The best architecture selection with deep neural network (DNN) method for breast cancer classification using MicroRNA data

S Ginanjar¹, Suhartono¹, A Wibowo¹ and E A Sarwoko¹

¹Department of Informatics, Faculty of Science and Mathematics, Diponegoro University
Jl. Prof.Dr.Soedarto, SH, Tembalang, Semarang 50275, Indonesia

Corresponding author: eko.adl.sarwoko@gmail.com.

Abstract. Breast cancer is one of the most common causes of death in the world. One way that can be done to reduce the number of death cases is to do early detection using MicroRNA data. MicroRNA is one of the cancer biomarkers that can help in the classification process. MicroRNA can be used to identify whether a cell is a cancer cell or not even in the earliest stages. The deep Neural Network (DNN) method consists of two or more layers of self-learning units (hidden units). The weight of hidden units that are fully connected between two layers can be learned automatically. However, DNN still has a weakness which is the changes in the distribution of each layer’s inputs that cause problems, because the layers need to continue to adapt to the new distribution and produce less optimal accuracy values. This research was conducted to get the best architectural selection of the DNN method used for breast cancer classification using MicroRNA data. The results of the DNN method with the best architecture obtained was 3 hidden layers with 200 hidden units each, 30% dropout rate in the combination of the ReLU, ReLU, ReLU activation functions and the learning rate of 0.04 resulting in the highest accuracy of 94.58% with a specificity of 96.45% and sensitivity of 91.02%.

1. Introduction
Breast cancer or carcinoma mammae is an uncontrolled cell growth in breast support tissue that surrounds the lobular (milk-producing glands), ductus (a nipple), blood vessels and lymph vessels [1]. According to data from the International Agency for Research on Cancer (IARC) (2018), it was found that breast cancer is the second leading cause of death in the world with 2,008,849 cases and 626,679 of them were declared dead. Statistical data on deaths worldwide by region shows that Asia ranks first in breast cancer cases with 911,014 cases and 310,577 declared dead. In fact, statistics on breast cancer by sex also show that the number of female deaths due to breast cancer cases is rank first in the world.

One of the efforts that can be done to reduce the number of cases of cancer deaths is early detection. Early detection of cancer can not only reduce mortality due to cancer but also improve the quality of life of sufferers (RI, 2015). Early detection using a computer is an automated aid that provides accurate results for analyzing a disease [2]. Classification is the method used by the machine to sort or classify based on specific characteristics [1]. One way that can be done to do early detection of breast cancer is to classify the MicroRNA data. MicroRNAs (miRNAs) are small, non-coded groups of Ribonucleic Acid (RNA) (ncRNAs) that regulate gene expression by targeting the appropriate carrier
RNA (mRNA) [3]. MicroRNA can be used to recognize whether a cell is a cancer cell or not even in the earliest stages [4].

MicroRNA classification process can be done in various ways. One classification technique that can be applied is to use the Deep Learning technique. Deep Learning or also called the Deep Neural Network (DNN) is one of the latest machine learning techniques and best for pattern recognition [5]. DNN has some hidden layer (hidden layer) and makes this model into a model to study the relationship expressive complex between its input and output [6]. The hidden layer gradually will map the low-level features that are not organized into a high-level data representation [5].

There are various studies using DNN. A Thomas & Sael (2017) study to predict the sequence of miRNA precursors using the 18.0 release miRBase dataset with 58 features shows that DNN with 96.8% accuracy outperforms the Support Vector Machine (SVM) method with an accuracy of 93.2%, Neural Network (NN) with an accuracy of 91.7%, Naïve Bayes Classifiers with an accuracy of 91.4%, K-Nearest Neighbors with an accuracy of 90.8%, and Random Forests with an accuracy of 93.7%. Research by Waspada et al. (2017) applied the supervised Machine Learning method for cancer classification according to MicroRNA data. The research shows that the Deep Learning method with an accuracy of 91.49% is superior to the Naïve Bayes method with an accuracy of 61.54%, Decision Tree with an accuracy of 34.15% and Neural Network with an accuracy of 5.48%. Other research by Joshi and Mehta (2018) using Neural Network and Deep Neural Network using the data Wisconsin Breast Cancer Dataset (WBCD) obtained from the UCI repository for the classification of breast cancer. The research shows that Neural Network with Linear Discriminant Analysis (LDA) gives better results in terms of accuracy compared to Deep Neural Network (97.06%), specificity (100%), and precision (100%). Deep Neural Network with LDA technique produces accuracy (62.94 %), specificity (00.00%), precision (100%).

