrt[p]{\sum_{r=1}^{n} | X_{ir} - X_{jr}|^p} \quad (15)$$

Minkowski distance is one of the most often used distances where p equals to one, two to infinity.

2) Parameter of K

The value of k is usually determined by the experimenters which depend on the data as well. Despite the boundary of classes will be less clear, the result of noise while classifying is minimized when the number of k is greater. A variety of heuristic methods such as cross-validation could be chosen due to a good option of the number of k. In this experiment, the authors decided to opt for the value of k which able to reduce the error of classifying. In the binary classification, an odd number of k is recommended in order to prevent equal votes.

E) SUPPORT VECTOR MACHINE CLASSIFER

"Support Vector Machine" (SVM) is a supervised machine learning algorithm that was developed by Vapnik and his co-worker, powerful for solving regression and classification problems [3] The goal of the SVM is to find a hyperplane in a N-dimensional space that classifies the data point distinctly. Consequently, classification performed by finding the hyperplane that differentiates the two classes.

Different from Logistic Regression (LR), SVM maximize the distance between the decision boundary and all instances. By implementing the SVM algorithm, we are able to find the best hyperplane in more than two dimensions in favor of separate space into classes. The distance of the vectors from the hyperplane, which is a separation of a line to the nearest class points is called the margin. A good margin is shown by the same distance from all sector vectors with the maximum margin hyperplane. Nonetheless, bad margin shown by the support vectors is either very close to class -1 or class +1. The objective of the SVM algorithm is to find a plane that has maximum margin. Maximizing the margin distance offers sufficient reinforcement so that potential data points can be identified with better conviction.

1) Hyperplane and Support Vector

Hyperplanes are decision boundaries that help analyze the data points. Data points that land on either side of the hyperplane may be assigned to different classes. Furthermore, the dimensions of the hyperplane depend on the number of features. As the number of input features increase, it can become a two-dimensional plane. However, if the number of input features is 2, the hyperplane is only displayed as a line.

Support vectors are data points relative to the hyperplane, which impact the hyperplane 's direction and orientation. The presence of support vectors enables the function of maximizing the margin of the classifier.

2) Cost Function and Gradient

Hinge loss is the loss function that helps maximize the margin, it maximizes the margin between the data points and the hyperplane. Equation below shows the hinge loss model for linear classifier proposed by Moore and DeNero.[9]

$$\ell(y) = max\,(0,1 + \max_{y \neq t} w_y x - w_t x) \quad (16)$$

Somehow, Weston and Watkins proposed a similar equation, but it included a sum rather than a max. [16]

$$\ell(y) = \sum_{y \neq t} max\,(0,1 + w_y x - w_t x) \quad (17)$$

In the SVM algorithm, regularization parameter is added into the cost function to balance the margin maximization and loss.

$$L(w) = \sum_{i=1} \underbrace{max\,(0,1 - y_i\,[w^T x_i + b]}_{\text{Loss function}} + \underbrace{\lambda\,||\,w||_2^2}_{\text{regularization}} \quad (18)$$

The application of hinge loss is to castigate misclassification as minimizing cost functions will result in a lower error between the actual values and predicted values. The insertion of regularization term has prevented overfitting by penalizing large coefficients in the solution vector. If the expected value and the real value are of the same form, the cost is 0. If cost is other than 0, calculation of loss value proceeds. When a data point is at the classifier 's margin, the hinge-loss is exactly zero, we have:

$$max\,(0,1 - y_i\,[w^T x_i + b]) = 0 \quad (19)$$

$$\Rightarrow y_i\,[w^T x_i + b] = 1$$

Therefore, we have:

$$x^+ \cdot \frac{w}{||\,w\,||} - x^- \cdot \frac{w}{||\,w\,||} = \frac{1-b}{||\,w\,||} - \frac{-b-1}{||\,w\,||} = \frac{2}{||\,w\,||} \quad (20)$$

