

Machine Learning Approach to Classify Breast Tissues: A Case Study Using Six-classed Breast Tissue Data

S. Santharooban^{1,*} and S. P. Abeysundara²

¹Faculty of Health-Care Sciences, Eastern University, Sri Lanka

²Department Statistics and Computer Science, Faculty of Science, University of Peradeniya, Sri Lanka

*Corresponding author: santharoobans@esn.ac.lk

Received: 22nd October 2022/ Revised: 05th December 2022/ Published: 31st December 2022

©IAppstat-SL2022

ABSTRACT

The present study investigates the effectiveness of six Machine Learning (ML) algorithms in classifying the breast tissue dataset generated using the electrical impedance spectroscopy method. This study used the breast tissue dataset available at the UCI machine learning repository, consisting of 106 spectral records with ten variables. The data were partitioned into train and test datasets. Sixty six percentage of data was allocated for the train dataset and balance for the test dataset. Six ML algorithms were tested for effectiveness using accuracy, Cohen's Kappa, sensitivity and specificity. The results revealed that the backpropagation algorithm (BPN) produced the highest accuracy and Kappa compared to other machine learning algorithms in classifying the six-classed breast tissue dataset. Both Support Vector Machine (SVM) and K-Nearest Neighbors (KNN) produced the second-highest accuracy and Kappa. The C5.0 decision tree algorithm takes the third level. The fourth and fifth levels of accuracy are Probabilistic Neural Network (PNN) and Learning Vector Quantization (LVQ), respectively. The sensitivity of all classes by the classification of BPN was more than eighty percentage, which is higher than other machine learning algorithms. The specificity of all classes predicted by BPN was more than ninety six percentage and was comparatively at the highest level than other machine learning algorithms. Therefore, the study concludes that the backpropagation algorithm will effectively classify the six classed breast tissue data.

Keywords: Support Vector Machines, Backpropagation algorithm, Probabilistic neural network, Learning Vector Quantization, K-Nearest Neighbors cluster analysis and C5.0 decision tree approach

1 Introduction

In the context of the increasing incidence of breast cancer, it is paramount to develop a decision support system to help pathologists enable the early detection of cancerous tissue. Numerous machine-learning techniques have been deployed to classify cancer tissue from that normal tissue. However, the efficiency of classification varies among different machine-learning techniques. One of the factors that influence the classification's efficiency is the number of classes to be classified. The following literature pieces of evidence have been organized in a manner increasing the number of classes and the techniques which were employed to classify them.

A study on breast fine needle tumour classification compared different artificial neural networks and support vector machines (George et al., 2012). The study used ANN techniques such as multi-layer perceptron (MLP) using the backpropagation algorithm, probabilistic neural network (PNN), and learning vector quantization (LVQ) in addition to the Support vector machine. The study used 10-fold cross-validation and the performance was compared based on the resulted error rate, correct rate, sensitivity, and specificity, using different datasets containing two classes such as benign and malignant and concluded that predictive abilities are good for probabilistic neural network, followed in order by support vector machine, learning vector quantization and multilayer perceptron for all dataset.

Some of the studies used neural networks to analyze the histological images directly. Araujo et al. (2017) used histological images to classify four kinds of breast tissues. The authors used both Convolutional Neural Network (CNN) classifier and SVM classifier. The article reported that the original image is divided into twelve contiguous non-overlapping patches. The patch class probability is computed using the patch-wise trained CNN and CNN+SVM classifiers. Both CNN and SVM classifiers achieve comparable results. It resulted in 77.8% accuracy for four classes and 83.3% of carcinoma and non-carcinoma achieved. The sensitivity of the cancer cases was 95.6%.

Another study used mammographic images for classification. That study aimed to evaluate the application of different breast tissue density segmentation strategies before the extraction of features in mammograms (Oliver et al., 2006). The study used a Bayesian classifier, which combines k-Nearest Neighbors and the C4.5 decision tree. There were five classes and 113 features for each class in the data set. The authors used Cohen's Kappa statistics to compare different segmentation strategies (Oliver et al., 2006).

According to Wu and Ng (2007), the Normalized weighted average algorithm for the unbiased linear fusion in multiple classifier systems works effectively to improve classification accuracy and achieve confident performance in terms of reliability index. This particular study also used the previously mentioned dataset and used the relative index (RI) to indicate the certainty level of the ensemble results. The same dataset of the previous study was used by another study conducted by Helwan et al. (2017). In this study, the

author used the backpropagation learning algorithm (BPNN) and radial basis function network (RBFN) and finally concluded that RBFN outperforms the classifying breast tissue.

In the present study, the methods used in literature to classify the two classes and five classes are selected to classify the six classed data. As such, support vector machines (SVM), the backpropagation algorithm (BPN), probabilistic neural network (PNN), learning vector quantization (LVQ) and k-Nearest Neighbors (KNN) were selected, and instead of the C4.5 decision tree, its advanced version C5.0 decision tree was selected to analyze six class data.

The present study used the dataset generated by Jossinet (1996). The study used electrochemical impedance spectroscopy (EIS) to measure six breast tissue impediments. The impedance measurement at a different frequency of an element will give valuable data about that element (Lvovich, 2014). This is the basics of EIS, and this is also used in living cells as well. The EIS is used to differentiate cancerous tissue from that normal tissue as the impedivity of tissue will be determined by dimension, internal structure, and arrangement of cells (Walker et al., 2000). This dataset has been used by a few studies to classify breast tissue. A study used the above dataset to classify six classes of breast tissue using applied linear discriminant analysis and achieved an overall classification efficiency of around 92% by hierarchical method (Estrela da Silva et al., 2000). Another study conducted by Olmez et al. (2021) used this dataset. This study used Probabilistic Neural Network (PNN) to classify the tissue types and the results were compared with the results of Multilayer Neural Network (MLNN) and Learning Vector Quantization (LVQ). The best average was obtained for PNN with 98.1%, followed by MLNN with 95.24% which is followed by LVQ with 92.38%. Another study used the same dataset for the classification of breast tissues using Naïve Bayes, Support vector machines, Radial basis neural networks, Decision trees J48 and simple CART algorithms. The study found that SVM-RBF Kernel outperforms other classifiers with respect to the accuracy, sensitivity, specificity and precision for both binary and multiclass datasets (Aruna et al., 2011).

