ISSN: 2581-3404 (Online)
*International Journal of Innovative Research in Technology and Management, Vol-4, Issue-6, 2020.*

# Cancer Malignancy Prediction using Two Layer Deep Neural Network

**Archana Pathak[1], Prof. Nidhi Ruthia[3], Prof. Chetan Agrawal[3]**

**Scholar[1], Assistant Professor[2,3]**

**Department of Computer Science & Engg.[1,2,3]**

**RITS Bhopal, (M.P.), India[1,2,3]**

**ABSTRACT**

In this real-world, BC (Breast Cancer) is the second most common cause of death in developed and undeveloped countries. Among 8 percent of women diagnosed with Breast Cancer (BC) are marked by gene mutation, chronic pain, size change, color (redness), and breast skin texture. Breast cancer classification leads pathologists to a systemic and rational prognosis, usually with binary (benign cancer/cancer) classified more commonly. Machine Learning techniques (ML) are now widely used in the issue of breast cancer classification. They offer high accuracy in classification and efficient diagnostic ability. This paper introduces two layers Deep Neural Network for the classification of breast cancer followed by gradient approach and back propagation method. The findings indicate that two layer deep neural network is the most accurate (99.91%) with the lowest error rate relative to the NB classifier (96.19%).

**Keywords:** gradient descent, back-propagation, accuracy, epoch, ReLU.

## Introduction

One of the most common forms of breast cancer in women has been breast cancer. It can be kind and cruel. Medical research focused on prompt cancer identification as it spreads in the tissues of the surrounding body [1].

For the survival rate after cancer treatment, a rapid diagnosis of cancer is critical. Recent research has shown that in developing countries, the breast cancer survival rate ranges from 80 to 90% [2], while in developed countries, it is much lower. To continue to improve survival, both medical and master-learning algorithms [3] have been performed through interdisciplinary experiments to identify cancer cells [4]. Medical experiments are based on identifying breast cell differences, while machine learning algorithms screen breast tissue differences by image classification [5]. Machine learning classification algorithms are also used to identify malignant breast cancer and most of them are less than 99 percent correct [6][7].

This study aims to suggest improvements to SVMs that are 99.6 percent accurate to malignant tumors. These results significantly contribute to academic literature because it is most accurate to use classification algorithms, especially SVMs, to the Wisconsin breast cancer dataset. In contrast to other current approaches, our proposal also results in a lower error rate. Therefore, our algorithm can be extended to different datasets to enhance opportunities to detect malignant breast cancer.

## Background

Deep learning approaches for breast cancer detection include neural networks for breast tissue classification [8]. In this area, researchers use cancer tissue images to develop algorithms of

classification which predict breast cancer growth. Neural networks can be combined with the collection of features [9]including the ridge and line discriminating analysis. These algorithms are broadly used in medical screening because they can identify a patient as stable or diseased and assess the breast cancer type without previous knowledge of breast cancer [10]. These algorithms also achieve high accuracies in cancer prediction, and can be used as an alternative diagnostic tool in combination with medical tests. Genes expression analysis are also used for the prediction of cancer in which genes are classified on the basis of behavior [11].

Computer training methods are used to derive text information from patients' medical records on common signs of malignant breast cancer. In order to find the most common signs of breast cancer identified by patients, Forsyth et al. [12] for example developed a machine-learning algorithm. They reviewed 103,564 sentences to find the most common signs of cancer, pain, exhaustion and nausea. The features of breast cancer are also studied by Zhang et al. (2019) in anticipation [13]. Such research expands medical understanding of the causes and symptoms of breast cancer and increases the probability of reliable and prompt diagnoses.

Classification methods are another common form of breast cancer analysis in addition to neural networks. Classification approaches use quantitative data to estimate the type of breast cancer, in comparison to work already reported. Wisconsin's brain cancer dataset is the most common dataset used. There are objective details regarding the physical features of breast cancer, while the target variable is a categorical variable that matches benign and malignant cancer. The grading task is to predict malignant cases. This dataset has multiple writers investigated in different ways. Liu et al. (2019) [14]have for example, built a new way to improve the accuracy of the prediction by using feature selection on the IGSAGAW CSSVM data set. They applied support vector machines and K-Nearest Neighbor, Naïve Bayes, and decision tree. In predicting malignant cases he achieved 97.13 percent accuracy. Classification approaches are intended to assist in the detection of breast cancer through physical characteristics.

