Breast cancer diagnosis through machine learning

Jonah M. Northwood
Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA 56267
north305@morris.umn.edu

ABSTRACT

Breast cancer is a prevalent cause of death for women across the globe. Survival rates of breast cancer are directly linked to how early a possible case is detected and treated. This is especially important in less developed countries that don't always have sophisticated testing methods to diagnose possible breast cancer patients. This paper looks at the use of various machine learning algorithms, namely k-nearest neighbors and support vector machines, to assist with breast cancer diagnosis using data regarding patient's tumors, along with other factors, to hopefully provide doctors a better way to properly diagnose cancerous tumors. This will allow patients to be treated in the early stages of cancer, lowering their risk significantly.

Keywords

Breast cancer, Support vector machines, Data visualization, Tumors, Machine learning

1. INTRODUCTION

Significant research is being done on applications and effectiveness of machine learning algorithms in the classification of tumors for breast cancer diagnosis. They will hopefully be used to help doctors improve their current diagnostic methods and minimize potential errors. These algorithms evaluate different features of a tumor collected through biopsies to predict whether it is malignant or benign.

In this paper, two machine learning algorithms will be looked at: k-nearest neighbours and support vector machines. We will evaluate both algorithms examining their performance at predicting the classification of tumors.

We will review several papers that compare multiple machine learning algorithms but we will be focusing on k-nearest neighbors and support vector machines as they seemed to be the most accurate based on the data from these papers. Some of the other algorithms analyzed in these papers include decision trees, random forest, and logistic regression.

We will begin this paper in section 2 by going over some background on cancer and machine learning techniques. Section 3 will look at how the data and machine learning algorithms are set up and evaluated for correctness. In section 4 we will discuss the studies featured in this paper. Finally, in section 5 we will look at the results from these papers and how k-nearest neighbors and support vector machines compare.

UMM CSci Senior Seminar Conference, October 2020 Morris, MN.

2. BACKGROUND

In this section, we will go over some background information that will be used throughout the paper. We will begin with some breast cancer background, talk about some basics of machine learning, and go over the two machine learning algorithms we will be discussing in this paper: k-nearest neighbors and support vector machine.

2.1 Breast Cancer Background

Abnormal production of cells that form in large clumps are known as tumors. Tumors can be benign, meaning that the cells in the tumor are normal and are generally not very worrisome. However, if these cells grow uncontrollably they are known as malignant tumors and the cells are known as cancerous cells. These can often become life threatening when they spread to other parts of the body through metastasis. Malignant tumors are very challenging to treat but it is much easier if the tumor is detected and treated early.

The first step to cancer diagnosis is detecting a tumor. This can sometimes be done by actually seeing or feeling a lump, but, often a tumor is only detectable through imaging such as mammograms or MRIs. If a tumor is detected, it needs to be analyzed to determine if it is benign or malignant. This is typically done through a biopsy, taking a small sample of the tumor which is then analyzed in a lab. With the information from the biopsy, a doctor determines whether it is malignant or not and what actions need to be taken. The machine learning approach being explored here is specifically looking to improve the effectiveness of determining whether a tumor is benign or malignant.

2.2 Machine Learning

Machine learning is a subset of artificial intelligence or AI.

There are several types of machine learning. Classification is one in which a label is given to whatever data point is being examined. In the case of breast cancer a tumor would be classified either as benign or malignant.