Within the loss function, we can derive partial derivatives with respect to the weights to find the gradients. Weights also can be updated by using the gradients.

$$\frac{\delta}{\delta w_k}\,\lambda\,||\,w\,||^2 = 2\lambda w_k \quad (21)$$

$$\frac{\delta}{\delta w_k}\,(1 - y_i < x_i, w >)_+ = \begin{cases} 0, & if\ y_i < x_i, w > \geq 1 \\ -y_i x_{ik}, & else \end{cases} \quad (22)$$

Regularization parameter works to perform gradient update in both situations either no misclassification or having misclassification.

$$w = w - \propto \cdot\,(2\lambda w) \quad (23)$$

Equation 23 shows gradient update when no misclassification occurs.

$$w = w + \propto \cdot (y_i \cdot x_i - 2\lambda w) \quad (24)$$

Equation 24 shows gradient update when misclassification occurs.

3) Tuning Parameter C:

A regularization parameter, which controls the tradeoff between smooth decision boundaries and minimizing the norm of the weights. The C parameter value is 1.0 by default. The C parameter determines how great your desire to avoid misclassifying each training example. The larger value of C, the optimization will not neglect the outlier and thus smaller margin produced. Inversely, a smaller value of C will cause the optimizer to search for a larger margin hyperplane, even if that hyperplane is misclassified and having outliers. A large value of C also indicates receiving more training points correctly.

4) Gamma:

The gamma parameter defines how far the influence of a single training example reaches. Low value indicates far reach for every point and conversely high value indicates close reach for every point. The model 's behavior is somewhat sensible to the gamma parameter. [13] If gamma value is too high, then the support vectors itself contain the radius of the field of influence of the support vectors and no amount of regularization with C would be able to avoid overfitting. On the contrary, a small value of gamma will not be able to capture the complexity or "shape" of the data and the model is too constrained.

To draw a conclusion, as we use the SVM algorithm for breast cancer diagnosis, we emphasize the accuracy of the learning algorithm. Nonetheless, the C parameter is using default value = 1.0, as increased value still provides the same accuracy result. The gamma value is using 'auto' as it takes the value of 1/features, in our case would be 1/32 or 0.03125. This gives us a 98.25% accuracy and does not restrict the classification areas.

V. RESULTS

To evaluate the performance of classifiers, a few metrices are used in the result, which are precision, recall, F1-score, and accuracy. A confusion matrix is constructed to visualize the classification results achieved by classifiers.

A) Classification Reports

Classification Report: Gaussian Naïve Bayes Classifier Overall Accuracy: 93.86%

Based on Table II, presented the performance metrices for Gaussian Naïve Bayes Classifier. The precision and recall for 67 'benign' are 94% and 96% while the f1_score which is the mean of precision and recall is 95%. For 47 'malignant' the precision and recall are 93% and 91% where the f1_score is 92% where it is slightly lower compared to benign. Somehow, the macro average and weighted average score for every performance metric are remain at 94%.

TABLE II. Measurements of Performance Metric for Gaussian Naïve Bayes Classifier

Label(s)	Precision	Recall	F1-score	Support
0	0.94	0.96	0.95	67
1	0.93	0.91	0.92	47
Macro Avg.	0.94	0.94	0.94	114
Weighted Avg.	0.94	0.94	0.94	114

Classification Report: Random Forest Classifier

Overall Accuracy: 96.49%

In this paper, Random Forest Classifier is tested with **n_estimators = 180, criterion = "entropy" and random_state = 0**. Entropy criterion is implemented for calculating the information gain rather than the impurity. **random_state** fixed at value 0 to ensure obtained reproducible splits result. **n_estimators** are fixed at the range between 100-300, one can get better accuracy results and avoid duplicating data with higher number splits presence.