In Wisconsin breast cancer datasets, 94.7% accuracy was achieved by the decision-tree classifier (Quinlan, 1996) whilst ensemble learning algorithms achieved 97.4% exactness [15]. Though not commonly used, 98.2% of data were achieved by a nero-regulation-based method [16]. The most widely used algorithms for prediction of malignant breast cancer are the supportive vector machines (SVMs). Research has shown that SVMs with RBF kernel are the best algorithm to detect malignant breasts cancer [17]. They have achieved 96.8 percent accuracy. Additional papers showed that SVMs could be the best way to classify malignant cancer by grouping [18].

Vrigazova & Ivanov [19] have already implemented the modification of SVMs (ANOVA-SVM-BOOTSTRAP) in previous research and obtained an accuracy of 97.5%. This paper provides a summary of the Wisconsin breast cancer data collection for ANOVA-BOOTSTRAP-SVM. In contrast to classical SVMs, we also demonstrate that our algorithm greatly improves the effects on other breast cancer data sets. We suggest an improved version of the vector support devices, which can be used for the identification of malignant cancer in different breast cancer data bases.

**Proposed Methodology**

The proposed work is based on back propagation and gradient descent algorithm. The work is done in three phases. First we create a neural network, train it using the gradient descent algorithm. Secondly, applying that basic network to the Wisconsin Cancer Data-set and predict if a tumor is benign or malignant, based on 9 different features Lastly, explore deeply how back-propagation and gradient descent work by comparing it with existing machine learning

approaches. The description of the proposed work is shown in figure 1.

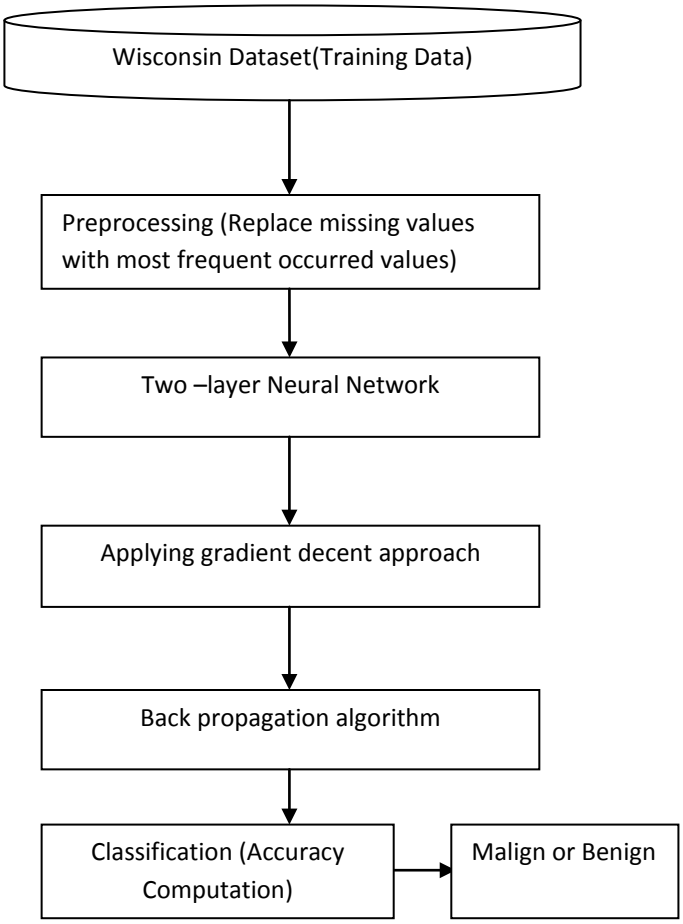


**Figure 1:** Flowchart of proposed work.

1. Neural Network – It has 2 layers (it is never counted for the input layer).
• Input: the network input comprises our data source. The number of neurons corresponds to the number of characteristics of our source data. The following graph uses four inputs. We will use 9 if we later use the cancer data collection from Wisconsin.
• First layer: it has a range of secret neurons for our first hidden layer. These neurons are bound in the layers around them to all units.
• The second and final layer has 1 single unit, network output.