Training in machine learning is where you take a section of a dataset and give it to the algorithm to learn from. This is done by giving the algorithm the data point as well as what classification it belongs to. Given enough data, the algorithm should 'learn' what in the data means it is more likely to be a certain classification over others. For example, if many of the tumors in the test set that were above a

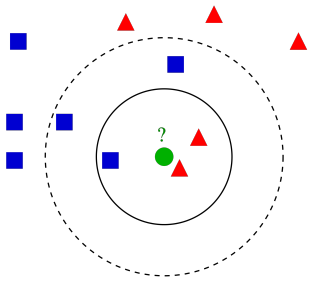

Figure 1: An example of KNN [3]

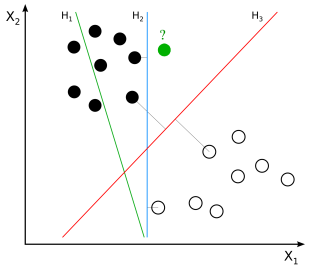

Figure 2: An example of SVM [9]

certain size were found to be malignant, then the algorithm would be more likely to think that other tumors above that size would be malignant. The algorithm is then able to use what is learned from in the data to classify future unknown data points.

Dimensionality is how many features are being looked at. For example, a two-dimensional space might have a tumor's radius and texture. While a three-dimensional space would have another feature like radius, texture, and compactness. The number of dimensions being considered can be quite high and can affect the performance of machine learning algorithms. To combat this, it is possible to reduce the number of dimensions while keeping most of the important data through methods like Principle Component Analysis, which will be discussed in section 3..

Different algorithms can have different advantages and disadvantages depending on the data evaluated. Across the papers examined, algorithms k-nearest neighbors and support vector machine generally performed the best on average.

2.2.1 *K-Nearest Neighbors*

K-Nearest Neighbors, or KNN, is a commonly used machine learning algorithm that is classified as using 'lazy learning'. When using a lazy learning method, the training data is processed during runtime, instead of processing the training data and then handling queries at a later time. KNN predicts the classification of a data point by looking at the classification of its k nearest neighbors, with k being a value the user chooses. Because of this, KNN's performance is directly related to the k value chosen, so multiple k values should be tested to find the most accurate one for a given dataset.

An example of KNN in two dimensions can be seen at Figure 1. If we take the blue squares to represent data points from the test data that were identified as benign tumors and the red triangles to represent data points of malignant tumors, we are then able to use this to predict the classification of the point we are looking at, in this case, the green circle. If k=3, the algorithm would look at the 3 closest points to the one being classified, in this case, the points inside the solid line. The KNN classifier would then predict that the point belongs to the red triangle class, or is a malignant tumor, as there are more red triangles then other classes in the nearest k neighbors. If k=5, it would take a look at the 5 closest points, in this case being the points inside the dotted line, and would predict the point belongs to the blue square, or the benign class.

KNN performs much better on datasets with lower dimensionality and, therefore, works well when paired with techniques that reduce dimensionality. As dimensions increase, the distance between the closest point and distance between the average point decreases. When the closest point isn't much closer to the point we are considering then the average point, KNN's predictive power is reduced. [2]

2.2.2 *Support Vector Machines*

Support Vector Machines, or SVMs, are another machine learning algorithm. SVMs predict classification by fitting a hyperplane in a high-dimensional space. The hyperplane is set one dimension lower than the one currently being examined. If we are looking in three dimensions, for example, a hyperplane is a 2-d plane. This gets harder to visualize in high dimensions, but a hyperplane will always split whatever you're looking at into two parts. If you have two clusters of data, the best way to separate them is to find the hyperplane that is furthest away from any data point. The hyperplane is then used for classification in that it is a divider between the two classes, so anything on one side is predicted to belong to class x and anything on the other is predicted to belong to class y.

An example of SVM in two dimensions can be seen in Figure 2. If we take the black circles to be data points belonging to the benign tumor class from our test dataset and white circles belonging to the malignant tumor class, then the red line is the hyperplane that best divides the data into its respective classes by being the furthest from any given point. The green line does not divide the data into the classes we want as it groups black circles with white circles and doesn't keep all the black circles together. The blue line does divide the data into the correct classes but not very well. If we wanted to classify the green circle, I think most people would predict it to belong to the black circle group based on distance, but if we went off the blue line it would belong to the white circle class.

3. METHODOLOGY

In this section, we review the methodology used in the experiments being examined. We begin talking about the dataset that they use, how they set up the data for the algorithms to effectively use the data, and how the algorithms will be evaluated so that they can be compared.