One of the advantages of Naïve Bayes is that it does not require a large amount of training data to determine the estimated parameters needed in the classification process. The weakness of the SVM method is that it is difficult to use in large-scale samples and theoretically this method was developed only for the classification of two classes. While weaknesses in the NN method must use training data large enough to get maximum results. DNN with various processing layers reaches a higher classification level than Naïve Bayes, SVM, and NN because DNN is more suited to deeper architecture at fewer parameters.

From the research previously stated, although the DNN method has good performance, the DNN method still has weaknesses, namely changes in the input distribution of each layer which causes problems because the layers need to continue to adapt to new distributions so as to produce suboptimal accuracy values [7].

In this research, the selection of the best architecture for the DNN method for classification of breast cancer using MicroRNA data with the selection of the best architecture from the combination of activation functions, optimization methods, the number of hidden layers and hidden units along with the number of epochs. Aktivasi used was Exponential Linear Units (ELU), Rectified Linear Units (ReLU) and Sigmoid.

2. Theoretical basis

2.1. MicroRNA as potential cancer biomarker

MicroRNAs (miRNAs) are small, non-coded RNA groups (ncRNAs) that regulate gene expression by targeting the appropriate carrier RNA (mRNA) [3]. Recent studies have shown that about 50% of MicroRNAs recorded are related to cancer and are located in fragile areas, which are an area of the genome associated with cancer [8].

Biomarkers are characteristics that are objectively measured and evaluated as an indicator of normal biological processes, pathogenic processes, or responses to therapy [3]. The need for biomarkers for early diagnosis of cancer is very important because survival and prognosis (prediction of possible treatment, duration, and outcome of a disease based on general knowledge of pathogenesis and the
presence of disease risk factors) patients are very dependent on the stage of the tumor at detection, with early diagnosis generally can provide a better prognosis [3]. The miRNA signs that have been identified will have strong potential as biomarkers of early diagnosis. Distorted miRNA production is the beginning of an event in cancer development and supports the possibility of using it for early detection of cancer [3].

2.2. Classification
Classification is the method used by the machine to sort or classify objects based on specific characteristics as a man trying to distinguish objects from one another. The classification process studies information patterns from existing data, such as features, variables, and features of data items that have been labeled previously [9].

2.3. Deep neural network (DNN)
Neural Network (NN) is an information processing system that is inspired by the working principle of biological neural networks that have inputs and outputs called dendrites and axons. A typical Artificial Neural Network (ANN) will have millions of units or processing elements, which form an interconnected network and process a number of information [2].

DNN follows the structure of a typical ANN with a complex network model. This helps in modeling and defining complex hierarchies in simple forms [2]. The DNN method is one of the newest and best machine learning techniques for pattern recognition. DNN consists of two or more layers of independent learning (hidden layer), with the weight of units that are fully connected between two adjacent layers can be learned automatically. The hidden layer gradually will map the low-level features that are not organized into a high-level data representation [5].

Research by Waspada et al. (2017) compared the performance of several supervised machine learning methods such as Decision Tree, Naïve Bayes, NN, and DNN for cancer classification which results in the accuracy of the method's performance. While this study uses the best architectural selection in the form of a combination of ReLU and / or ELU activation functions, learning rate, and the number of hidden layers with the DNN method for breast cancer classification, the results of which are accuracy, specificity, and sensitivity.

2.3.1. The architecture of the DNN method. DNN is considered as a set of nerve cells in a pile. DNN has some kind of coating that consists of n number of units in each of its layers. Architecture DNN has one input layer (input layer) and allows two or more layers of hidden (hidden layer) which are directly connected to the output layer (output layer) with each unit connected is assigned a weight value [2]. DNN network architecture can be seen in Figure 1.