It's possible to add additional layers and provide a 10 or 20 layer network. We're working with 2 in this article for convenience. As we will find out shortly, a 2 levels neural network can do a lot. We have entered some data in the input layer of our network. The network is also shown what output is the same as this input, which output will be shown at the network output (the second level).

Every unit in the network layers has a corresponding weight (and a bias later). There are only numbers that are usually randomly initialised at the beginning of the learning process. The neural network performs such calculations that combine input and weight data. And these calculations spread throughout the network until a final result is made.
The result is a function that maps output inputs. The calculations indicate a feature. Our aim is to improve the value of the weights for our network. Since it can estimate various functions through the calculations used by the network in accordance with the different layers. For this to happen, the names of all the variables involved in our project must be explained with accuracy.
• X is the input layer, the network data which we feed.
•Y will be the target output, the output to be obtainable at the end of the network following its calculations, that matches the input X.

The output we generate after we feed the network with X, will reflect our prediction, $\hat{Y}_h$. Y is the ideal output, so Y is the output that the network creates when our data is supplied.
• W is going to reflect network layer weights.
Let's begin with this calculation: W X (the product between W and X), a first layer, our hidden level.
It makes a weighted amount:
• Each layer unit is connected to the previous layer of each unit.
• For each of these links, a weight value is available.
• The new value of each unit in one layer will be the sum of the results by weighting the relation between the previous unit and the unit we are

currently evaluating to multiply the value of each previous unit.

In one way, the weight represents the strength or weakness of links between the various network units. We will add something additional to the product now a prejudicial term. Adding a word of choice provides the network with more flexibility. It enables the unit's linear calculations to 'jump around,' thereby increase the network's ability to learn the mystery functions faster.
B : This is the term of the units' discrimination (bias).
We've got it there: $WX + b$. We call this a linear equation because it describes a linear relationship (a relationship that can be represented with one line) between the input and the output, using a product and a sum. The output of the calculation will also be called
• Z is the performance of the layer calculation
$Z1 = W1\,X + b1$ therefore
Note that for each unit of each sheet, this calculation should be performed. We can use a vectorized implementation when programming the network. This implies that all calculations in a layer in one mathematical operation are combined using matrices.
Gradient - The gradient of a function at a time is often called its derivative and at that point it represents the rate at which the function output changes. If gradients (derivatives) become smaller (The output of the function is flat). In the back-propagation algorithm more discussion on the gradient, which was used extensively in deep learning, determines how to tweak the network weights by using gradients to see how each network parameter affects the network output. The vanishing gradients are a concern because it is very difficult to understand how much the performance of the system changes at a time when the gradient at a point becomes too small or null. We can also talk about the opposite issue, the gradients that burst. The network can become very unstable when the gradient values become very high. There may be different benefits of different activation functions. But they also can suffer from gradient problems that go away and burst. So in fact, a neural grid is a functional chain, some linear and some nonlinear, which produces a complex function, the mystery that connects your input data to your desired outputs. Note at this point that the values of W and b are the great unknowns out of all the variables in this equation. This is where it is important to understand. Somehow the network needs to learn the correct W and b values to allow the correct function to be measured. So we train our network to find the right $W_1$,$b_1$, $W_2$ and $b_2$ values. But we must initialize these values before we can start the training. Some of the activation functions that will be used are Relu and Sigmoid functions. And the forward function carries out the above calculations. By adding the first bias matrix $b_1$, we multiply the weights of the first layer and generate $Z_1$. Then we use $Z_1$, to generate $A_1$with the Relu function. Next, we multiply the second layer weight matrix, $A_1$ (the first layers output, that is the second layer input), and add the second bias matrix, $b_2$, so that $Z_2$ is generated. The Sigmoid function is then applied to $Z_2$ to generate the $A_1$, which is $\hat{Y}_h$, the network output. For this the production that we achieved can be contrasted with that that we ought to achieve: $\hat{Y}_h$. and Y. We will add a final network feature, the loss feature, to quantify this.

