STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
*Stat., Optim. Inf. Comput.*, Vol. **14**, July 2025, pp 229–246.
Published online in International Academic Press (www.IAPress.org)

# Application of RBF Neural Network in Predicting Thalassemia Disease in Mosul City

Mohammed Faris Ali*, Hutheyfa Hazem Taha

*Department of Statistics and Informatic, College of Computer Science and Mathematics, University of Mosul, Mosul, Iraq*

**Abstract** Thalassemia is a hereditary blood disorder that can be born to children if both parents are carriers of the gene mutation. The effect of this mutation is a faster than normal rate of destruction of red blood cells and thus iron accumulation and decreased availability of hemoglobin, The quantity, quality and shape of red blood cells are also reduced. In the current dataset, the number of severe thalassemia = 131, moderate thalassemia = 149 and 13 variables were used. A sample taken from Al-Hadbaa Specialized Hospital for Hematology and Bone Marrow Transplantation in Mosul, Iraq was used. In this study, before applying Python to this model, these variables were data cleaned to remove any gaps. The data was divided into several sections using cross-validation to test the model. Radial Basal Function (RBF) networks were used in this study to classify thalassemia patients with respect to the specified model performance measurement criteria. The experimental results revealed that RBF networks performed well with test accuracy of 96%; F1 score of 96%; high sensitivity of 97%; high specificity of 95%; and a high positive predictive value of 95%. The resulting area under the curve was 99.5%, which is very close to the ideal for the sample. Through experiments, we found that the best setting is a learning rate of 0.1 and sixteen neurons in the hidden layer. Furthermore, a random forest model was used to identify the most significant features influencing the differentiation between types of thalassemia. The results showed that the most important features are HBA1 (adult hemoglobin) and HBF (fetal hemoglobin), which represent the main indicators for determining the type of thalassemia due to their significant impact on classification. This is followed by the HB (total hemoglobin) feature as a third important feature, and then growth delay and HBA2 with varying degrees of importance. These analyses helped identify the fundamental factors associated with the genetic and clinical differences between major thalassemia and intermediate thalassemia, contributing to enhancing the understanding of the precise classification of the disease and improving diagnostic and treatment strategies.

**Keywords** Machine learning, Thalassemia disease, RBF Model

**AMS 2010 subject classifications** 68-Txx, 62-07

**DOI:** 10.19139/soic-2310-5070-2303

## 1. Introduction

Thalassemia is the second most common genetic disease after sickle cell anemia, it is a genetic disease in the world and is more prevalent in the Mediterranean region, the Middle East and Southeast Asia, Thalassemia is caused by a defect in the production of a protein called hemoglobin, which plays a crucial role in transporting oxygen in the blood, As a result, people with thalassemia suffer from chronic anemia and other health problems such as (enlarged spleen, iron overload, congestive heart failure, osteoporosis and delayed growth). Such accurate measurements are now of great importance in medicine and diagnosis in general, but they are especially necessary when trying to distinguish between one type of thalassemia and another or determine the severity of the disease, The importance of classification lies in its ability to diagnose the correct type from the given unclassified data samples

*Correspondence to: Mohammed Faris Ali (Email: mohammed.22csp63@student.uomosul.edu.iq). Department of Statistics and Informatic, College of Computer Science and Mathematics, University of Mosul, Mosul, Iraq.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

and predict the same type for new samples that were not included in the training phase, as well as to recognize other categories that were not included in the training phase, Artificial intelligence methods have previously helped in better classification of thalassemia and helped in faster and better classification diagnosis. This research includes a supervised learning strategy to classify thalassemia patients; Hence, it requires consideration of the target variable, The rationale for using radial basis function based neural networks in the case of nonlinear functions is to obtain an estimate of these functions to allow efficient classification of the data. Furthermore, to reduce the error rate and improve the classification accuracy, Adam optimization algorithm is used.