![Figure 1. DNN architecture](image_url)
2.3.2. Exponential Linear Units (ELU) activation function. Exponential Linear Units (ELU) function with 0 < α is shown in equation 1.

\[
f(x) = \begin{cases} 
  x & \text{if } x > 0 \\
  \alpha(\exp(x) - 1) & \text{if } x \leq 0
\end{cases}
\]  

(1)

ELU has extra alpha constants that must be positive numbers. ELU has an average activation approaching 0 which is an exponential function. ELU acts like ReLU if x is positive, but for negative values, its function is limited by a fixed value of -1 to α = 1. This behavior helps encourage the average activation of neurons near zero which is useful for learning and helps learn stronger representations of noise.

2.3.3. Rectified Linear Units (ReLU) activation function. Rectified Linear Units (ReLU) is an activation function introduced by [10]. ReLU works by using a threshold value of 0 [11]. This function is shown in equation 2.

\[
f_{\text{ReLU}}(x) = \max(0, x)
\]  

(2)

The ReLU function will output 0 when x < 0, and ReLU will produce a linear function when.

2.3.4. Sigmoid Activation Function. Sigmoid function transforms the range of values from xinput to between 0 and 1 [12]. \(S(x)\) The Sigmoid activation function is shown in equation 3.

\[
S(x) = \frac{1}{1+e^{-x}}
\]  

(3)

Information:
\(S(x)\) = Sigmoid function from x
\(e\) = epsilon
\(S(x)\) will produce a curve in the range 0-1 on the y-axis. If x is a very large positive number, then the value \(e^{-x}\) has a value close to 0 which produces an output close to 1. Whereas if x a negative number is very large, then the value \(e^{-x}\) has a large value and produces an output close to 0.

2.3.5. Dropout. Dropout is a technique to prevent overfitting. Dropout refers to the removal of units (hidden or not) in a network. Dropout temporarily removes one unit along with all incoming and outgoing connections on that unit. The selection of which units to dropout will be done randomly [6]. Figure 2. is an illustration of Dropout in Neural Networks (NN).

![Figure 2. Model dropout in NN. (a) NN before dropout, (b) NN after dropout [6].](image)
2.3.6. Stochastic Gradient Descent (SGD). Methods Stochastic Gradient Descent (SGD) aims to minimize the empirical risk on a model of the way of calculating the derivative function repeatedly in a single training sample and update the corresponding model parameter [13]. SGD function is shown in equation 4.

\[ x' = x - \alpha(\Delta f(x)) \]  

\( x' \) = new value  
\( x \) = old value  
\( \alpha \) = learning rate  
\( \Delta f(x) \) = result of function change of \( f(x) \)

2.3.7. DNN training. DNN training consists of two phases, namely the forward propagation phase and the backward propagation phase. In the forward propagation phase, the value in the input layer is calculated until the output layer produces a lost value using the activation and Dropout functions. In backward propagation, the value of the change in weight is calculated using the formula for the derivative function in forward propagation. Weight changes are calculated by the SGD method. The forward and backward propagation phases will be repeated until the stop condition i.e. the predetermined number of epochs is met.

DNN method architecture that will be trained to apply the ReLU or ELU activation function for each hidden layer and Sigmoid activation function in the output layer. The training algorithm for the DNN method with two hidden layers that applies vectorization is as follows:

1. Step 0: Initialize the weights with small random numbers, specify epoch = 1, the learning rate (\( \alpha \)), and the number of hidden layers.
2. Step 1: If the termination condition is not met, i.e. epoch \( \neq \) maximum epoch, then go to step 2.
3. Forward Propagation Phase
4. Step 2: Each input unit receives a signal and passes it to the hidden unit.
5. Step 3: Each output value at the layer is denoted as \( A[l] \), with \( l = 0, 1, ..., p \) and output value unit \( i \) on layer \( l \) is denoted as \( a_i[l] \), with \( i = 1, 2, ..., j \). Calculate the output value on the first hidden layer until the last hidden layer with the following steps [14]:
   a. Calculate the value of input weights for \( m \) training data using equation 5.
      \[ Z[l] = W[l], A[l-1] + b[l] \]  
   b. Calculate the output value on the hidden layer with the ReLU activation function using equation 6 or with the ELU activation function using equation 7.
      \[ A[l] = f_{ReLU}(Z[l]) = \max(0, Z[l]) \]  
      \[ A[l] = f_{ELU}(Z[l]) = \begin{cases} Z & \text{if } Z > 0 \\ \alpha(\exp(Z) - 1) & \text{if } Z \leq 0 \end{cases} \]  
1. Step 4: Calculate the output value at the output layer with the following steps [14]:
   a. Calculate the weight value on the output layer using equation 8.
      \[ Z[p] = W[p], A[p-1] + b[p] \]  
   b. Calculate the output value at the output layer with the Sigmoid activation function using equation 9.
      \[ A[p] = \sigma(Z[p]) = \frac{1}{(1+e^{Z[p]})} \]  
2. Step 5: Calculate the error value at the output layer with binary cross-entropy using equation 10.
\[
L(A^p, Y) = -\frac{1}{m} \sum_{i=1}^{m} ((y_i \log a_i) + ((1 - y_i) * (\log(1 - a_i))))
\]  

(10)

Backward propagation phase

3. Step 6: Calculate the value of the change in parameter weights and at the output layer by deriving the function in the forward propagation phase with the following steps [14]:
   a. Calculate the gradient of the loss value at the output layer using equation 11.
   \[
dA^p = \frac{dL(a,y)}{dA^p}
\]  

(11)
   b. Calculate value of \(dZ^p\) on the output layer using equation 12.
   \[
dZ^p = dA^p * \sigma'(A^p)
\]  

(12)
   c. Calculate the value of the parameter derivative \(W\) using equation 13.
   \[
dW^p = \frac{1}{m} * dZ^p * A^{p-1}\top
\]  

(13)
   Calculate the value of the parameter derivative \(b\) using equation 14.
   \[
   db^p = \frac{1}{m} * \sum_{i=1}^{m} dZ^p
\]  

(14)
   d. Calculate the output value at the output layer using equation 15.
   \[
dA^{p-1} = W^p \top * dZ^p
\]  

(15)

4. Step 7: After getting the values from the parameter derivative \(W\) and \(b\), calculate the change in parameter weights using the SGD method using equation 16 and equation 17.
   \[
   W'[p] = W^p - \alpha(dW^p)
   \]  

(16)
   \[
   b'[p] = b^p - \alpha(db^p)
   \]  

(17)

5. Step 8: Calculate the rate of change of weights in the hidden layer to last until the hidden layer first by reducing the propagation phase advanced functions with the following steps:
   a. Calculate the value of \(dZ^{p-1}\) on the hidden layer using equation 2.18.
   \[
dZ^{p-1} = dA^{p-1} * f'_{ReLU/ELU}(A^{p-1})
\]  

(18)
   b. Calculate the value from parameter derivative \(W\) and \(b\) using equations 13 and 14.
   c. Calculate the output value in the hidden layer using equation 15.

6. Step 9: After getting the value of the derived parameter and, calculate the weight change parameters in the hidden layer with the SGD method using equation 16 and equation 17. Perform calculations to the hidden layer first.

7. Step 10: If the termination condition is not met, i.e. if epoch \(\neq\) maximum epoch, repeat the calculation starting from step 1.

2.3.8. DNN testing. The testing process is done by classifying test data, then the outputs are compared with the actual target. The steps for DNN testing are as follows [14]:
   1. Step 1: Enter all input values of \(X\).
   2. Step 2: Determine the weight value using the final weight value obtained from the training results.
   3. Step 3: Perform forward propagation phase calculations, i.e. calculating the output value in the hidden layer using equation 5 and equation 6 or equation 7. Calculate the output value at the output layer using equation 8 and equation 9. Generate an error value from the test data by calculating using equation 10.
4. Step 4: With a threshold do a threshold on the final result of the error value. The threshold used is 0.5. If $y > 0.5$ then the output value is 1 and vice versa if $y \leq 0.5$ then the output value is 0.