Calculus brings us something called the chain rule of derivatives, which when we look at it in depth really is a fairly basic idea. First of all, a partial derivative studies the change in a variable by modifying a different variable. The Chain Rule says we can chain partial derivatives between them by multiplying them to understand the effect of modifying a variable on another variable, if they're distant from each other. It is time to address the Back-Propagation algorithm [20] inside the neural network, and in this case in our 2 layer network in particular. The back-propagation uses the chain rule to assess the extent to which changes in our network's various parameters affect its final loss.

ISSN: 2581-3404 (Online)
*International Journal of Innovative Research in Technology and Management, Vol-4, Issue-6, 2020.*

Let's select one of our parameters and understand the action chain law and starts with the loss equation:

$Loss = -(YLogY_h + (1-Y)Log(1-Y_h))$

Let us measure how the loss is affected by a shift in $Y_h$ and the outcome. We can continue to chain derivatives until we reach $W_1$ after this. We look for the derivative equation of the loss function in calculating this derivative. By refreshing your calculus a bit or looking up online, you can easily find the derivatives of any kind. We find that in this case:

$$dLoss\_Yh = -(Y/Yh - (1-Y)/(1-Yh)) \_(Yh$$

Yh = sigmoid (Z2)

$$dSigmoid = sigmoid(x) * (1.0 - sigmoid(x))$$

We represent this difference equation as dSigmoid in order to simplify the writing. Hence:

$$dYh\_Z2 = dSigmoid(z2)$$

At this point, these 2 derivatives can already be chained (multiplied) for the Z2 derivative.

$$dLoss\,Z2 = dLoss\_Yh * dSigmoid(z2)$$

$$Z2 = A1\,W2 + b2$$

**Results and Discussion**

The whole work is implemented using Python 3 programming language. Below figures and tables show the results of the implemented model and the comparison with the existing approach. The proposed algorithms is compared with the algorithms such as K-NN, Naïve Bayes, Random Forest and Neural Network.

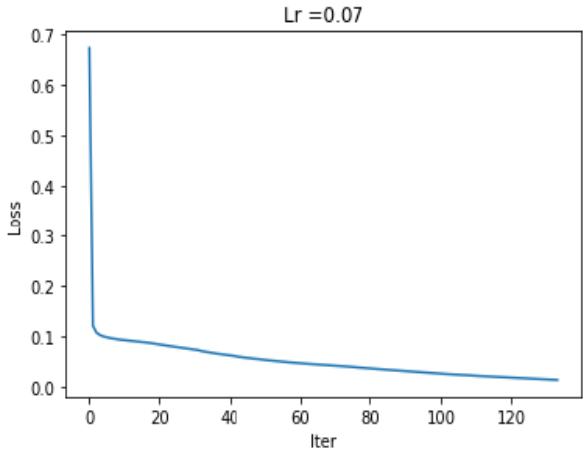


**Figure 2:** Loss computation.

Figure 2 has shown the loss graph and it can be observed that computed loss from the proposed method is very less.