## 2. Literature review

It should only be noted here that the synthesis of the research works that have been conducted earlier is essential in the evaluation of the characteristics of the nature of scientific research development because as prior accomplishments are acknowledged and previous chests established, one stands a better chance of to build toward the future. In the present research, therefore, we focused more on only those previous researches which we determined to be most related. The study conducted by [1] aimed to design an AI-based system that enables early detection of fires occurring in both indoor and outdoor spaces. Data was collected from two different types of smoke detectors based on the Internet of Things. In this research, 7 machine learning algorithms were used: RBF, MLP, NB, KNN, RF, DT, LMT. The results indicated that the Random Forest algorithm outperformed the other methods in terms of accuracy, achieving 99.98%, demonstrating a significant improvement in fire detection capabilities. To diagnose thyroid diseases, researchers [2] used data from a machine learning repository that included 7,200 patients, classified according to the patient's condition into hyperthyroidism, hypothyroidism, or normal. The methods used included neural networks, such as the radial basis function network and the multilayer perceptron. The RBF was trained using the k-means clustering algorithm. The results showed that the combination of RBF and MLP achieved a high accuracy rate of 97%. This indicates the effectiveness of the proposed method in diagnosing thyroid diseases. The study conducted by [3] aimed to establish a new indicator system for assessing the balance between economic benefits and water pollution protection in institutions. The sample used consisted of 241 institutions, and in this research, the Kmeans-RBF neural network was employed. The results showed that the institutions were effectively classified into three categories, with strong stability of the classifier. The new indicator system contributes to the effective assessment of the economic and environmental balance of institutions, supporting efforts made in environmental protection. The study conducted by [4] aimed to create an effective model using neural network techniques for classifying flight delays. The study involved the use of two neural network techniques, namely Radial Basis Function Network and Backpropagation Network. The data was divided into 70% for training and 30% for testing. The results indicate that RBF neural networks can be trained much faster than simple neural networks. The lowest training error was achieved using RBFN. The study conducted by [5] aimed to identify the factors influencing the Gleason score for prostate cancer. Data from 97 patients were analyzed using neural network models, specifically the radial basis function network and the multilayer perceptron. The data was divided in a 70:30 ratio for training and testing. The multilayer perceptron model achieved the best performance with an accuracy of 96% and a sensitivity of 93.3%. The results demonstrated the effectiveness of artificial intelligence in classifying patients and identifying important indicators related to the disease. The study conducted by [6] aimed to apply machine learning models to predict diabetes and analyze the performance of these models. Machine learning models were used, including SVM, KNN, MLP, and RBF. The experimental results showed that the KNN classifier outperformed the other models with an accuracy of 98%. In a machine learning work by [7], potential blood donors and non-donors were distinguished with 748 subjects sourced from blood transfusion centers in Iran. Several classification methods were applied such as Bayesian Knife, K-NN, MLNN, and RBF, and hence the result shows that compared to all the other classification techniques the RBF network gave better efficiency and accuracy of 93%. A classification system of red blood cells of patients with thalassemia and normal patients was developed by [8] using digital image processing. The study was done with an objective of reducing examination time and avoiding subjectivity. To extract features, they collected 7,108 images of red blood cells, ranging from type 1 to type 9. Four approaches were used in this study to classify red blood cells: MLP,

Naive Bayes, RBF and SVM. From the experiment, MLP was seen to have the highest results, with the following results; recall=89.6%, accuracy=89.3%, precision = 89.6%. Also, the second and third classifiers were the support vector machine and the Naive Bayes classifier respectively. The classification of red blood cells for anemia (ID) and thalassemia (TH) was the main focus of the research conducted by [9]. Image processing techniques were used to enhance edge detection and reduce noise in the classified red blood cells. The researchers examined the efficiency of five different machine learning classifiers. These classifications were as follows: logistic regression (LR), radial basis function network (RBF), multilayer perceptron (MLP), naive Bayes classifier (NB), and classification tree (CART). The best performance was achieved through logistic regression, which reached an accuracy of 83.5%, a sensitivity of 83.5%, and a predictive value of 83.3%. Research was undertaken in Iran utilising Zafer data to predict diabetes in thalassemia patients [10], Using 256 observations, scientists were able to detect those at risk for beta-thalassemia and avoid the effects of iron excess, such as diabetes. The researchers used both the radial basis function network (RBFN) and the k-nearest neighbor (KNN) approach. The RBFN methodology produced a high accuracy of 81.7%, whereas the KNN method achieved 69.12%. Complete blood counts (CBCs) were used in research by [11] to distinguish thalassemia patients from healthy people. 304 clinical data of public-school students who had thalassemia screenings at Ozeri Hospital were utilised. As machine learning methods, they used radial basis function neural networks (RBFs), probabilistic neural networks (PNNs), and closest neighbor algorithms. 100% of $\beta$-thalassemia patients, 93% of normal people, and 93% of $\alpha$-thalassemia carriers were correctly diagnosed by RBF. That means RBF performs better than both PNN and KNN in thalassemia patients and healthy people.