2.4. System evaluation
The evaluation system used in this study is the confusion matrix. A confusion matrix is a method used to calculate accuracy in the field of data mining. Confusion matrix will later be performing calculations that perform four main outputs, namely recall (proportion of positive cases correctly identified), precision (proportion of cases with positive results are correct), accuracy (comparison cases identified correctly by the total number of cases) and error rate (cases identified incorrectly with a total number of cases).

In this study, three outputs are used, namely sensitivity (recall), specificity (the proportion of negative cases correctly identified), and accuracy which are the basis for calculating accuracy in the health field (Zhou, et al., 2009). Table 1. shows the calculation of the infusion matrix based on predicted classes and actual classes.

| Table 1. Confusion matrix |
|---------------------------|
| **Actual class** | **Predicted class** |
| C1 | C2 |
| C1 | TP | FN |
| C2 | FP | TN |

1. True Positives (TP)
   C1 data (positive) predicted as C1 (positive)
2. True Negatives (TN)
   C2 data (negative) predicted as C2 (negative)
3. False Positives (FP)
   C2 data (negative) predicted as C1 (positive)
4. False Negatives (FN)
   C1 data (positive) predicted as C2 (negative)

The following is the formula for calculating accuracy, sensitivity, and specificity.

1. **Accuracy**
   The accuracy value is the number of correct predictions divided by the total number of cases. The formula for calculating accuracy can be seen in equation 19.
   \[
   \text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (19)
   \]

2. **Sensitivity**
   The sensitivity value is the ratio between the correct prediction of a positive class and the number of positive actual classes. The sensitivity calculation formula can be seen in equation 20.
   \[
   \text{Sensitivity} = \frac{TP}{TP + FN} \quad (20)
   \]

3. **Specificity**
   The specificity value is the ratio between the correct prediction of the negative class and the number of actual negative classes. The formula for calculating specificity can be seen in equation 21.
   \[
   \text{Specificity} = \frac{TN}{TN + FP} \quad (21)
   \]

3. Research method
Implementation of the best architectural selection using the DNN method for the classification of breast cancer using MicroRNA data is carried out in several stages. An outline diagram of the problem resolution is shown in Figure 3.
The stages of implementation begin with data collection, mapping data into input and output data, division of data into training data and test data, training using the DNN method, DNN testing, evaluation, and conclusions. The selection of the best architecture is done at the DNN training and testing stage by selecting the activation function, the number of dropouts, the optimization method, and the number of epochs. Evaluation of the results of DNN tests conducted by selecting various architectures will provide results that will be used later at the conclusion stage. At this conclusion stage, it will be known as the best architectural selection based on the results of the evaluation obtained in the previous stage.

3.1.1. Data collection.
Data collection for this study was conducted with data retrieval MicroRNA obtained from GDC Data Portal National Cancer Institute (https://portal.gdc.cancer.gov/).

Each data has 1881 attributes and 17 attributes that will mostly take effect on breast cancer will be taken based on research conducted by Lan, et al. (2015) and label the data value of 0 for normal and 1 for cancer. The description of the attributes and labels can be seen in Table 2.

![Diagram of the problem resolution](image)

**Figure 3. Diagram of the problem resolution**

| No | Attribute       | Type |
|----|----------------|------|
| 1. | hsa-mir-10b    | Numerik |
| 2. | hsa-mir-21     | Numerik |
| 3. | hsa-mir-30a    | Numerik |
| 4. | hsa-mir-92a-1  | Numerik |
| 5. | hsa-mir-92a-2  | Numerik |
| 6. | hsa-mir-125b-1 | Numerik |
| 7. | hsa-mir-125b-2 | Numerik |
| 8. | hsa-mir-141    | Numerik |
| 9. | hsa-mir-145    | Numerik |
| 10.| hsa-mir-200a   | Numerik |
| 11.| hsa-mir-200b   | Numerik |
| 12.| hsa-mir-155    | Numerik |
| 13.| hsa-mir-191    | Numerik |
| 14.| hsa-mir-203a   | Numerik |
| 15.| hsa-mir-203b   | Numerik |
Data is divided into input data consisting of MicroRNA features and output data consisting of label data. There is 1200 total with two classes of data in the form of a label, which is a class of data 412 cancer (breast cancer) with label 1 and 788 class data of normal (healthy) with the label 0. The data input consists of 17 attributes MicroRNA and data output as label data, the example of data that users can be seen in Table 3.