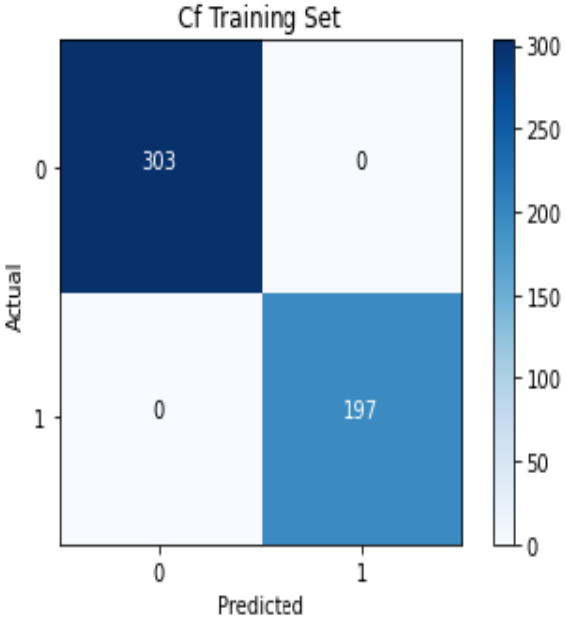


**Figure 3:** Training Set confusion matrix.

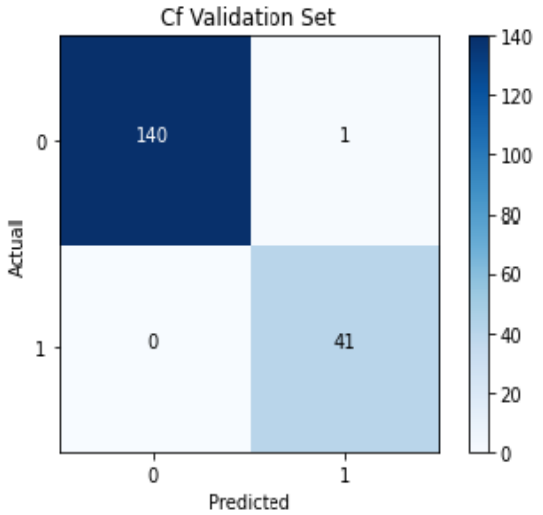


**Figure 4:** Testing Set confusion matrix.

The dataset is split into training and testing dataset in which 500 records are considered as the training dataset and 181 as the testing set. Above figure 3 and figure 4 has shown the confusion matrix of the training set and testing set respectively. However

we have achieved the accuracy of 100% on training dataset and 99% on testing dataset.

**Table 1.** Comparison of existing algorithms with proposed classifier

| Model | Accuracy (%) |
|---|---|
| KNN (K-Nearest Neighbor) | 97.51 |
| NB (Naïve Bayes) | 96.19 |
| Random Forests | 96.85 |
| neural networks | 93.02 |
| Deep Neural Network (Proposed) | 99.91 |

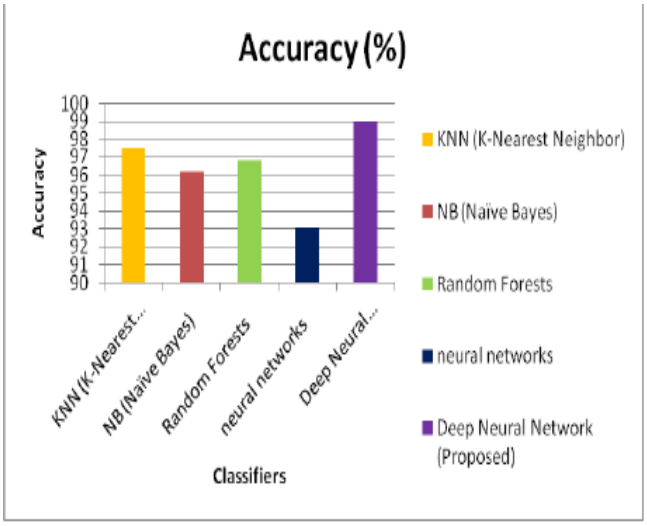


**Figure 5:** Accuracy comparison graph.

Figure 5 has shown the accuracy comparison graph of the exiting classifiers with the proposed classifier. The proposed model has achieved the accuracy of more than 99% and this is better than other states of the art algorithms.