2 Methodology

The study used a dataset generated by Jossinet (1996). The dataset is available at the UCI Machine Learning Repository (Jossinet, 2010). The data set consisted of 106 spectra recorded in breast tissue samples from 64 patients undergoing breast surgery. In the study conducted by Jossinet (1996), nine attributes had been identified from the EIS, in addition to the class data, i.e. type of tissue. Those attributes are Impedivity (ohm) at zero frequency (I0), Phase angle at 500 KHz (PA500), High-frequency slope of phase angle (HFS), Impedance distance between spectral ends (DA), The area under the spectrum (AREA), Area normalized by DA (A/DA), Maximum of the spectrum

(IPMAX), Distance between I0 and the real part of the maximum frequency point (DR) and Length of the spectral curve (P).

The dataset has six groups of breast tissues that include both normal and cancerous tissue. Those tissues are shown in the following table (Table 1).

Table 1: Types of tissues in the dataset

Type of Tissues		Code	No. of cases
Normal Tissue	connective tissue	con	14
	adipose tissue	adi	22
	glandular tissue	gla	16
Cancer Tissue	Carcinoma	car	21
	fibro-adenoma	fad	15
	mastopathy	mas	18

As a first step to fit machine-learning models, the data was partitioned as a train and test dataset. Sixty-six per cent of the data was used to fit the model, and the rest was used as test data. As all the variables are in different ranges, the data were preprocessed using the scale and centre method or range method before fitting machine-learning models. For the backpropagation neural network, the class data was flattened in the training dataset i.e. each class appears alone in separate columns as this is a multiclass data.

The CRAN R (v.3.5.3) statistical package was used to run the algorithms. The accuracy and Cohen's kappa coefficient (κ) were mainly used to compare the output. Besides, sensitivity, specificity, and confusion matrices were used to compare the performance.

The following sections illustrate the theory behind each classification algorithm.

2.1 Support Vector Machine (SVM)

Support Vector Machines, introduced by Boser et al. (1992) is one of the supervised learning techniques and successfully applied to several applications such as classification and regression, and time series prediction (Müller et al., 1997; Sapankevych and Sankar, 2009) and face recognition (Tefas et al., 2001). In simpler terms, SVM can be considered as a surface, also known as a hyperplane, which is a boundary between different types of points in data. The hyperplane divides the most homogeneous points in each sub-region in a multidimensional space (Dangeti, 2017).

The basic theory behind the support vector machine can be explained as follows as described by Bonaccorso (2017). Consider a dataset for classification: $X = \{\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n\}$ where $\bar{x}_i \in R$

It is assumed as a binary classification and set class labels as -1 and 1: $Y = \{\bar{y}_1, \bar{y}_2, \dots, \bar{y}_n\}$ where $\bar{y}_n \in \{-1, 1\}$

The goal is to find the best-separating hyperplane, for which the equation is:

$$\bar{W}^T \bar{x} + b = 0 \text{ where } \bar{W} = \begin{pmatrix} w_1 \\ \vdots \\ w_m \end{pmatrix} \text{ and } \bar{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix} \quad (1)$$

As such the function of the classifier can be written as:

$$\tilde{y} = f(\bar{x}) = \text{sgn}(\bar{W}^T \bar{x} + b) \quad (2)$$

In a real setting, there will be a margin between two classes with two boundaries, on which a few elements from both classes lie. These elements are called support vectors. When renormalizing the dataset the support vectors will lie on two hyperplanes with equations:

$$WTx + b = -1 \text{ and } WTx + b = 1 \quad (3)$$

2.2 Backpropagation algorithm (BPN)

The Backpropagation Algorithm is one of the artificial neural networks and is a supervised learning method. In this approach, the algorithm is provided with inputs and target outputs and the network computes the outputs, which are compared with the actual target output and error will be calculated between the target output and computational output. Then the algorithm will be backpropagated to adjust the weights so that the error will be reduced. Therefore, the backpropagation algorithm begins with assigning random weights and then the weights will be adjusted to minimize the error until the algorithm learns the training data. The solution is the combination of appropriate weights which has a minimal error (Puig-Arnavat et al., 2015). The algorithm consists of one or more hidden layers, which requires an activation function to help the algorithm to learn the linear and non-linear connection between input and output. The most commonly used activation functions are tan-sigmoid, log-sigmoid, and linear (Puig-Arnavat et al., 2015). According to Rojas (1996), this algorithm can be broken down into the following main steps: 1. Feed-forward computation 2. Backpropagation to the output layer 3. Backpropagation to the hidden layer 4. Weight updates The algorithm can be explained using the following example. Let's consider the X_i , Y_i , H_i , b_i and w_i are inputs, outputs, hidden layers, errors, and weights respectively. The hidden notes can be calculated based on the inputs, errors, and weights which have been randomly assigned using the following formula:

$$H_1 = x_1 w_1 + x_2 w_2 + b_1 \quad (4)$$

and

$$H_2 = x_1 w_3 + x_2 w_4 + b_1 \quad (5)$$

We will apply the following activation function (sigmoid function) to get the output.

$$\text{sigmoid} = \frac{1}{1 + e^{-x}} \quad (6)$$

These results in the output of H_1 and H_2 are as follows:

$$\text{out}H_1 = \frac{1}{1 + e^{-H_1}} \quad (7)$$

and

$$\text{out}H_2 = \frac{1}{1 + e^{-H_2}} \quad (8)$$

Now we can calculate the y_1 and y_2 and using the sigmoid function we can get the $\text{out}y_1$ and $\text{out}y_2$

$$y_1 = \text{out}H_1w_5 + \text{out}H_2w_6 + b_2 \quad (9)$$

and

$$y_2 = \text{out}H_1w_7 + \text{out}H_2w_8 + b_2\text{out}y_1 = \frac{1}{1 + e^{-y_1}} \quad (10)$$

and

$$\text{out}y_2 = \frac{1}{1 + e^{-y_2}} \quad (11)$$

Calculating total Error.