3.1 Dataset

The datasets looked at for different experiments vary in the number of samples, as well as what features are included in them. Naveen et al. [5] used the breast cancer dataset from the University of California, Irvine, or UCI (UCID). This dataset included 116 samples of tumors which

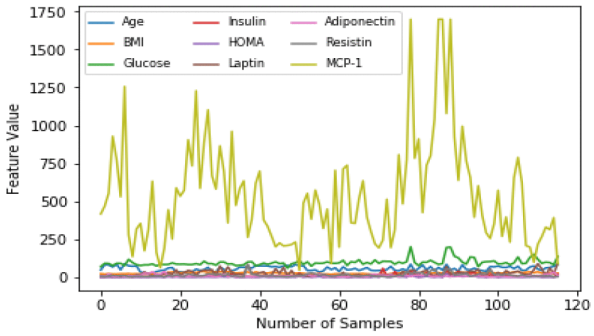

Figure 3: Graph on UCID's feature variance [5]

included nine features: Age (years) how old the patient is, BMI (kg/m2) mass divided by body height squared, which is a rule of thumb measurement typically used to determine underweight/overweight, Glucose (mg/dL) blood sugar levels, Leptin (ng/mL) a hormone that controls appetite and energy level and relates to the amount of fat tissue in the body, Adiponectin (ug/mL) a protein hormone that regulates glucose levels and is used to measure risk of type 2 diabetes, Resistin (ng/mL) a hormone which is related to cholesterol levels, Insulin (Uu/mL) a hormone that regulates metabolism, HOMA a measure of insulin resistance, and MCP-1 (pg/dL) a protein that is involved in inflammation.

Sharma et al. [8], Chakradeo et al. [1], Saoud et al. [7], and Kaklamanis et al. [4] all used the Wisconsin Breast Cancer dataset (WBCD). This has two datasets, the diagnostic dataset and the prognostic dataset. The diagnostic dataset has 569 entries with 357 benign tumors and 212 malignant tumors. The prognostic dataset has 198 entries with 151 benign tumors and 47 malignant tumors. These datasets include nine features: radius of the tumor, area of the tumor, perimeter of the tumor measured by counting the number of pixels, texture is a measurement of the variance in the grey-scale intensity of pixels, smoothness or the distance between lengths of the tumor, compactness which is the perimeter squared divided by area, concavity which is a measure of the number and severity of indentations, concave points which is only the number and not the severity of indentations, symmetry is the length difference between the two smallest sides of the tumor, and fractal dimension is a measure of how detail changes with scale.

3.2 Data Preprocessing

Data preprocessing is very important in machine learning as it can substantially improve the performance of the algorithms.

Feature importance gives weights to the different features being analyzed; this is very important as different features impact the chance of a tumor being benign or malignant by a different amounts. If feature importance is not considered all the features would be seen as equally as impactful by the algorithm, which would skew its results. An example of why this is important can be seen on the graph of the Naveen et al. dataset's variance on Figure 3. MCP-1 varies much more than the other features and, therefore, would overshadow them if scaling wasn't done.

Different features can also vary in their scale and units. In order to combat this, features can be normalized with methods like standard scaling or min-max normalization. Standard scaling [5] is accomplished by subtracting a value by its mean and then dividing that by its standard deviation as seen here:

$$y = \frac{x - \text{mean}(x)}{\text{Stdev}(x)}$$

Standard scaling scales all feature so that they have a mean value of 0 and a standard deviation of 1. This reduces the variance between features, making it so one feature can't overshadow others by having a larger variance.

An example of min-max normalization [8] can be seen here:

$$y = \frac{x - \min(x)}{\max(x) - \min(x)}$$

where x is the original value and min and max are the smallest and largest values of that feature. Min-max normalization rescales all features so that the range is between 0 and 1. This removes the issue of different features varying wildly in ranges.