| No | hsa-mir-10b | hsa-mir-21 | … | hsa-mir-210 | hsa-mir-373 | label |
|----|-------------|------------|----|-------------|-------------|-------|
| 1  | 113225      | 467274     | … | 658         | 0           | 1     |
| 2  | 268413      | 292500     | … | 376         | 0           | 1     |
| 3  | 131944      | 3603126    | … | 293         | 0           | 1     |
| 4  | 219237      | 771427     | … | 22          | 0           | 1     |
| 5  | 127604      | 740009     | … | 2156        | 1           | 1     |
| …  | …           | …          | … | …           | …           | …     |
| 1196| 657754      | 167632     | … | 251         | 0           | 0     |
| 1197| 1708199     | 301647     | … | 4438        | 0           | 0     |
| 1198| 1915120     | 122887     | … | 360         | 0           | 0     |
| 1199| 947790      | 242173     | … | 41          | 0           | 0     |
| 1200| 647400      | 266967     | … | 187         | 0           | 0     |

3.2. Division of data into training data and test data
The next stage is the division of data into training data and test data. Training data is data used for the training process, while test data is data used for the testing process.

3.3. DNN architecture
The architecture used in this study consists of two architectures. The first architecture is the DNN method architecture using 200 hidden units, 30% dropout, 150 epochs, and \( p - 1 \) the number of hidden layers in layers \( l \) with \( p = 2 \) and the number of layers in the range 3-4 and \( k \) as value of learning rate \( (\alpha) \), with \( k = 0.01, 0.02, 0.03, 0.04, 0.05, 0.001, 0.0001 \). The general architecture of the DNN method can be seen in Figure 4.

3.4. DNN training and testing
In the discussion of the DNN method, it is assumed that there are two data with two features as input and using the DNN architecture which has two hidden layers with two hidden units each, the learning rate \( (\alpha) \) of 0.01, no dropout and 1 epoch. Phase training on methods DNN shortly notices training data is divided into two phases, namely the propagation of forwarding and backward propagation. Whereas in the testing process, the calculation is only done at the forward propagation and calculation accuracy steps. The testing process is carried out using the final weights of the training results.

4. Results
4.1. Scenario 1
This test is carried out using the DNN method with 200 hidden units, a 30% dropout rate, and 150 epochs. This testing process is carried out using several learning rate values \((\alpha)\) and
3 hidden layers. Learning rate value ($\alpha$), conducted an experiment using a value of 0.01, 0.02, 0.03, 0.04, 0.05, 0.01, 0.001, and 0.0001 which requires 32 tests.

The learning rate parameter controls the speed or pace of the learning model so that when the learning rate is correctly configured, the model will learn to estimate the best function with the available resources (number of layers and number of nodes each layer) in a certain number of training periods. In general, a large learning level allows the model to learn faster but will produce too much weight and the performance of the model will go far beyond the training data. When the learning level is too large, gradient descent can increase training errors rather than reduce them. In the worst case, a weight update that is too large can cause the weight to explode (for example resulting in numerical overflow). The smaller learning allows the model to learn weights that are more optimal but require more time to be trained. When the level of learning is too small, training is not only slower but may become permanently stopped with high training errors. In the first scenario in this study, it was found that the appropriate learning rate is 0.04.

The activation function is a mathematical equation that determines the output of a neural network. This function is attached to each neuron in the network and determines whether it must be activated or not based on whether each neuron's input is relevant for the prediction of the model. The activation function also helps normalize the output of each neuron to a range between -1 and 1. A neural network has only one hidden layer that processes the input and the output layer that provides the final output of the model. Whereas the Deep Neural Network (DNN) generally has between 2-8 additional neuron layers. This activation function is a mathematical gate between the input which gives the current neuron value and the result goes to the next layer. The gradient of the activation function is very important to train neural networks. Neural networks are trained using a process called backpropagation, this is an algorithm that traces the output of the model back through the various neurons involved in producing output and then returns to the original weight applied to each neuron. Backpropagation shows the optimal weight for each neuron that produces the most accurate predictions. The difference in the activation function is in the range depending on the input data. In the first scenario in this study, it was found that the appropriate combination of activation functions was a combination of ReLU, ReLU, ReLU.