**Conclusion**

Data mining represents a very popular approach in different areas of our lives. This statement also applies to medicine and healthcare with a few possible explanations. We can use suitable analytical methods to support the diagnostics process, identify potential hidden relations in the medical data, and extract new combinations of the biomarkers or drugs to improve the treatment process. The paper proposed two layers of Deep Neural Network to classify breast cancer on the Wisconsin Breast Cancer datasets, followed by a gradient approach and backpropagation method. The findings indicate that two layer deep neural network is the most accurate (99.91%) with the lowest error rate relative to the NB classifier (96.19%). In the future we will use some capsule network based cancer classification of the image database consist of images of X-rays, MRI or CT-SCANs.

**References**

[1] A. McGuire, J. A. L. Brown, and M. J. Kerin, “Metastatic breast cancer: the potential of miRNA for diagnosis and treatment monitoring,” Cancer Metastasis Rev., 2015.

[2] M. A. Schonberg, E. R. Marcantonio, D. Li, R. A. Silliman, L. Ngo, and E. P. McCarthy, “Breast cancer among the oldest old: Tumor characteristics, treatment choices, and survival,” J. Clin. Oncol., 2010.

[3] M. D. Ganggayah, N. A. Taib, Y. C. Har, P. Lio, and S. K. Dhillon, “Predicting factors for survival of breast cancer patients using machine learning techniques,” BMC Med. Inform. Decis. Mak., 2019.

[4] S. Horibata, T. V. Vo, V. Subramanian, P. R. Thompson, and S. A. Coonrod, “Utilization of the soft agar colony formation assay to identify inhibitors of tumorigenicity in breast cancer cells,” J. Vis. Exp., 2015.

[5] R. Turkki et al., “Breast cancer outcome prediction with tumour tissue images and machine learning,” Breast Cancer Res. Treat., 2019.

[6] A. Reeves and Y. Xie, “Software for density calculation and timing of surgery in subsolid nodules,” J. Thorac. Oncol., 2015.

[7] R. Nair and A. Bhagat, “Feature selection method to improve the accuracy of classification

algorithm," Int. J. Innov. Technol. Explor. Eng., 2019.

[8] A. M. Abdel-Zaher and A. M. Eldeib, "Breast cancer classification using deep belief networks," Expert Syst. Appl., 2016.

[9] E. Rolls and A. Treves, Neural Networks and Brain Function. 2012.

[10] P. Subramanian, N. O. Oranye, A. M. Masri, N. A. Taib, and N. Ahmad, "Breast cancer knowledge and screening behaviour among women with a positive family history: A cross sectional study," Asian Pacific J. Cancer Prev., 2013.

[11] R. Nair and A. Bhagat, "Genes expression classification using improved deep learning method," Int. J. Emerg. Technol., 2019.

[12] K. A. Parato, D. Senger, P. A. J. Forsyth, and J. C. Bell, "Recent progress in the battle between oncolytic viruses and tumours," Nature Reviews Cancer. 2005.

[13] Y. Li, S. Li, X. Meng, R. Y. Gan, J. J. Zhang, and H. Bin Li, "Dietary natural products for prevention and treatment of breast cancer," Nutrients. 2017.

[14] S. Alelyani, J. Tang, and H. Liu, "Feature Selection for Clustering: A Review," in Data Clustering, 2019.

[15] S. Bashir, U. Qamar, and F. H. Khan, "Heterogeneous classifiers fusion for dynamic breast cancer diagnosis using weighted vote based ensemble," Qual. Quant., 2015.

[16] R. Setiono, "Generating concise and accurate classification rules for breast cancer diagnosis," Artif. Intell. Med., 2000.

[17] V. Chaurasia and S. Pal, "Data mining techniques: To predict and resolve breast cancer survivability," Int. J. Comput. Sci. Mob. Comput. IJCSMC, 2017.

[18] R. M. Paz, E. A. Fontan, A. C. Maldonado, and J. I. Armesto, "A morphological analysis of pigmented skin lesions through digital image processing," in IFMBE Proceedings, 2015.

[19] B. Vrigazova and I. Ivanov, "Tenfold bootstrap procedure for support vector machines," Comput. Sci., 2020.

[20] M. Cilimkovic, "Neural Networks and Back Propagation Algorithm," Fett.Tu-Sofia.Bg, 2010.