Now we can compare the calculated outputs with target output and calculate the total error as follows:

$$E_o = \sum \frac{1}{2}(\text{target} - \text{output})^2 \quad E_{Total} = E_1 + E_2 \quad (12)$$

Now we need to update the weights to minimize the error. Let's consider the w_5

$$\text{Error at } w_5 = \frac{\partial E_{total}}{\partial w_5} \quad (13)$$

Now we will split the above partial differentiation into multiple times.

$$\frac{\partial E_{total}}{\partial w_5} = \frac{\partial E_{total}}{\partial \text{out } y_1} * \frac{\partial \text{out } y_1}{\partial y_1} * \frac{\partial y_1}{\partial w_5} \quad (14)$$

Where $\frac{\partial E_{total}}{\partial w_5}$ is the change in w_5 .

Now we will update the w_5

$$w_5 = w_5 - \theta * \frac{\partial E_{total}}{\partial w_5} \quad (15)$$

where θ is the learning rate ($0 \leq \theta \leq 1$)

In the same way, we can update the error for w_6 , w_7 , and w_8 . Now we can update the weight w_1 as follows

$$\frac{\partial E_{total}}{\partial w_1} = \frac{\partial E_{total}}{\partial out H_1} * \frac{\partial out H_1}{\partial H_1} * \frac{\partial H_1}{\partial w_1} \quad (16)$$

Now we will update w_1 .

$$w_1 = w_1 - \theta * \frac{\partial E_{total}}{\partial w_1} \quad (17)$$

Where θ is learning rate ($0 \leq \theta \leq 1$)

In the same way, we can update the error for w_2 , w_3 , w_4 . Now we will do the forward process with the updated weights and we will check whether the calculated output is near to target value. This process will be repeated until we get satisfactory results.

2.3 Probabilistic neural network (PNN)

The PNN is a Bayes–Parzen classifier (Masters, 1995) and was developed by Specht (1990). Here the Bayes–Parzen classifier is breakdown into the number of simple processes implemented in a multilayer neural network. To understand the PNN, the Bayes theorem for conditional probability and Parzen’s method for estimating probability density function should be known.

Let’s assume sample x is taken from a collection of samples belonging to several distinct populations $(1, 2, \dots, k, \dots, K)$. $x = [x_1, x_2, \dots, x_p]$

Assume that,

- the posterior probability that a sample belongs to the k^{th} population is h_k .
- the cost associated with misclassifying that sample is c_k .
- the true probability density function of all populations is $f_1(x), f_2(x), \dots, f_k(x), \dots, f_k(x)$

Then, Bayes theorem classifies an unknown sample into the i^{th} population if $h_i c_i f_i(x) > h_j c_j f_j(x)$ for all population $i \neq j$

However, in Bayes’ classification, the probability density function is usually not known. But in a standard classification algorithm, the distribution of the population should be known or assumed. In general, the normal distribution is assumed and this assumption cannot always be safely justified. This

may result in the highest misclassification rate. Therefore, there is a need to derive the distribution from the training dataset. This derived distribution will be a multivariate probability density function (PDF) that combines all the explanatory random variables.

To derive a distribution estimator, Parzen (1962) method is generally used. This method originally proposed the univariate case of PDF, which was later extended to the multivariate case by Cacoullos (1966). As such the multivariate PDF estimator ($g(x)$) can be written as follows:

$$g(x_1, x_2, \dots, x_p) = \frac{1}{n\sigma_1\sigma_2\dots\sigma_p} \sum_{i=1}^n W\left(\frac{x_1-x_{1,i}}{\sigma_1}, \frac{x_2-x_{2,i}}{\sigma_2}, \dots, \frac{x_p-x_{p,i}}{\sigma_p}\right) \quad (18)$$

Where,

σ is smoothing parameters around the mean of p random variables x .

W is Weighting function

n is the total number of training data point

When all the smoothing parameters are assumed to be equal ($\sigma_1 = \sigma_2 = \dots = \sigma_p$), then the Gaussian function is used for W . Thus, the equation will be reduced as follows:

$$g(x) = \frac{1}{(2\pi)^p/2n\sigma^p} \sum_{i=1}^n \exp\left(-\frac{\|x - x_i\|^2}{2\sigma^2}\right) \quad (19)$$

Where x is the vector of random variables (explanatory variables), and x_i is the i^{th} training vector. As the sample size (n) increases, the PDF estimator asymptotically approaches the true probability density function.

2.4 Learning Vector Quantization (LVQ)

The LVQ is similar in architecture to Self-Organizing Maps (SOM) developed by Kohonen (1990). However, unlike the SOM, the LVQ is a supervised learning method (Da Silva et al., 2017). The LVQ has many advanced features such as good network performance, fast training speed, fewer neurons, high recognition rate, simple network structure, and input vectors that need to be normalized (Feng and Zhao, 2013). The LVQ as a classification algorithm supports both binary and multi-class classification problems (Brownlee, 2016). LVQ model makes codebook vectors, which represent class regions, through learning training datasets. The codebook vectors consist of elements placed around the respective class according to their matching level. If the element matches, it comes closer to the target class, if it does not match, it moves farther from it. With these codebooks, the model classifies new data.

The algorithm of LVQ can be explained as follows, as explained by Zhang and Zhao (2003). Let's assume the input vector as $x = (x_1, x_2, \dots, x_n)$, and the reference vector for i^{th} output neuron w_i as $w_i = (w_{1i}, w_{2i}, \dots, w_{3i})$

The Euclidean distance between the input vector and the reference vector of the i neuron can be defined as

$$D(i) = \sqrt{\sum_{j=1}^n (x_j - w_{ji})^2} \quad (20)$$

When the $D(i)$ is a minimum, the input vectors are compared to the reference vectors and the closest match is found. The winning reference vector, w_{i*} is then obtained by the formula as follows:

$$|w_{i*} - x| \leq |w_i - x| \quad (21)$$

Then the reference vectors will be updated using the following rule.