Each test with a combination of different activation functions and various learning rate values is calculated for their accuracy, specificity, and sensitivity. Based on Table 4 it is known that the combination of the activation functions of ReLU, ReLU, ReLU with a learning rate of 0.04 produces the highest accuracy value, which is 94.58% with a specificity of 96.45% and sensitivity of 91.02%. This architecture is the best architecture of the DNN method in breast cancer dataset in this study. The test results are shown in Table 4.

| Activation Function | Confusion Matrix(%) | Learning Rate ($\alpha$) |
|---------------------|---------------------|-------------------------|
|                     | 0.01    | 0.02    | 0.03    | 0.04    | 0.05    | 0.001   | 0.0001  | 0.0000  |
| ReLU                | 94.42   | 94.00   | 93.75   | 94.58   | 94.33   | 94.50   | 93.50   | 65.67   |
| ReLU, ReLU          | 96.45   | 95.68   | 95.56   | 96.45   | 95.94   | 96.45   | 95.43   | 100.00  |
| ReLU, ReLU          | 90.53   | 90.77   | 90.28   | 91.02   | 91.26   | 90.77   | 89.80   | 0.00    |
| ReLU, ELU           | 94.17   | 93.83   | 93.58   | 93.50   | 93.66   | 93.58   | 93.50   | 65.67   |
| ReLU, ELU           | 96.19   | 95.56   | 95.56   | 95.56   | 95.81   | 95.43   | 95.43   | 100.00  |
| ReLU, ELU           | 90.29   | 90.53   | 89.80   | 89.55   | 89.55   | 90.05   | 89.80   | 0.00    |
| ReLU, ELU           | 92.91   | 93.17   | 93.08   | 92.67   | 90.48   | 92.91   | 93.50   | 72.08   |
| ReLU, ELU           | 95.56   | 95.68   | 95.68   | 96.07   | 95.56   | 95.56   | 95.56   | 99.24   |
| ReLU, ELU           | 87.86   | 88.34   | 88.10   | 86.89   | 79.80   | 87.86   | 89.56   | 20.13   |
| ReLU, ELU           | 93.58   | 93.41   | 93.33   | 93.25   | 91.74   | 93.25   | 93.58   | 67.42   |
| ReLU, ELU           | 95.68   | 95.68   | 95.56   | 95.68   | 95.94   | 95.56   | 95.43   | 99.75   |

Table 4. Results hidden layer 3 on DNN
Table 5. Results hidden layer 4 on DNN

| Activation Function | Confusion Matrix (%) | Learning Rate (α) |
|---------------------|----------------------|-------------------|
| ReLU, ReLU, ReLU, ReLU | 0.01 0.02 0.03 0.04 0.05 0.001 0.0001 0.0000 | 1 1 1 1 1 1 1 1 |
| ReLU, ReLU | 93.83 93.66 93.75 93.58 93.91 93.66 93.67 65.67 |
| ReLU, ReLU | 95.56 95.30 95.43 94.79 95.30 95.43 95.43 100.00 |
| ReLU, ReLU | 90.53 90.53 90.53 91.26 91.26 90.29 90.29 0.00 |
| ELU, ELU | 93.75 93.66 93.50 93.16 93.67 93.83 93.75 65.84 |
| ELU, ELU | 95.81 95.81 95.56 94.93 95.56 95.43 95.43 100.00 |
| ELU, ELU | 89.80 89.55 89.56 89.80 90.04 90.77 90.53 0.49 |
| ELU, ELU | 93.41 93.16 92.83 66.56 53.06 93.41 93.75 77.16 |
| ELU, ELU | 95.56 95.81 96.07 78.61 60.00 95.56 95.56 98.61 |
| ELU, ELU | 89.31 88.09 86.63 43.69 40.00 89.31 90.29 36.13 |
| ELU, ELU, ReLU | 93.83 93.83 93.58 93.83 93.58 93.66 93.75 66.08 |
| ELU, ELU, ReLU | 95.56 95.43 95.56 95.61 94.80 95.43 95.43 100.00 |
| ELU, ELU, ReLU | 90.53 90.77 89.79 90.53 91.25 90.28 90.53 1.22 |