Entries with missing values, something referred to as N/A values, can be dealt with by simply removing those entries from the dataset or filling the missing value using the mean or mode of the feature.

Standardization is generally preferred as it is not affected by outliers as much as min-max normalization. However, min-max normalization can lead to smaller standard deviation, which can be useful in certain situations.

A form of preprocessing commonly used across these experiments was a correlation matrix. A correlation matrix seeks to compare how closely different features of a given data set correlate to each other. This information can then be used to help in reducing dimensionality with highly correlated features through methods such as principal component analysis.

Principal component analysis , or PCA, is used to reduce the dimensionality of data. Dimensions in data refer to how many inputs, in this case features, are in the data. Reducing the dimensions in data is often important because more dimensions will mean the data points will be 'further' from each other, increasing the amount of noise in the data. While reducing dimensions is useful in this regard, by removing inputs you lose some amount of data. PCA seeks to reduce the amount of data lost by merging highly correlated inputs together into new composite variables. An example of highly correlated data would be something like a person's height and weight; while it's not always true that as height increases, weight increases, it is generally the case. PCA might then merge most of the data that is correlated between the two variables into one variable. Merging the highly correlated data together results in most of the 'important' or 'unique' data being concentrated in fewer variables and can remove some of the less 'important' variables to reduce dimensions. This is especially helpful for KNN as it behaves less accurately in higher dimensions.

PCA does have one big downside which is its interpretability. While PCA calculates what in the data should be important for results, it doesn't provide information on HOW it arrived at the new variables that it has. This makes it

		Prediction	
		Benign	Malignant
True	Benign	100(TP)	10(FN)
	Malignant	5(FP)	50(TN)

Table 1: Example confusion matrix

quite difficult for medical professionals to use it, as they would want to know how PCA arrived at its conclusion. Understanding WHY certain variables are deemed important would help to compare the results with current knowledge on classifying tumors. This knowledge could be used to enhance current techniques, find flaws in the algorithms, or better understand how the algorithms themselves are working. It is hard to fully trust AI, especially in potentially life-threatening situations like breast cancer, when it is unknown how the AI arrived at its conclusion.

3.3 Algorithm Evaluation

There are many different ways in which algorithms can be evaluated to understand how effective they were, as well as metrics to measure them by. Some of the metrics used that we will be looking at here are confusion matrix, sensitivity, specificity, accuracy, area under curve, and Cohen's kappa.

First off, all of our classifications can be grouped into four categories: true positives, true negatives, false positives, and false negatives. True positives are when positives in the data are properly predicted as positives, in our case when a tumor is that benign is properly predicted as benign. True negatives are when negatives, or in our case malignant tumors, are predicted as malignant. False positives and negatives then are when the algorithm predicts incorrectly, with a false positive being when a malignant tumor was improperly predicted as benign and a false negative being a benign tumor that was predicted by the algorithm as malignant.

A confusion matrix is a table layout that shows visually the performance of an algorithm. It shows true positive and negatives vs false positives and negatives and gives a visualization of the accuracy of an algorithm, showing what kind of errors might be happening. Examples of confusion matrices can be seen at Table 1. The visual nature of confusion matrices often makes it an easier way to analyze data compared to other numerical methods. For example, if there's a large value in the bottom left, the algorithm is predicting many false positives, meaning the algorithm is missing many malignant tumors.