$w_{i*}(new) = w_{i*}(old) + \alpha(t)(x - w_{i*}(old))$ if x is in the same class as w_{i*} ,
 $w_{i*}(new) = w_{i*}(old) - \alpha(t)(x - w_{i*}(old))$ if x is in a different class from w_{i*} ,
 $w_{i*}(new) = w_{i*}(old)$ if x is not the index of the winning reference vector.
 Where the $\alpha(t)$ is the learning rate which ranges between 0 to 1.

2.5 K-Nearest Neighbors classification (KNN)

KNN is one of the simple algorithms in machine learning (Khamis et al., 2014; Dhriti and Kaur, 2017) and is instance-based learning and also called lazy learning (Dhriti and Kaur, 2017). When we get the dataset for training we will not process them, instead, it will be stored, but when we get a new instance, it will be classified based on similarity with the stored data (Jabbar et al., 2013).

The KNN simply classifies a case based on the closest neighbouring training cases in the feature spaces (Imandoust and Bolandraftar, 2013) and Khamis et al. (2014) stated that “objects that are ‘near’ each other will also have similar characteristics”. Thus if you know the characteristic features of one of the objects, you can also predict it for its nearest neighbour”. Since it is using the neighbouring cases to classify the new cases, the distance between the given unknown case and other cases in the training dataset will be calculated. The smallest distance corresponds to the sample in the training set closest to the unknown sample (Imandoust and Bolandraftar, 2013).

Let's assume $(x_i, \theta_i); i = 1, 2, \dots, n$ be given, where $x_i \in R^m; i = 1, 2, \dots, n$ and θ_i denotes the label of x_i for each i . In other words, θ is the class from which the observation x_i has come up.

Let's assume that the number of classes is $c; c \geq 2$, i.e. $\theta_i \in \{1, 2, \dots, c\}; \forall_i$

Let x be a point for which the label is not known. Therefore we need to find out the label for x . We can follow the following steps to determine the label for x .

1. Let k be a positive integer.

2. We can calculate the $d(x, x_i) \forall i; i = 1, 2, \dots, n$ where d denotes the Euclidean distance.
3. We will get n number of distances and we will arrange these distances in non-decreasing order.
4. We can then take the first k distances.
5. Find those k points corresponding to these k distances.
6. Let k_i denotes the number of points belonging to the i th class, among the k points; $i = 1, 2, \dots, c$
7. We can assign the x in class i if $k_i > k_j \forall j \neq i$

The performance of classification using KNN is based on two aspects such as the selection of K and the distance metric used (Ayyad et al., 2019). If the K is smaller, the estimation tends to be poor and if the K is higher, the estimating is over smoothing and decreases the performance (Imandoust and Bolandraftar, 2013). Thus in most of the KNN operations the K , in general, will be 3-10. There are different distance measures but the commonly used method is Euclidean distance (Ayyad et al., 2019).

2.6 C5.0 Decision tree approach

C5.0 algorithm is an advancement of Quinlan's C4.5 algorithm (Kuhn and Johnson, 2013) and the C5.0 is advanced in terms of speed, and memory and creates simpler trees than C4.5 (Shirali, 2016). It is one of the classification algorithms suitable for big data (Revathy and Lawrance, 2017). The theory behind the C5.0 algorithm can be explained as follows as explained by Moik (2018).

Let's assume given set S consist of n samples and each observation has X attributes that are assigned to class Y .

The algorithm creates a subset of observation by making the group of the same classes. The degree of being grouped into the same classes is called purity. The highest purity indicates the group has more observation of the same class. The purity will be measured by Entropy (E) using the following formula:

$$E(S) = \sum_{i=1}^c -p_i \log_2(p_i) \quad (22)$$

Where $E(S)$ is the purity of a group, c is several classes, and p_i is the proportion of observation within a class. If there are two classes, the $E(S)$ ranges between 0 to 1, while in more than 2 class cases, this will range from 0 to $\log_2 n$. Then the algorithm computes the change in Entropy, which is

called information gain (A). This results from a split on each of the possible attributes. This will be computed using the following formula:

$$Gain(A) = E(S) - E(S|A) \quad (23)$$

Where $E(S)$ denotes the entropy of a segment before the split and $E(S|A)$ is conditional information entropy, which is the entropy of the segments after the split. $E(S|A)$ is the sum of all the entropies after the split weighted by the proportion of the observations falling into each of the segments and this can be calculated using the following formula:

$$E(S|A) = - \sum_{j=1}^v p'_i \sum_{i=1}^m p_{ij} \log_2(p_{ij}) \quad (24)$$

Then the gain ratio will be calculated using the following formula.

$$GainRatio(A) = \frac{Gain(A)}{- \sum_{j=1}^v p'_j \log_2(p'_j)} \quad (25)$$

These steps will be repeated with all the other attributes and their values. Then, the information gain ratios will be compared with each other and the attribute with the highest information gain ratio will be used as a node that splits the dataset.

After that, the same procedure will be repeated with the subsets as long as one of the three conditions are met: (1) all the observations in the subset have the same class; (2) there are no attributes left that can divide the subsets and (3) all observations have the same value of an attribute but still belong to different classes.

3 Results

The SVM was tuned at three kernels to get the optimum parameter, and the output indicated that the best performance would be obtained at the cost (C) of 100 and gamma 0.4 at the radial kernel. This model identified 59 support vectors, and the average accuracy was 62.87% (the minimum is 42.90%, and the maximum is 85.71%). The model is then used to predict the test dataset and it results in an accuracy of 80.56% with a 95% confidence interval of 63.98% and 91.81% and with a Kappa value of 76.40%. The confusion matrix (Table 2) shows that except for the class car and gla., other classes have misclassification.