Table III demonstrated the performance of Random Forest Classifier in different metrics. For 67 predicted 'Benign', it shows 96% of precision and 99% of recall indicates the failed prediction rate is extremely low. Aforementioned, 47 predicted 'Malignant' consist of 98% of precision and 94% recall indicates the algorithm has made slightly wrong prediction on FN = False Negative values which is predicted as malignant but actual result is benign. By hook or crook, the macro average and weighted average have only 1% different based on these three performance criteria.

TABLE III. Measurements of Performance Metric for Random Forest Classifier

Label(s)	Precision	Recall	F1-score	Support
0	0.96	0.99	0.97	67
1	0.98	0.94	0.96	47
Macro Avg.	0.97	0.96	0.96	114
Weighted Avg.	0.97	0.96	0.96	114

Classification Report: Decision Tree Classifier
DT classifier: Entropy
Overall Accuracy: 93.86%

Table IV displayed the score for each performance metrics achieved by the decision tree classifier with Entropy as its split criterion. It has achieved a 94% PPV and 96% TP rate while having an impressive harmonic mean between precision-and-recall at 95% in the classification of 67 'benign' observations. On the other hand, it achieved a precision score of 93% and a slightly lower recall score of 91% with corresponding f1-score standing at 92% in the classification of 47 'malignant' observations. In terms of macro and weighted average, the score remained unchanged at 94% for every performance metric computed.

TABLE IV. Measurements of performance metrics for DT classifier: Entropy

Label(s)	Precision	Recall	F1-score	Support
0	0.94	0.96	0.95	67
1	0.93	0.91	0.92	47
Macro Avg.	0.94	0.94	0.94	114
Weighted Avg.	0.94	0.94	0.94	114

DT Classifier: Gini Index

Overall Accuracy: 93.86%

Table V described the performance of the decision tree classifier that used Gini Index as its split criterion evaluated using different metrics. In classifying 67 'benign' observations, it recorded outstanding precision and f1-score both standing at 95%. In contrast, the performance is slightly lacking in the classification of 'malignant' observations as the precision and f1-score has only recorded a value of 92% and 93% respectively. Nevertheless, the recall is maintained at 94% for both classifications. Similar to the previous results, the score for macro average and weighted average for each label remained the same at 94%.

TABLE V. Measurements of Performance Metrics for DT classifier: Gini Index.

Label(s)	Precision	Recall	F1-score	Support
0	0.95	0.94	0.95	67
1	0.92	0.94	0.93	47
Macro Avg.	0.94	0.94	0.94	114
Weighted Avg.	0.94	0.94	0.94	114

Classification Report: K-Nearest Neighbors
Overall Accuracy: 96%
Distance: Euclidean

Based on the Table VI, it has outlined every performance metrics produced by using the model of K-Nearest Neighbors with Euclidean distance. A total of 67 labeled as "benign" having the rate of precision and recall as 94% and 100% respectively while follow by the f1_score is 97%. Apart from that, there are 47 labeled "malignant" with the precision of 100% and recall of 91% as well as with a slightly lower percentage of f1_score which is 96% compared to benign observations. Furthermore, for the average of macro and weighted, they both are having the same percentage of precision which is 97% while the recall and f1_score is sharing the same percentage of 96%.

TABLE VI. MEASUREMENTS OF PERFORMANCE METRICS FOR K-NEAREST NEIGHBORS: EUCLIDEAN DISTANCE

Label(s)	Precision	Recall	F1-score	Support
0	0.94	1.00	0.97	67
1	1.00	0.91	0.96	47
Macro Avg.	0.97	0.96	0.96	114
Weighted Avg.	0.97	0.96	0.96	114

Classification Report: K-Nearest Neighbors
Overall Accuracy: 96%
Distance: Manhattan

Based on Table VII, it has described that every performance metrics produced by using the model of K-Nearest Neighbors with Manhattan distance. The percentage of precision and recall are 94% and 99% correspondingly for the 67 observations of "benign" and the average of precision and recall which is known as f1_score is valued 96% whereas for the total of 47 "malignant" with a percentage of precision 98% and 91% for the recall as well as the f1_score is 95%. Furthermore, the percentage of macro average for precision is 96% and both recall and f1_score is 95% whereas the precision, recall and f1_score of weighted average are sharing 96%.