## 3. Methodology

### *3.1. Radial basis function network*

It is a feed-forward neural network commonly used for time series prediction, function approximation, classification, and control. It is also utilized to handle nonlinear issues owing to its straightforward structure and its capacity to carry out the learning process in a way that is both smooth and understandable. There are a number of notable benefits associated with RBFN, including excellent dependability, quick convergence, and minimal mistakes [15]. The radial base function network consists of three layers, as shown in Figure 1.

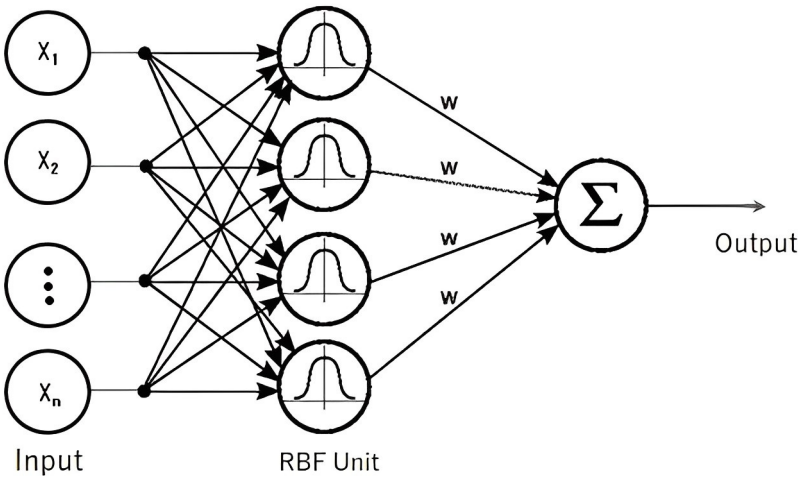



Figure 1. The structure of the radial basis function (RBF) network.

The input layer is the one that is given the vector of independent variables $X_i = [x_1, x_{2,\ldots} x_n]^T$,where $i = 1, 2, \ldots n$ and $n$ stands for the total number of observations, and there is only one hidden layer; each neuron in the hidden layer employs a radial basis function as a nonlinear activation function for the input. This function

computes the Euclidean distance between the input $X$ and the center of each neuron $C$ [16]. As in the equation:

$$r = \|X_i - C_j\|$$

To identify the hidden layer, a radial basis function is used, which transforms the data into a nonlinear form. The radial basis activation function uses a Gaussian equation to calculate the output of the hidden layer, as in the following formula:

$$P_j(x) = \exp\left(\frac{\|X_i - C_j\|^2}{2\sigma_j^2}\right), \sigma > 0 \tag{1}$$

Where $C_j = [c_1, c_2, \ldots c_m]^T$ represents the vector of centers of the radial basis function, $\sigma_j = [\sigma_1, \sigma_{2,\ldots}\sigma_m]^T$ represents the vector of standard deviation of neurons, and $m$ represents the number of neurons in the hidden layer [17]. The artificial neural network $Sm(x)$ produces its results by multiplying the outputs of the hidden layer $P_j(x)$ by the weights between the output layer and the layer that is hidden. The following equation illustrates this

$$S_m(x) = \sum_{j=1}^{m} w_{j^k} P\|X_i - C_j\| + wo \tag{2}$$

The radial basis function is denoted as $P\|X_i - C_j\|$ , the weights between the hidden and output layers are represented by $w_j$, where $W = [w_1, w_2, \ldots w_m]^T$, and the bias value is represented by $w_0$ [18].