For the backpropagation neural network, the number of layers and associated neurons was decided arbitrarily by repeatedly fitting the model and analyzing the accuracy and kappa level at a constant learning rate at the logis-

tic activation function. The model with a single layer of 100 neurons resulted in the highest accuracy and Kappa. The learning rate is an essential parameter in the neural network. The backpropagation keeps changing the weights until there is the most significant reduction in errors by an amount known as the learning rate (Ciaburro and Venkateswaran, 2017). The smallest value of the learning rate will give the highest accuracy, but the neural network's convergence will be difficult. While the enormous learning rate makes the model converge quickly, it compromises the accuracy of prediction. In this study, the different learning rate was tested and the best accuracy and Kappa were found in the range of 0.01 to 0.05 and 0.08 for the Logistic activation function. This can be shown in the following graph (Fig 1).

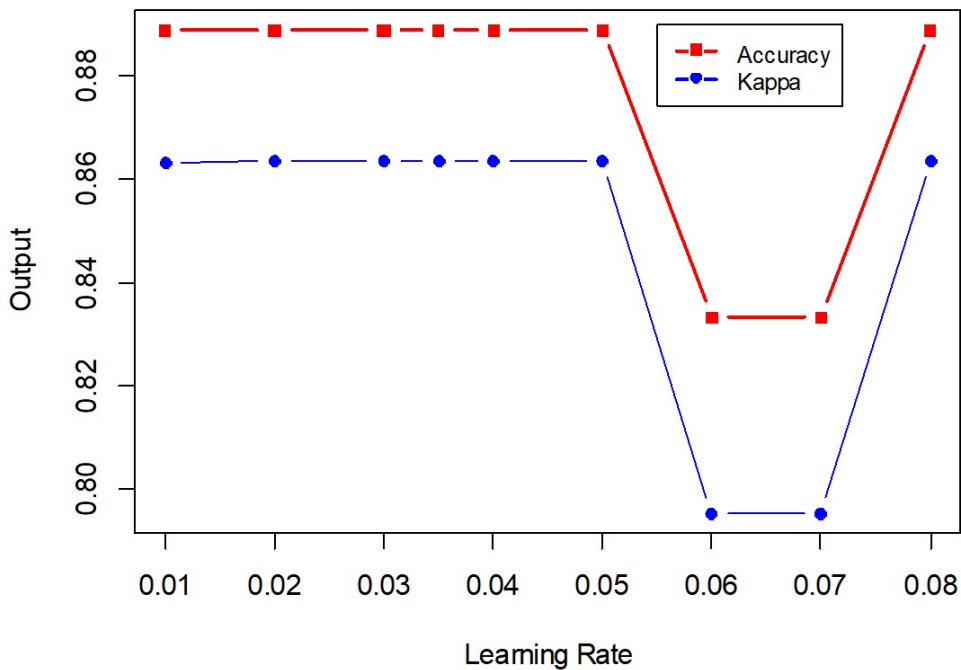


Fig. 1: The accuracy and the Kappa for different learning rates for the back-propagation algorithm

The model was then used to test the testing dataset with the above optimization parameters, and it resulted in an accuracy of 88.89% with a 95% confidence interval between 73.94% and 96.89%. The Kappa was 86.36%. The confusion matrix of the model yielded the highest accuracy (Table 2).

In the PNN, the training dataset was used to fit the model. The fitted model was then used for prediction using the test dataset. The model resulted in an accuracy of 69.44% with a 95% confidence limit of 51.89% and 83.65%. The Kappa value is 62.64%. The confusion matrix (Table 2) shows that misclassification was observed in all the classes except for the class car. Comparatively,

Table 2: Confusion matrix of different classification

SVM		Reference					
Predicted		adi	car	con	fad	gla	mas
	adi	7	1	0	0	0	0
	car	0	7	0	0	0	0
	con	1	0	6	0	0	0
	fad	0	1	0	2	0	3
	gla	0	0	0	0	5	0
	mas	0	1	0	0	0	2

BPN		Reference					
Predicted		adi	car	con	fad	gla	mas
	adi	8	0	0	0	0	0
	car	0	9	0	0	0	0
	con	0	0	6	0	0	0
	fad	0	0	0	2	0	3
	gla	0	0	0	0	5	0
	mas	0	1	0	0	0	2

PNN		Reference					
Predicted		adi	car	con	fad	gla	mas
	adi	6	0	1	0	0	0
	car	0	9	0	0	0	0
	con	2	0	3	0	0	0
	fad	0	0	0	2	0	3
	gla	0	0	0	0	5	2
	mas	0	1	2	0	0	0

LVQ		Reference					
Predicted		adi	car	con	fad	gla	mas
	adi	6	0	0	0	0	0
	car	0	8	0	0	0	0
	con	2	0	4	0	0	0
	fad	0	1	2	2	1	5
	gla	0	0	0	0	4	0
	mas	0	1	0	0	0	0

KNN		Reference					
Predicted		adi	car	con	fad	gla	mas
	adi	8	0	0	0	0	0
	car	0	9	0	0	0	0
	con	0	0	4	0	0	0
	fad	0	0	2	2	1	2
	gla	0	0	0	0	4	1
	mas	0	1	0	0	0	2

C5.0		Reference					
Predicted		adi	car	con	fad	gla	mas
	adi	8	0	0	0	0	0
	car	0	9	0	0	0	0
	con	0	1	6	0	0	0
	fad	0	0	0	0	2	4
	gla	0	0	0	2	3	1
	mas	0	0	0	0	0	0

the output is less significant than the previous two methods, such as SVM and the backpropagation algorithm.

In the LVQ, the best accuracy was obtained in the codebook size of 60 and k of 1. The average accuracy is 60.45%, and the average Kappa is 51.54%. The model was then used to predict the test data, and the confusion matrix was obtained. As such, the model yielded an accuracy of 66.67% with a confidence interval of 49.03% and 81.44% and with a Kappa value of 60.18%. The misclassification was observed in the classes con, fad, and mas (Table 2).