TABLE VII. Measurements of Performance Metrics for K-Nearest Neighbors: Manhattan Distance

Label(s)	Precision	Recall	F1-score	Support
0	0.94	0.99	0.96	67
1	0.98	0.91	0.95	47
Macro Avg.	0.96	0.95	0.95	114
Weighted Avg.	0.96	0.96	0.96	114

Classification Report: SVM Classifier
Overall Accuracy: 98.49%

Based on Table VII, presented the performance metrices for SVM Classifier. The precision and recall for 67 'benign' are 97% and 100% while the f1_score which is the mean of precision and recall is 99%.

TABLE VIII. Measurements of Performance Metric for SVM Classifier

Label(s)	Precision	Recall	F1-score	Support
0	0.97	1.00	0.99	67
1	1.00	0.96	0.98	47
Macro Avg.	0.99	0.98	0.98	114
Weighted Avg.	0.98	0.98	0.98	114

B) Confusion Matrix

For 47 'malignant' the precision and recall are 100% and 96% where the f1_score is 98% where it is slightly lower compared to benign. Somehow, the macro average and weighted average score for every performance metric are remain at 98%.

Based on the confusion matrix on Fig. 5, it is shown that the Gaussian Naïve Bayes Classifier has correctly classified a total of 107 observations, which 64 of them are observations from the '0' category which is the benign whereas the remaining 43 comes from the '1' category which is malignant. In total, there are only 7 misclassified observations which lead to 6.14% of misclassification rate, that still fall within the acceptable range.

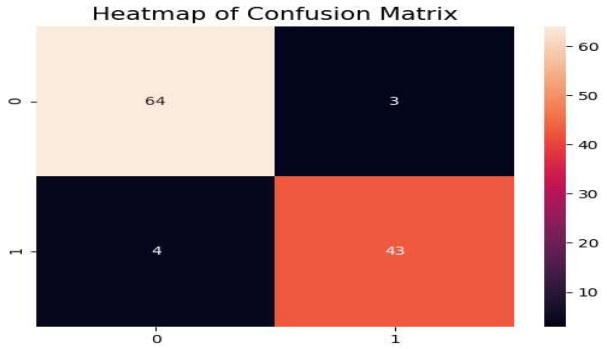

Fig. 5. Confusion Matrix of Gaussian Naïve Bayes Classification

According to the confusion matrix on Fig. 6, it is shown that the Random Forest Classifier has correctly classified a total of 110 observations, which 66 of them from the TN (True Negative) group, the success negative prediction and 44 of them from the TN (True Positive) group, the success positive prediction. In total, there are 4 misclassified observations which lead to 3.50% of misclassification rate, that still fall within the acceptable range.

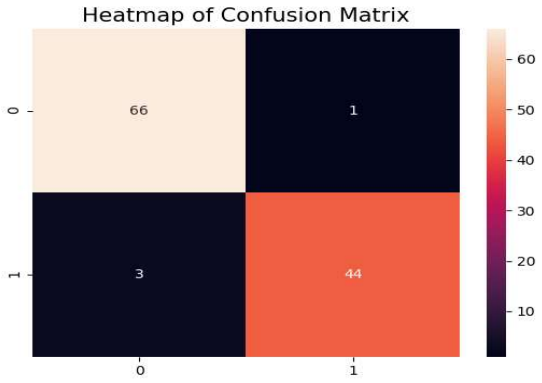

Fig. 6. Confusion matrix of random forest classification

Based on the confusion matrix on Fig. 7, it is shown that the decision tree classifier has correctly classify a total of 107 observations, which 43 of them are observations from 'malignant' category whereas the remaining 64 comes from the 'benign' category. In total, there are only 7 misclassified observations which lead to 6.14% of misclassification rate, that still fall within the acceptable range.