*3.1.1. Radial basis function artificial neural network (RBF) training* At the beginning of the RBF training process, there are two stages: For the first stage, which identifies the number of clusters and standard deviation, it is an unsupervised learning area, but after the identification of the centers of the clusters in the second stage, using gradient descent to find the best weights at which learning becomes supervised.

1. Unsupervised learning of radial basis function: This learning technique uses the radial basis training to the network in the hidden layer. Swapping of the positions of the center $C_j$ and the standard deviation $\sigma_j$ is the general objective of this kind of learning.
2. This type of learning is performed in the output layer and is called supervised learning of the radial basis function, When training the radial basis function parameters in the hidden layer containing the centers and standard deviation, the gradient descent technique is applied in training the weight used in the network between the hidden layer and the output layer, so the overall purpose of this training will be to pretty much push the cost function as close to zero as possible [19].

In this research, we used the logistic regression cost function, or the so called binary cross-entropy, in the case of binary classification, as in the following formula:

$$L(w, c, \sigma) = -\frac{1}{n}\sum_{i=1}^{n}\left[y^i \log\left(R\left(x^{(i)}\right)\right) + \left(1 - y^i\right)\log\left(1 - R\left(x^{(i)}\right)\right)\right] \tag{3}$$

Where n is the number of samples being used for training, $y^i$ represents the actual classification of sample $i$, which is either (0 or 1), while $R\left(x^{(i)}\right)$ represents the expected output of the activation function Sigmoid for sample i, as in the following formula:

$$R\left(x^{(i)}\right) = \frac{1}{1 + \exp^{\left(-\ S_m\left(x^{(i)}\right)\right)}} \tag{4}$$

where $S_m\left(x^{(i)}\right)$ represents the expected output RBFN.

*3.1.2. Estimation of RBFN parameters* To estimate the parameters of the radial basis function network, we derive the cost function for each parameter of the model as follows:

1. We derive the cost function binary cross entropy with respect to the weights by applying the chain role rule as in the following formula:

$$\frac{\partial L}{\partial w_{jk}} = \frac{\partial L}{\partial R\left(x^{(i)}\right)} \cdot \frac{\partial F\left(x^{(i)}\right)}{\partial S_m\left(x^{(i)}\right)} \cdot \frac{\partial R_m\left(x^{(i)}\right)}{\partial w_{jk}} \tag{5}$$

$$\frac{\partial L}{\partial R\left(x^{(i)}\right)} = -\left[\frac{y^i}{R\left(x^{(i)}\right)} - \frac{1-y^i}{1-R\left(x^{(i)}\right)}\right] \tag{6}$$

$$\frac{\partial R\left(x^{(i)}\right)}{\partial S_m\left(x^{(i)}\right)} = R\left(x^{(i)}\right).\left(\mathbf{1} - R\left(x^{(i)}\right)\right) \tag{7}$$

$$\frac{\partial S_m\left(x^{(i)}\right)}{\partial w_{jk}} = P_j\left(x^{(i)}\right) \tag{8}$$

Then we substitute Equations 6, 7, and 8 into Equation 5 to obtain the following formula:

$$\frac{\partial L}{\partial w_{jk}} = -\left[\frac{y^i}{R\left(x^{(i)}\right)} - \frac{1-y^i}{1-R\left(x^{(i)}\right)}\right].R\left(x^{(i)}\right)\left(\mathbf{1} - R\left(x^{(i)}\right)\right).P_j\left(x^{(i)}\right) \tag{9}$$