In the KNN, the critical parameter is the number of neighbouring points (k) for decision-making. Different numbers of neighbouring points (k) were applied for KNN and the resulting accuracy and Kappa were plotted against k (Fig 2). As such, the best accuracy was obtained at the k value of 9 and 10. Both the training and testing datasets were together used to fit the model at k of 9, and the confusion matrix was obtained. The model resulted in an accuracy of 80.56% with a 95% confidence interval of 63.98% and 91.81% and a Kappa value of 76.32%. The confusion matrix is given as follows (Table 2). The misclassification was observed in the classes of fad, gla, and mas.

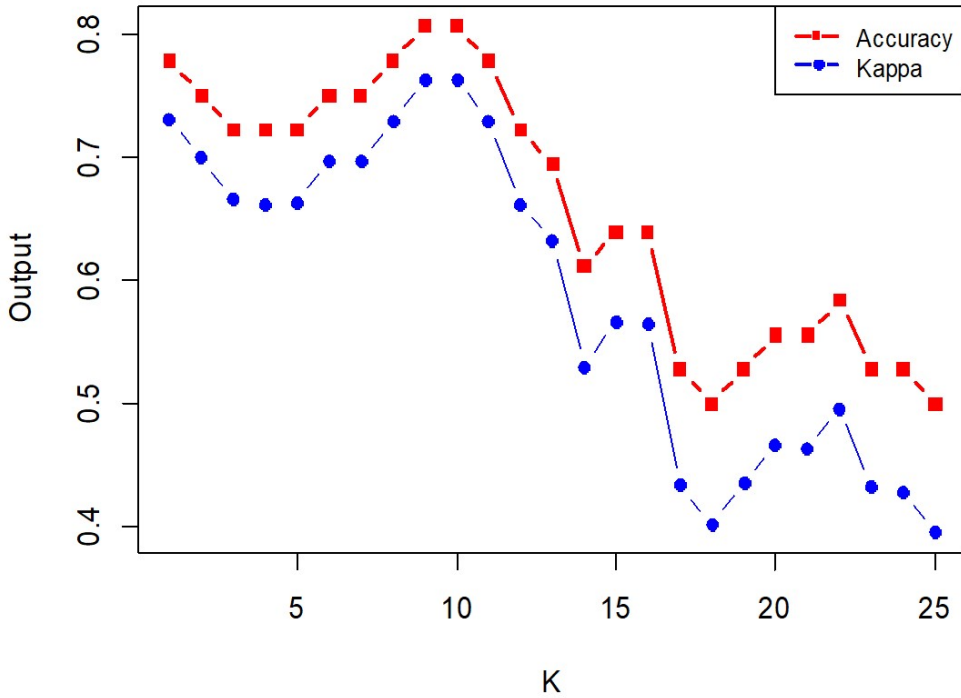


Fig. 2: Accuracy and Kappa for different k for KNN

In the C5.0 decision tree, the model resulted in an average accuracy of 65.71% (Min.42.86%, and Max. 100%) and the average kappa is 44.91% (Min. 28.20% and Max. 100%). The highest accuracy and Kappa were obtained at the rule-based and winnowing models. The fitted model is used to predict the test data and the confusion matrix. The model resulted in an accuracy of 72.22% with a 95% confidence interval of 54.81% and 85.80% and a Kappa value of 65.97%. Except for the classes, adi and car, misclassification is observed in all other classes (Table 2).

The overall comparison of accuracy is given in Table 3, and an overall comparison of all the algorithm's sensitivity and specificity is given in Table 4.

4 Discussion

A study conducted by George et al. (2012) reported that the predictive ability of the probabilistic neural network (PNN) and support vector machine was stronger than LVQ for two-class classification. However, in the present study of six class classifications, SVM surpasses the predictive ability of PNN and LVQ both by the accuracy of classification of the sensitivity of classes. Estrela da Silva et al. (2000) applied linear discriminant analysis for the breast tissue

data used in this study and reported that the straightforward one-step linear discriminant analysis was poorly performed. Instead, they applied a hierarchical approach using linear discriminant analysis to classify the tissue types. Moreover, Wu and Ng (2007) reported that the highest normalized weighted average algorithm resulted in the highest accuracy of 72.64% for the same six classed breast tissue data.

Table 3: Overall comparison of the accuracy of different machine-learning approaches

Algorithm	Kappa (%)	Accuracy (%)	95% CI		Conditions
			Lower	Upper	
BPN	86.36	88.89	73.94	96.89	Hidden layer with 100 neurons; $\eta=0.01-0.05$
SVM	76.40	80.56	63.98	91.81	Radial kernel, $C=100, \gamma=0.4$
KNN	76.32	80.56	63.98	91.81	$k=9/10$
C5.0	65.97	72.22	54.81	85.8	Rule-based with winnow
PNN	62.64	69.44	51.89	83.65	
LVQ	60.18	66.67	49.03	81.44	codebook size=35, $k=2$

However, this study found that the highest accuracy (88.89%) and Kappa (86.36%) were observed in the backpropagation algorithm. A study conducted by Helwan et al. (2017) to compare the radial basis function neural network and backpropagation algorithm on the same dataset reported that the backpropagation resulted in an accuracy of 91.67% for the test dataset (no. of testing sample 36) for 80 numbers of the hidden neuron and at the learning rate of 0.3. However, this was tested in the present study and found no accuracy level for the same number of neurons and learning rates.

The second-highest level is achieved by Support Vector Machine and K-Nearest Neighbours. The C5.0 decision tree algorithm, PNN and LVQ result in the third, fourth and fifth levels respectively at the accuracy level.