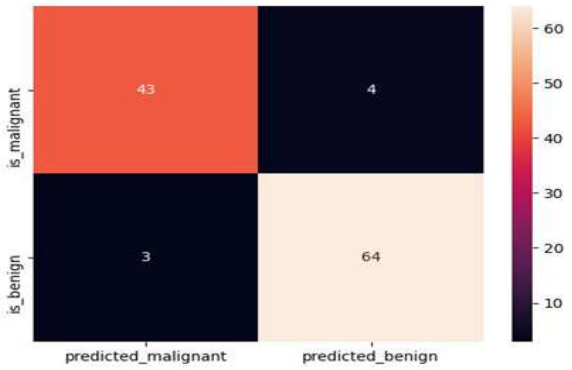

Fig. 7. Confusion Matrix of DT classifier: Entropy visualized through heatmap.

According to the confusion matrix as shown in Fig. 8, the number of correct classification is 107 and the number of incorrect classification is 7 which translate to 6.14% of misclassification rate, which is similar to previous decision tree classifier. And, among the 107 correctly classified observations, 44 of them belonged to the 'malignant' group and 63 were under the 'benign' group.

According to the confusion matrix accomplished by K-Nearest Neighbors using Euclidean distance in Fig. 9, the model has precisely classified a total of 110 correct observations which 43 of them are malignant and 67 are benign whereas there are just 4 of observations were classified wrongly.

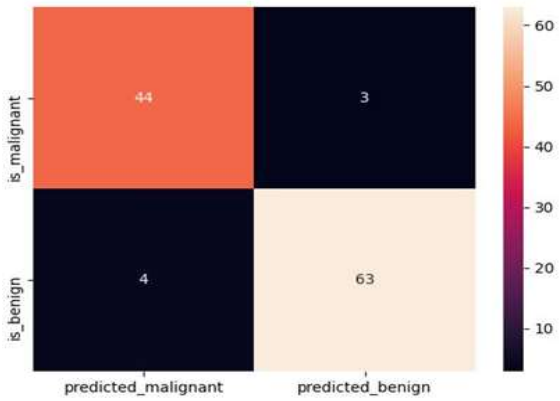

Fig. 8. Confusion matrix of DT classifier: Gini Index visualized through heatmap.

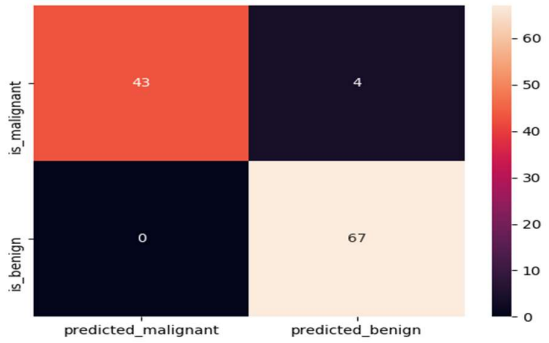

Fig. 9. Confusion Matrix of K-Nearest Neighbors: Distance of Euclidean

Based on the Fig. 10 confusion matrix achieved by K-Nearest Neighbors using Manhattan distance, the algorithm has classified correctly for a number of 109 which are 43 are malignant and 66 are benign whereas there is a total of 5 observations were incorrectly classified which are 1 is malignant and 4 are benign.

Based on the confusion matrix in Fig. 11, it is shown that the Kernel SVM Classifier has correctly classified a total of 112 observations, which 67 of them are observations from the '0' category which is the benign whereas the remaining 45 comes from the '1' category which is malignant. In total, there are only 2 misclassified observations which lead to the accuracy of 98.25% and 1.75% of misclassification rate, where it is highly precise and accurate.