Sensitivity, also known as recall, is a measure of true positive rate, in this case, the percentage that the algorithm correctly classifies a benign tumor as benign. Sensitivity ranges from 0 to 1 with 0 meaning no positives were predicted as positives and 1 meaning all positives were predicted as positives.

$$\text{sensitivity} = \frac{\text{number of true positives}}{\text{number of true positives} + \text{false negatives}}$$

Specificity in this case is measured as the percent of malignant tumors correctly identified as malignant. Specificity ranges from 0 to 1, with 0 meaning no negatives were predicted as negatives and 1 meaning all negatives were predicted as negatives.

$$\text{specificity} = \frac{\text{number of true negatives}}{\text{number of true negatives} + \text{false positives}}$$

Having a good balance between specificity and sensitivity is important. You could have an algorithm that classifies everything as positive, giving you a sensitivity of 1 every time. But in this case, with positives being a malignant tumor, you would be predicting all tumors that are actually benign to be malignant, causing more tests to be performed on the patient costing them time and money, and in the worst case causing them to undergo harsh treatments for cancer when they don't have it to begin with. On the other side of this, you don't want to sacrifice sensitivity for specificity as that means more malignant tumors would go untreated. Because of this, a balance needs to be struck between predicting positives vs negatives.

Accuracy combines specificity and sensitivity into one percentage. This is the metric used by most of the experiments looked at here. Accuracy ranges from 0 to 1 with 0 meaning that all negatives were predicted as positives and all positives were predicted as negatives. A value of 1 means that everything was predicted correctly.

$$\text{accuracy} = \frac{\text{true positives} + \text{true negatives}}{\text{total number of data points}}$$

Area under curve, or AUC, is another comparison of sensitivity and specificity. It is more of a direct comparison of the trade-off between true positive rate and false positive rate. Because of this, in certain situations this can provide much more meaningful data than other metrics. It is created using an ROC curve which is a graph of true positive rate, or sensitivity, vs false positive rate, or $(1 - \text{specificity})$. One is then able to find the probability that the algorithm correctly classifying tumors with a value of 1, or 100%, being correctly classifying a value of 0.5, or 50%, being equal to a model that just classifies randomly. More can be read about AUC from Sharma et al. [8] and Wikipedia [6].

An evaluation technique known as Cohen's kappa can also be used as a way to evaluate the information from a confusion matrix. It compares the observed accuracy, in this case

$$\text{observed accuracy} = \frac{\text{true benign} + \text{true malignant}}{\text{total number of data points}}$$

with the expected accuracy or the accuracy a random classifier would be expected to have which, in this case, is calculated with taking the probability that the algorithm correctly predicts benign (CPB) tumors at random and adding it to the probability that the algorithm correctly predicts malignant (CPM) tumors at random.

$$\text{CPB} = \frac{\text{TP} + \text{FN}}{\text{total data points}} * \frac{\text{TP} + \text{FP}}{\text{total data points}}$$

$$\text{CPM} = \frac{\text{FP} + \text{TN}}{\text{total data points}} * \frac{\text{FN} + \text{TN}}{\text{total data points}}$$

$$\text{expected accuracy} = \text{CPB} + \text{CPM}$$

This is sometimes the preferred evaluation technique as it takes into account random chance. This is beneficial because things like accuracy can be misleading in some situations. Imagine if you have 100 patients and 2 of them are sick. An

algorithm could get a 98% accuracy just by classifying every patient as healthy and giving no useful information. Cohen's kappa ranges from -1 to 1, with a value of 0 meaning the algorithm is as good as random guessing, -1 meaning it was categorizing everything incorrectly, and a value of 1 meaning it was categorizing everything correctly.

The formula for κ is:

$$\kappa = \frac{\text{observed accuracy} - \text{expected accuracy}}{1 - \text{expected accuracy}}$$

If we look at Table 5 and compute kappa, the observed accuracy = 0.965 and the expected accuracy = 0.537, therefore:

$$\kappa = \frac{0.965 - 0.537}{1 - 0.537} = 0.924$$

These are just the evaluation methods used most commonly across the experiments looked at in this paper, many others can be used.

4. FEATURED STUDIES

In this section, we will look at several papers that apply machine learning to the problem of breast cancer diagnosis, including what algorithms they look at, what preprocessing they used, and how they evaluated their algorithms. We will be focusing on KNN and SVM as they typically performed well. Other algorithms were tested across these studies, however we will not be going into detail about them here.