To measure the agreement between reference and prediction, the Kappa statistics were used. According to Landis and Koch (1977) the backpropagation algorithm result in an ‘almost perfect’ agreement, while SVM and KNN result in a ‘substantial’ agreement and the rest of them result in a ‘moderate’ agreement. The overall comparison of the sensitivity and specificity of each class was given in Table 4. Power et al. (2013) reported that ‘for a test to be useful, sensitivity+specificity should be at least 1.5 (halfway between 1, which is useless, and 2, which is perfect).’ Accordingly, all six algorithms are not up to the standard in classifying the tissue mas, as their values of sensitivity plus specificity are less than 1.5. The algorithm PNN and LVQ are not

Table 4: Overall comparison of sensitivity and specificity of different machine-learning approaches

Sensitivity & Specificity of Algorithms		Class					
		adi	car	con	fad	gla	mas
SVM	Se	0.88	0.7	1	1	1	0.4
	Sp	0.96	1	0.97	0.88	1	0.97
BPN	Se	1	0.9	1	1	1	0.4
	Sp	1	1	1	0.91	1	0.97
PNN	Se	0.75	0.9	0.5	1	1	0
	Sp	0.96	1	0.93	0.91	0.94	0.9
LVQ	Se	0.75	0.8	0.67	1	0.8	0
	Sp	1	1	0.93	0.74	1	0.97
KNN	Se	1	0.9	0.67	1	0.8	0.4
	Sp	1	1	1	0.85	0.97	0.97
C5.0	Se	1	0.9	1	0	0.6	0
	Sp	1	1	0.97	0.82	0.9	1

Se- Sensitivity; Sp-Specificity

up to the standard in classifying the tissue con. The algorithm C5.0 is also not useful in classifying the tissue fad. Comparatively, the backpropagation algorithm results in an adequate level of sensitivity except for the class ‘mas’ and a good level of specificity.

5 Conclusion

The highest accuracy (88.89%) and Kappa (86.36%) were observed in the backpropagation algorithm with one hidden layer with 100 neurons at any learning rate between 0.01 to 0.05. The support vector machine and k-nearest neighbours are at the second level. The support vector machine results in 80.56% accuracy and a Kappa of 76.4% at the radial kernel, cost of 100 and gamma of 0.4, while the K-nearest neighbour results in the same accuracy slightly different Kappa of 76.38% at the k of three. The C5.0 decision tree algorithm is at the third-highest accuracy of 72.22% and Kappa of 66.42%. This resulted in a rule-based model with winnowing. Probabilistic Neural Network and Learning Vector Quantization result in fourth and fifth-highest accuracy, respectively. Accuracy and Kappa for Probabilistic Neural Network were 69.44% and 62.64%, respectively, and for Learning Vector Quantization 66.57% and 60.18% respectively at the codebook size 60 and k of one.

The backpropagation algorithm resulted in comparatively highest-level specificity and sensitivity for most classes compared to other machine learn-

ing algorithms. A more or less similar order of accuracy was followed in the sensitivity and specificity among the classifiers. Overall, the backpropagation algorithm outperforms well in the classification of six-class breast tissue impedivity data. The study has a limitation i.e., that the findings are highly specific to the dataset used and cannot be generalized due its small size.

References

- Araujo, T., Aresta, G., Castro, E., Rouco, J., Aguiar, P., Eloy, C., Polonia, A. and Campilho, A. (2017), 'Classification of breast cancer histology images using Convolutional Neural Networks', *PLoS One* 12(6), 1 – 14. doi: 10.1371/journal.pone.0177544.
- Aruna, S., Rajagopalan, S. and Nandakishore, L. (2011), 'Knowledge based analysis of various statistical tools in detecting breast cancer', *Computer Science Information Technology (CS IT)* pp. 37–45. doi: 10.5121/csit.2011.1205.
- Ayyad, S. M., Saleh, A. I. and Labib, L. M. (2019), 'Gene expression cancer classification using modified K-Nearest Neighbors technique', *BioSystems* 176(December 2018), 41–51. doi: 10.1016/j.biosystems.2018.12.009
- Bonaccorso, G. (2017), *Machine Learning Algorithms*, Packt Publishing Ltd., Birmingham, UK.
- Boser, B. E., Guyon, I. M. and Vapnik, V. N. (1992), 'A Training Algorithm for Optimal Margin Classifiers', in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, ACM, New York, NY, USA, pp. 144–152. doi: 10.1145/130385.130401
- Brownlee, J. (2016), '*Learning Vector Quantization for Machine Learning*'. URL: <https://machinelearningmastery.com/learning-vector-quantization-for-machine-learning/>
- Cacoullos, T. (1966), 'Estimation of a multivariate density', *Annals of the Institute of Statistical Mathematics* 18(1), 179–189. DOI: doi.org/10.1007/BF02869528
- Ciaburro, G. and Venkateswaran, B. (2017), *Neural Networks with R: Smart models using CNN, RNN, deep learning, and artificial intelligence principles*, Packt Publishing, Livery Place.

- Da Silva, I., Spatti, D. H., Luisa, R. A. F. and Liboni, L. H. B. F. d. R. A. (2017), *Artificial Neural Networks*, Springer International, Switzerland. doi: 10.1007/978-3-319-43162-8
- Dangeti, P. (2017), *Statistics for Machine Learning*, Packt Publishing, Birmingham. URL: <https://worldcat.org/title/995570058>
- Dhriti, D. and Kaur, M. (2017), ‘K-Nearest Neighbor Classification Approach for Face and Fingerprint at Feature Level Fusion’, *International Journal of Computer Applications* 60(14), 13–17. doi: 10.5120/9759-1517
- Estrela da Silva, J., Marques de Sa, J. and Jossinet, J. (2000), ‘Classification of breast tissue by electrical impedance spectroscopy.’, *Medical biological engineering computing* 38(1), 26–30 pp. 26–30. doi: 10.1007/BF02344684
- Feng, Y. and Zhao, C. (2013), ‘Application of learning vector quantization neural network in the financial failure prediction’, *BioTechnology An Indian Journal* 8(2), 186–192. doi: 10.1167/6.11.8
- George, Y. M., Elbagoury, B. M., Zayed, H. H. and Roushdy, M. I. (2012), ‘Breast Fine Needle Tumor Classification using Neural Networks’, *International Journal of Computer Science Issues* 9(5), 247–256.
- Helwan, A., Idoko, J. B. and Abiyev, R. H. (2017), ‘Machine learning techniques for classification of breast tissue’, *Procedia Computer Science* 120 (2017), 402–410. doi: 10.1016/j.procs.2017.11.256
- Imandoust, S. B. and Bolandraftar, M. (2013), ‘Application of K-Nearest Neighbor (KNN) Approach for Predicting Economic Events: Theoretical Background’, *Journal of Engineering Research and Applications* 3(5), 605–610.
- Jabbar, M. A., Deekshatulu, B. and Chandra, P. (2013), ‘Classification of Heart Disease Using K- Nearest Neighbor and Genetic Algorithm’, *Procedia Technology* 10, 85–94. doi: 10.1016/j.protec.2013.12.340
URL: <https://linkinghub.elsevier.com/retrieve/pii/S2212017313004945>
- Jossinet, J. (1996), ‘Variability of impedivity in normal and pathological breast tissue’, *Medical and Biological Engineering and Computing* 34(5), 346–350. doi: 10.1007/BF02520002