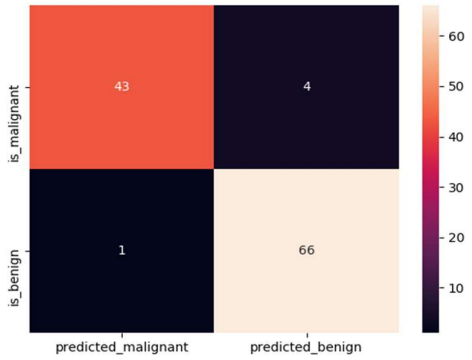

Fig. 10. Confusion matrix of K-Nearest Neighbors: distance of manhattan

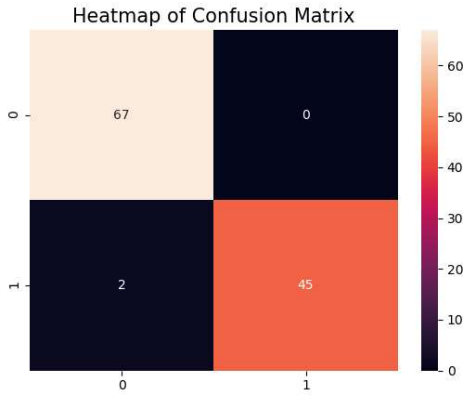

Fig. 11. Confusion matrix of Kernel SVM classification

VI. COMPARATIVE RESULT

Table 10 show the results for dataset of breast cancer using different algorithms which are Naïve Bayes, Random Forest Classifier, Decision Tree Classifier, K-nearest Neighbors Classifier and Support Vector Machine.

TABLE IX. RESULTS OF CLASSIFIERS ON THE DATASET

Classifiers	Accuracy	F1-score
Gaussian Naïve Bayes Classifier	0.9386	0.9247
Random Forest Classifier	0.9649	0.9565
Decision Tree Classifier (Entropy)	0.9386	0.9247
K-Nearest Neighbors Classifier (Euclidean)	0.9649	0.9556
Support Vector Machine Classifier	0.9825	0.9783

Accuracy and F1-score are used as the metrics for statistical analysis of the datasets. Accuracy measures all correctly identifies cases when all the classes are equally important. F1- score measure the mean of Precision and Recall as well as provide a better result of incorrectly classified cases than the accuracy metric.

The metrics used are as follows:

$$Accuracy = \frac{True\ Positive + True\ Negative}{(True\ Positive + False\ Positive + True\ Negative + False\ Nagative)} \quad (25)$$

$$F1 - score = \frac{Recall^{-1} + Precision^{-1}}{2} = 2 * \frac{(Precision * Recall)}{(Precision + Recall)} \quad (26)$$

Based on the accuracy on Table, Support Vector Machine Classifier performed better among all classifier by reaching an accuracy of 98.25% while the Gaussian Naïve Bayes Classifier and Decision Tree Classifier both having the worst accuracy with 93.86%. Random Forest Classifier and K-nearest Neighbors Classifier having the same accuracy which is 96.49%, which is 1.76% lower than Random Forest Classifier but 2.63% higher than Gaussian Naïve Bayes Classifier and Decision Tree Classifier. In term of F1-score, Support Vector Machine achieved a highest score among all classifier by reaching 97.83% while the Gaussian Naïve Bayes Classifier and Decision Tree Classifier both achieve a lowest F1-score which is 92.47%. Random Forest Classifier achieved 2.18% lower F1-score than Support Vector Machine Classifier which is 95.65%. Lastly, the K-Nearest Neighbors Classifier achieved a slightly lower F1-score than Random Forest Classifier which is 95.56%. By comparing all classifiers with accuracy and f1-score, Support Vector Machine Classifier performed the best among all by achieving both highest accuracy and f1-score. Gaussian Naïve Bayes and Decision Tree Classifier both performed the worst by having lowest accuracy and f1-score among all classifier.