After simplifying 9, By using the following formula, we can derive the final result of the cost function's derivative with respect to the weights:

$$\frac{\partial L}{\partial w_{jk}} = \sum_{i=1}^{n}\left[\left(R\left(x^{(i)}\right) - y^i\right) \; P_j\left(x^{(i)}\right)\right] \tag{10}$$

2. Deriving the cost function for centers according to the chain rule and as in the following formula:

$$\frac{\partial L}{\partial c_j} = \frac{\partial L}{\partial R\left(x^{(i)}\right)} \cdot \frac{\partial R\left(x^{(i)}\right)}{\partial S_m\left(x^{(i)}\right)} \cdot \frac{\partial S_m\left(x^{(i)}\right)}{\partial P_j\left(x^{(i)}\right)} \cdot \frac{\partial P_j\left(x^{(i)}\right)}{\partial c_j} \tag{11}$$

$$\frac{\partial P_j\left(x^{(i)}\right)}{\partial c_j} = \frac{x^{(i)} - c_j}{\sigma_j^2} \cdot P_j\left(x^{(i)}\right) \tag{12}$$

After simplifying the equation, we get the final result as in the following formula:

$$\frac{\partial L}{\partial c_j} = -\sum_{i=1}^{n}\left[\frac{y^i}{R\left(x^{(i)}\right)} - \frac{1-y^i}{1-R\left(x^{(i)}\right)}\right] R\left(x^{(i)}\right)\left(\mathbf{1} - R\left(x^{(i)}\right)\right).w_{jk}.P_j\left(x^{(i)}\right)\frac{x^{(i)} - c_j}{\sigma_j^2} \tag{13}$$

3. Deriving the cost function with regard to the standard deviation in accordance with the chain rule and using the formula that is shown below:

$$\frac{\partial L}{\partial \sigma_j} = \frac{\partial L}{\partial R\left(x^{(i)}\right)} \cdot \frac{\partial R\left(x^{(i)}\right)}{\partial S_m\left(x^{(i)}\right)} \cdot \frac{\partial S_m\left(x^{(i)}\right)}{\partial P_j\left(x^{(i)}\right)} \cdot \frac{\partial P_j\left(x^{(i)}\right)}{\partial \sigma_j} \tag{14}$$

$$\frac{\partial P_j\left(x^{(i)}\right)}{\partial \sigma_j} = P_j\left(x^{(i)}\right)\frac{\left\|x^{(i)} - c_j\right\|^2}{\sigma_j^3} \tag{15}$$

After simplifying the equation, we get the following formula:

$$\frac{\partial L}{\partial \sigma_j} = -\sum_{i=1}^{n}\left[\frac{y^i}{R\left(x^{(i)}\right)} - \frac{1-y^i}{1-R\left(x^{(i)}\right)}\right].R\left(x^{(i)}\right)\left(\mathbf{1} - R\left(x^{(i)}\right)\right).w_{jk}.P_j\left(x^{(i)}\right)\frac{x^{(i)} - c_j}{\sigma_j^3} \tag{16}$$

After calculating the partial derivatives mentioned above, we update the parameters as follows:

$$\Delta w_{jk} = w_{jk} - \eta_w \frac{\partial L}{\partial w_{jk}} \tag{17}$$

$$\Delta c_j = c_j - \eta_c \frac{\partial L}{\partial c_j} \tag{18}$$

$$\Delta \sigma_{\mathrm{j}} = \sigma_j - \eta_\sigma \frac{\partial L}{\partial \sigma_j} \tag{19}$$

Where $\eta_\sigma,\ \eta_c,\ \eta_w$ are the learning rates for each of the weights, standard deviation, and centers.