Kaklamanis et al. [4] compared four different machine learning algorithms using the same preprocessing and evaluations. They looked at CART, KNN, Naïve Bayes, and SVM and used standard scaling and PCA on their data. The evaluation techniques looked at were accuracy and kappa.

Naveen et al. [5] looked at six different machine learning algorithms: decision tree, SVM, multilayer perceptron, KNN, logistic regression, and random forest. They were specifically comparing the performance of machine learning algorithms using different models. They used standard scaling on the data and evaluated for accuracy, a confusion matrix, precision, sensitivity, and F1-score.

Saoud et al. [7] examined six algorithms comparing supervised vs unsupervised learning techniques. Supervised learning takes a set of already classified data, and learns from that, to categorize future unknown data. Unsupervised learning, on the other hand, does not use a training dataset and learns as it classifies; this is useful in situations where you don't have pre-classified data. For this comparison, they looked at SVM, KNN, and random forest for their supervised algorithms and expectation-maximization, simple K means, and filtered clustering for their unsupervised algorithms. They used an environment called WEKA to preprocess the data and run their algorithms but did not mention what preprocessing techniques were used. Algorithms were only evaluated for accuracy.

Sharma et al. [8] evaluated 3 machine learning algorithms: logistic regression, KNN, and SVM. They used min-max normalization on their data and evaluated for sensitivity, specificity, accuracy, and AUC.

Information on these studies' datasets and scaling methods cab be found at Table 2.

Paper	Dataset	Scaling	# of ML algorithms
Kaklamanis et al. [4]	WBCD	Standard	4
Naveen et al. [5]	UCID	Standard	6
Saoud et al. [7]	WBCD	N/A	6
Sharma et al. [8]	WBCD	Min-max	3

Table 2: Featured works and their properties

Metric	KNN	SVM
Specificity	94.7%	84.9%
Sensitivity	90.09%	88.2%
Accuracy	93.06%	89.55%
AUC	92.39%	86.55%

Table 3: Results on WBCD diagnostic dataset from Sharma et al. [8]

5. RESULTS

Looking at various experiments using these algorithms gives some insight into the benefits of using K-Nearest neighbor or Support Vector Machine and why one might be used over another for breast cancer diagnosis.

Naveen et al. [5] showed KNN could perform very well even reaching 89.9% accuracy in prediction of breast cancer patients and healthy individuals when the training dataset is much larger than the test set, 9:1 in this case. When the training to test dataset ratio is brought to 8:2 KNN's accuracy drops to 87.5%. A major part of the success of KNN is finding an optimal k value for the algorithm to use. The accuracy of KNN fluctuated below 85% with various K values between 1 and 7 with the optimal K value being found to be 5 for this data set. SVM was found to have an accuracy of 83.33%.

KNN typically performed better for larger data sets across various experiments. Sharma et al. [8] found that KNN resulted in having a stronger accuracy than SVM when used on a larger data set. This can be seen when looking at the researcher's diagnostic data set, which contained 699 entries (It wasn't clear if they meant the Breast Cancer Wisconsin (Diagnostic) Data Set here or the Breast Cancer Wisconsin (Original) Data Set). The results for this dataset can be seen on Table 3. In particular, KNN performed much better, 94.7% vs 84.9%, on specificity, meaning it was much better at correctly predicting negatives as negatives in this case. On the other hand, SVM typically has higher accuracy on smaller datasets like their prognostic data set, which contained 199 entries, as seen in Table 4.

Both of these data sets were tested using a 7:3 training to testing ratio. [8] In this case, SVM performed much better in specificity, 79.7% vs 61.2%.