Jossinet, J. (2010), 'Breast Tissue Data Set'.

URL: <http://archive.ics.uci.edu/ml/datasets/breast+tissue> (Accessed: 10 January 2018).

Khamis, H. S., Cheruiyot, K. W. and Kimani, S. (2014), 'Application of k-Nearest Neighbour Classification in Medical Data Mining', *JICT, International Journal of Information and Communication Technology Research* 4 No. 4(April), 990–1000.

Kuhn, M. and Johnson, K. (2013), *Applied Predictive Modeling*, Springer, New York. doi: 10.1007/978-1-4614-6849-3

Landis, J. R. and Koch, G. G. (1977), 'The Measurement of Observer Agreement for Categorical Data', *Biometrics* 33(1), 159–174.

Lvovich, V. F. (2014), *Electrochemical Impedance Spectroscopy (EIS) Applications to Sensors and Diagnostics*, Springer New York, New York, NY, pp. 485–507. doi: 10.1007/978-1-4419-6996-567

Masters, T. D. (1995), *Advanced algorithms for neural networks*, Wiley. Moik, C. (2018), *Predicting Credit Default by C5.0 Decision Tree Algorithm*. URL: <https://cmdataresearch.com/2018/02/11/predicting-credit-default-by-c5-0-decision-tree-algorithm/>

Müller, K.-R., Smola, A. J., Rätsch, G., Schölkopf, B., Kohlmorgen, J. and Vapnik, V. (1997), Predicting Time Series with Support Vector Machines, in *Proceedings of the 7th International Conference on Artificial Neural Networks, ICANN '97*, Springer-Verlag, Berlin, Heidelberg, pp. 999– 1004. URL: <http://dl.acm.org/citation.cfm?id=646257.685538>

Oliver, A., Freixenet, J., Martí, R. and Zwigelaar, R. (2006), 'A comparison of breast tissue classification techniques.', *Med Image Comput Comput Assist Interv* 9(Pt 2), 872–879.

Olmez, E., Areta, U. and Orhan, E. (2021), 'Classification of Breast Cancer using Artificial Neural Network Algorithms', *Artificial Intelligence theory and applications* pp. 57–68.

Parzen, E. (1962), 'On Estimation of a Probability Density Function and Mode', *The Annals of Mathematical Statistics* 33(3), 1065 – 1076. DOI: 10.1214

/ aoms / 1177704472

- Power, M., Fell, G. and Wright, M. (2013), 'Principles for high-quality , high-value testing', *Evidence-Based Medicine* 18(1), 5–10. doi: 10.1136/eb-2012-100645 1Pharmacy
- Puig-Arnavat, M., Bruno, J. C. and Bruno, C. (2015), Chapter 5 - Artificial Neural Networks for Thermochemical Conversion of Biomass, in A. Pandey, T. Bhaskar, M. Stöcker and R. K. Sukumaran, eds, *Recent Advances in Thermo-Chemical Conversion of Biomass*, Elsevier, Boston, pp. 133–156. doi: 10.1016/B978-0-444-63289-0.00005-3
- Revathy, R. and Lawrance, R. (2017), 'Comparative Analysis of C4.5 and C5.0 Algorithms on Crop Pest Data', *International Journal of Innovative Research in Computer and Communication Engineering* 5(1), 50–58.
- Rojas, R. (1996), *Neural Networks: A Systematic Introduction*, Springer Berlin, Heidelberg. doi: 10.1007/978-3-642-61068-4
- Sapankevych, N. I. and Sankar, R. (2009), 'Time Series Prediction Using Support Vector Machines: A Survey', *IEEE Computational Intelligence Magazine* 4(2), 24–38. doi: 10.1109/MCI.2009.932254
- Shirali, R. (2016), *Classification Trees and Rule-Based Modeling Using the C5 . 0 Algorithm for Self-Image Across Sex and Race in St . Louis*, PhD thesis, Washington University in St. Louis.
- Specht, F. (1990), 'Probabilistic Neural Networks', *Neural Networks* 3, 109–118.
- Tefas, A., Kotropoulos, C. and Pitas, I. (2001), 'Using support vector machines to enhance the performance of elastic graph matching for frontal face authentication', *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23(7), 735–746.
- Walker, D. C., Brown, B. H., Hose, D. R. and Smallwood, R. H. (2000), 'Modelling the electrical impedivity of normal and premalignant cervical tissue', *Electronic letters* 36(19).doi: 10.1049/el:20001118
- Wu, Y. and Ng, S. C. (2007), 'Breast tissue classification based on unbiased linear fusion of neural networks with normalized weighted average algo-

rithm', *IEEE International Conference on Neural Networks - Conference Proceedings* pp. 2846–2850. doi: 10.1109/IJCNN.2007.4371411.

Zhang, Y.X. and Zhao, Y.H. (2003) 'Learning Vector Quantization for Classifying Astronomical Objects', *Chinese Journal of Astronomy and Astrophysics*, 3(2), pp. 183–190. doi: 10.1088/1009-9271/3/2/183.