*3.1.3. Optimization algorithm* According to this paper, the Adam optimization technique is used to minimize the cost function and improve classification accuracy. The name Adam comes from adaptive moment estimation. Adam optimization algorithm is different from the classical stochastic gradient algorithm, which uses a standard stochastic gradient to update weights at a single learning rate ($\eta$) during the training period. In the Adam optimization algorithm, a separate learning rate is assigned to each parameter weight. So that the learning rate is adjusted and updated independently for each parameter. This algorithm chooses a smaller learning rate for the parameters that are updated periodically, and gives a larger learning rate for the corresponding parameters with low-frequency variables. Adam optimization algorithm combines the advantages of Adagrad algorithm and RMSprop algorithm [20]. In the Adam optimization algorithm, the first moment (mt) and the second moment (vt) are used in each iteration. As shown in the following Equation:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1)\, g_t \tag{20}$$

Since $m_t$ represents the weighted arithmetic average of the derivatives at time t, $\beta_1$ is the rate of exponential decay at the first moment, where its default value is 0.9, $m(t-1)$ represents the weighted arithmetic average of the derivatives at the previous time $t-1$, and $g_t$ represents the derivative of the cost function with respect to each parameter of the model.

As for the case of the second moment, it is calculated as follows:

$$v_t = \beta_2 v_{t-1}\, (1 - \beta_2)\, g_t^2 \tag{21}$$

Where $v_t$ represents the weighted decentralized variance of the derivatives at time $t$, $\beta_2$ is the rate of exponential decay for the second moment, with a default value of 0.999, $v(t-1)$ is the weighted decentralized variance of the derivatives at the previous time $t-1$, and $g_t^2$ represents the square derivative of the cost function with respect to each parameter of the model.

In the next step, the corrected bias for the exponential moving average of the derivatives $(m_t)$ and the exponential moving average of the squared derivatives $(v_t)$ is performed. Therefore, at the beginning of training, these averages are biased towards zero $m_t = v_t = 0$, so the corrected bias for the first moment is calculated as follows:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{22}$$

$\hat{m}_t$ represents the bias corrected for the first moment at time $t$, $\beta_1^t$ is the exponential decay rate for the first moment at time $t$. As for calculating the bias corrected for the second moment, as in the formula below:

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{23}$$

$\hat{v}_t$ represents the bias-corrected second moment at time $t$, $\beta_2^t$ is the exponential decay rate of the second moment at time $t$. Finally, these corrected biases are used to update a specific weight as in the formula below:

$$w = w - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}t + \epsilon}} \tag{24}$$

Where $\eta$ represents the learning rate, which is a value restricted to $0 < \eta \leq 1$, while for $\epsilon$, the value is $e^{-10}$ to prevent the denominator from being 0 [21]

## 4. The importance of features using the random forest algorithm

It is ranked as one of the supervised learning algorithms that mostly rely on the formation of a decision tree to sort the data. Bootstrap method with randomly selected features is also used. When the algorithm is applied to a data set containing n observations and p explanatory variables (predictors), it is divided into stages [25]:

1. Bootstrap: The dataset for training is selected randomly with replacement, resulting in each decision tree having a different portion of the data to learn from.
2. Tree formation: For each dataset sampled, a decision tree is created using a random subset of features. This helps avoid the creation of highly correlated trees.
3. Averaging OR Voting: Once all the decision trees are formed, each makes its own predictions. In regression, the expected values from each tree are calculated together. In the case of classification, the expected results from each tree are combined, and the class with the most votes is the final prediction.

### *4.1. Entropy*

Is a measure of how random or unpredictable the data is. The higher the entropy value, the harder it is to draw clear conclusions. When the probabilities are 0 or 1, the entropy $H(X)$ is zero, meaning that the system is free of randomness and completely predictable. When the probabilities are equal (0.5), the entropy reaches its highest value, reflecting that the data is completely random and unpredictable. In a decision tree, entropy is zero at the leaf nodes because the data is completely sorted and free of randomness. Branches with entropy greater than zero must be further split to reduce the randomness and better organize the data [27]. The entropy of a single property (feature) is calculated using the following formula:

$$Entropy = E\left(S\right) = -\sum_{i=1}^{c}\left(p_i\right)\log_2 p_i \tag{25}$$

Since $S$ represents the dataset associated with a specific category within this feature, $p_i$ Probability of category $i$, $c$ Represents the number of possible categories. This study deals with multiple features, so we need to calculate the entropy for each class in those features.