VII. CONCLUSION

In this paper, we studied five machines learning classifier for the classification of breast cancer. The dataset that have been used in this study is Wisconsin Prognostic Breast Cancer dataset from UCL learning repository. Our focus is to study on the algorithm and modify one of the algorithms to improve it accuracy. The algorithm that have been modified is Support Vector Machine Classifier. After the pre-processing of data, five predictive models were implemented. Data were encoded into binary where 1 represent as Malignant and 0 represent Benign. Ten-cross validation is used to measure the accuracy. Precision, recall, accuracy, and f1-score are used as metrics to measure the performance of the classifier. Support Vector Machine Classifier is the best algorithm for classification of breast cancer among all the classifiers as it achieved 98.25% of accuracy which is very high. While the algorithm which achieved the least performance are Gaussian Naïve Bayes Classifier and Decision Tree Classifier where both achieved the same accuracy which is 93.86%.

ACKNOWLEDGEMENTS

Deep appreciation is given to the authors' mentor, Mr. Zailan Arabee Bin Abdul Salam (Senior Lecturer, Asia Pacific University of Technology and Innovation), and the dataset creators, Dr. William H. Wolberg (General Surgery Dept, University of Wisconsin), W.Nick Street (Computer Science Dept, University of Wisconsin), and Olvi L. Mangasarian (General Surgery Dept, University of Wisconsin) towards this study.

REFERENCES

[1] A. Elsayad, "Diagnosis of Breast Cancer using Decision Tree Models and SVM," vol.83, pp. 19-29, December 2013.

[2] Asri, H., Mousannif, H., Moatassime, H. and Noel, T. (2010) Using Machine Learning Algorithms for Breast Cancer Risk Prediction and Diagnosis. *Procedia Computer Science*, 83, pp.1064-1069.

[3] Boser, B., Guyon, I. and Vapnik, V., 1992. A training algorithm for optimal margin classifiers. Proceedings of the fifth annual workshop on Computational learning theory - COLT '92,.

[4] Breiman L: Random forests. Machine Learning 2001, 45:5-32.

[5] H. Palanisamy and P. Sampath, "Performance Analysis Of Breast Cancer Classification Using Decision Tree Classifiers," vol.2, pp. 19-25, March 2017.

[6] Jhajharia, S., Verma, S. and Kumar, R., 2016. Predictive Analytics for Breast Cancer Survivability. *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies - ICTCS '16*.

[7] Kaklamanis, M. and Filippakis, M., 2019. A comparative survey of machine learning classification algorithms for breast cancer detection. *Proceedings of the 23rd Pan-Hellenic Conference on Informatics - PCI '19*.

[8] Medjahed, S.A., Saadi, T.A. and Benyettou, A. (2013) Breast Cancer Diagnosis by using k-Nearest Neighbor with Different Distances and

Classification Rules. *International Journal of Computer Applications*, 62(1), pp.1-5.

[9] Moore, R. and DeNero, J., 2020. [online] Isca-speech.org. Available at: <https://www.isca-speech.org/archive/mlslp_2011/papers/ml11_001.pdf> [Accessed 21 August 2020].

[10] Saoud, H., Ghadi, A., Ghailani, M. and Abdelhakim, B., 2018. Application of Data Mining Classification Algorithms for Breast Cancer Diagnosis. *Proceedings of the 3rd International Conference on Smart City Applications - SCA '18*,.

[11] Sawarkar, S.D., Ghatol, A.A. and Pande, A.P. (2006) Neural Network Aided Breast Cancer Detection and Diagnosis Using Support Vector Machine. *7th WSEAS International Conference on Neural Networks*, June, pp. 158-163.

[12] T. Mathew, "Simple And Ensemble Decision Tree Classifier Based Detection Of Breast Cancer," vol.8, pp. 1628-1637, November 2019.

[13] Weston, Jason & Watkins, Christopher. (1999). Support Vector Machines for Multi-Class Pattern Recognition. Proc of the 7th European Sympo-sium On Artificial Neural Networks. 219-224.