Figure 4. Accuracy hidden layer 3 on DNN

Figure 5. Accuracy hidden layer 4 on DNN

4.2. Scenario 2
This test is carried out using the DNN method with a total number of hidden units of 200, 30% dropout rate and 150 epochs. The testing process is carried out using several learning rate values (α) and 4 hidden layers. Learning rate value (α), conducted an experiment using a value of 0.01, 0.02, 0.03, 0.04, 0.05, 0.01, 0.001, and 0.0001 which requires 32 tests.

Each test with a combination of different activation functions and various learning rate values is calculated for their accuracy, specificity, and sensitivity. Based on Table 5 it is known that the combination of the activation functions of ReLU, ReLU, ReLU, ReLU with a learning rate of 0.05 produces the highest accuracy value, which is 93.91% with a specificity of 95.43% and sensitivity of 90.28%. The test results are shown in Table 5.

5. Conclusion and suggestion
5.1. Conclusion
Based on The Best Architecture Selection with Deep Neural Network (DNN) Method for Breast Cancer Classification Using MicroRNA Data can be concluded that:

1. DNN method with 3 hidden layers in the combination of the activation function ReLU, ReLU, ReLU with a learning rate of 0.04 produces the highest accuracy value, which is 94.58% with a specificity of 96.45% and a sensitivity of 91.02%.

2. The DNN method with 4 hidden layers in the combination of the ReLU, ReLU, ReLU, ReLU activation functions with a learning rate of 0.05 produces the highest accuracy value, which is 93.91% with a specificity of 95.43% and a sensitivity of 90.28%.

3. The best architectural selection using breast cancer MicroRNA data on the DNN method with the architecture in the form of the number of hidden layer 3 in the combination of the activation functions of ReLU, ReLU, ReLU and the learning rate of 0.04 obtained the highest accuracy of 94.58% with specificity 96.45% and sensitivity 91.02%.

5.2. Suggestion
Suggestions that can be given for further research can be focused on the application of other optimization algorithms such as the RMS Prop or Adam optimization algorithm with the DNN method using normalization to get the best architecture. Future research is aimed at finding out whether other optimization algorithms can better influence the architecture and performance of the DNN method when compared to using normalization.

References
[1] Society A C 2017 Cancer Facts and Figures 2017-2018 Atlanta
[2] Karthik S, Perumal R S and Chandra P V S S R 2018 Knowledge Computing and Its Applications pp. 227-241
[3] Lan H, Lu H, Wang X dan Jin H 2015 BioMed Research International vol. 2015 pp. 1-17
[4] Madhavan D, Cuk K, Burwinkel B and R Yang 2013 Frontiers in Genetics vol. 4, no. 116, pp. 1-13
[5] Spencer M, Eickholt J and J Cheng 2015 IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB) vol. 12 no. 1 pp. 103-112
[6] Srivastava N, Hinton G, Krizhevsky A, I Sutskever and R Salakhutdinov 2014 Journal of Machine Learning Research vol. 15 pp. 1929-1958
[7] Ioffe S and Szegedy C 2015 arXiv preprint arXiv:1502.03167
[8] Pal M K, Jaiswar S P, Dwivedi V N, Tripathi A K, Dwivedi A and P Sankhwar 2015 Cancer Biol Med. vol. 12 pp. 328-341
[9] Ahmad A 2017 Jurnal Teknologi Indonesia pp. 1-5
[10] Hahnloser R H, Sarpeshkar R, Mahowald M A, Douglas R J and Seung H S 2000 Nature 405 vol. 6789 no. 2000 p. 947
[11] Agarap A F M 2018 arXiv preprint arXiv:1803.08375
[12] Balaji S A and K Baskaran 2013 IJCSEIT vol. 3 pp. 13-31
[13] Hardt M, Recht B and Y Singer 2015 arXiv preprint arXiv:1509.01240
[14] Ng A Y -T deeplearning.ai Online Available https://www.youtube.com/channel/UCclIXc5mJshVY7TZR1maL5l9w/playlists Diakses 9/7/2018