Kaklamanis et al. [4] used a 7:3 training to testing data ratio. The confusion matrix for KNN can be seen at Table 5 and the confusion matrix for SVM can be seen at Table 6. KNN was found to have an accuracy of 96.49% while SVM was found to have an accuracy of 95.32%. However, SVM

Metric	KNN	SVM
Specificity	61.2%	79.7%
Sensitivity	40.89%	41.2%
Accuracy	82.56%	89.73%
AUC	51.045%	60.45%

Table 4: Results on WBCD prognostic dataset from Sharma et al. [8]

		Prediction	
		Benign	Malignant
True	Benign	106(TP)	5(FN)
	Malignant	1(FP)	59(TN)

Table 5: Kaklamanis et al. KNN Confusion matrix [4]

		Prediction	
		Benign	Malignant
True	Benign	105(TP)	6(FN)
	Malignant	2(FP)	58(TN)

Table 6: Kaklamanis et al. SVM Confusion matrix [4]

had a kappa value of 0.8988, while KNN had a kappa of 0.8145. This shows the importance of using multiple evaluation metrics because ,while KNN's accuracy was higher in this case, SVM had a higher kappa value. Using different evaluations of evaluating the algorithms can lead to different conclusions, so it's important to look at as many as possible to understand the performance of the algorithm as best as possible.

Saoud et al. [7] looked at a dataset with 569 different entries and found SVM to have a higher accuracy of 97.8% and incorrectly identified 12 different entries, both false positives and negatives. KNN, on the other hand, had an accuracy of 96.13% and incorrectly identified 22 different entries. [7]

6. CONCLUSION

While it is difficult to directly correlate different experiments due to a multitude of factors, my comparison of the data from these experiments suggests that the tipping point between KNN outperforming SVM is somewhere close to being between 569-699 entries of data using similar testing ratios and evaluation methods to these experiments. KNN also performs much better in lower dimensions so the dimension reducing techniques used in many experiments likely benefited KNN more than SVM in these tests.

Breast cancer is a prevalent issue worldwide. By using methods like machine learning to improve the current diagnosis of breast cancer, we can hope to lessen the impact of the disease. Experiments like these help show what the advantages are of using machine learning for medical diagnosis. They can help doctors better understand how to improve current diagnostic methods by taking a different look at how different features correlate to a tumor being cancerous. Both k-nearest neighbours and support vector machine are strong options in classifying tumors and can hopefully be used to improve cancer diagnosis in the future.

Acknowledgements

I would like to thank Nic McPhee, Elena Machkasova, and Yujing Song for their helpful advice and feedback.

7. REFERENCES

[1] K. Chakradeo, S. Vyawahare, and P. Pawar. Breast cancer recurrence prediction using machine learning. In *2019 IEEE Conference on Information and Communication Technology*. Institute of Electrical and Electronics Engineers, 2019.

[2] P. Grant. k-nearest neighbors and the curse of dimensionality.

[3] k-nearest neighbors. Wikipedia, The Free Encyclopedia, 2020.

[4] M. M. Kaklamanis and M. E. Filippakis. A comparative survey of machine learning classification algorithms for breast cancer detection. In *Proceedings of the 23rd Pan-Hellenic Conference on Informatics*. Association for Computing Machinery, 2019.

[5] Naveen, R. K. Sharma, and A. Ramachandran Nair. Efficient breast cancer prediction using ensemble machine learning models. In *2019 4th International Conference on Recent Trends on Electronics, Information, Communication Technology (RTEICT)*. Institute of Electrical and Electronics Engineers, 2019.

[6] Receiver operating characteristic. Wikipedia, the free encyclopedia, 2020.

[7] H. Saoud, A. Ghadi, and M. Ghailani. Proposed approach for breast cancer diagnosis using machine learning. In *Proceedings of the 4th International Conference on Smart City Applications*. Association for Computing Machinery, 2019.

[8] A. Sharma, S. Kulshrestha, and S. Daniel. Machine learning approaches for breast cancer diagnosis and prognosis. In *2017 International Conference on Soft Computing and its Engineering Applications (icSoftComp)*. Institute of Electrical and Electronics Engineers, 2017.

[9] Support vector machine. Wikipedia, The Free Encyclopedia, 2020.