The equation below represents the calculation of total entropy when we have several categories or divisions based on a specific property.

$$E\left(T,X\right) = \Sigma_{C\epsilon X}\; p\left(c\right)E\left(c\right) \tag{26}$$

The equation $E(T,X)$ represents the total entropy when the data is split using feature $X$, where $X$ is the feature or variable we use to split the data, $c$ refers to the class or division within the feature, and $P(c)$ is the probability of class $c$, which is the ratio of the number of samples in class $c$ to the total number of samples. $E(c)$ is the entropy of class $c$, and the entropy is calculated for each class individually. Thus, the total entropy is calculated by combining the entropies of the different classes based on their probabilities.

### *4.2. Gini index*

The Gini index is a tool used to determine the level of purity or impurity of a particular feature in a dataset. It is primarily used in the CART algorithm to build a decision tree by splitting the data into parts. The Gini index is an important factor when selecting features to split the data, as features that produce a lower Gini index are preferred because they reflect a higher level of purity in the split [26]. The Gini index is calculated using the following formula:

$$Gini\ Index\ = 1 - \sum_{i=1}^{n} p_i^2 \tag{27}$$

Where $p_i$ represents the proportion of samples belonging to class $i$ in the group. The smaller the Gini index value, the purer and more effective the split, which improves the performance of the decision tree [27].

## 5. Model evaluation measures

Predictions are made for both positive and negative classes (0,1) using binary classification models. In order to evaluate the effectiveness of binary classification models, several classification metrics are calculated. These metrics, which can be calculated using a confusion matrix, include precision, recall, and F1 score, which are among the most commonly used metrics.

### *5.1. Confusion matrix*

A comparison of the predicted values with the actual values is carried out with the use of the confusion matrix, which contains both the accurate and wrong predictive values. The confusion matrix is organized so that the rows reflect the actual classes, while the columns represent the classes that the model predicted [22]. As in the Table 1

Table 1. The actual and expected categories in the confusion matrix

| | | Prediction | |
|---|---|---|---|
| | | 0 | 1 |
| Actual | 0 | True Negative (TN) | False positive (FP) |
| | 1 | False Negative (FN) | True Positive (TP) |

TN: Corresponds to the number of observations accurately labelled as negative.
FP: A Type I error is a term often used to describe the number of truly negative results that are incorrectly classified as positive.
FN: It means the number of observations that were labeled negative when they are in fact positive and is coined as a type II error.
TP: represents the number of samples accurately identified as positive.

*5.1.1. Model performance indicators analysis* **Accuracy**: Percentage of the correct results predicted by the model in relation to the total number of times that model actually made the predictions. As in the formula

$$ACC = \frac{TN + TP}{TN + TP + FP + FN} \tag{28}$$

**Precision**: is the ratio of true positives to total positives, which takes into account the total number of occurrences that were properly classified as positives in comparison to the total number of instances that were really positive. The formula is as follows

$$Precision = \frac{TP}{TP + FP} \tag{29}$$

**Sensitivity**: Another name for this measure is the True Positive Rate which reflects the model's capability to accurately identify positive cases. As in the following formula

$$Sen = \frac{TP}{TP + FN} \tag{30}$$

**Specificity**: Also known as TNR, it refers to the rate at which a model accurately predicts negative cases. As in the following formula

$$Spe = \frac{TN}{TN + FP} \tag{31}$$

**F1-score**: F-measure is less biased by the excess of either true positives or true negatives and is calculated as the harmonic mean of the value of precision and the value of recall. F-measure is very good if both recall and precision are good [23].

*5.1.2. Receiver operating characteristics (ROC) analysis* The ROC curve is produced using the true positive rate (TPR or sensitivity) and the false positive rate (FPR, 1 minus specificity) for different diagnostic test cutoff values from (0, 0) to (1, 1). As TPR and FPR vary from 0 to 1, the curve's points at the top left corner (0, 1) are more accurate, while those near the diagonal line are more random. The test's efficacy depends on how soon the curve reaches the optimum point. Faster means more accurate diagnostic results. The likelihood ratio (LR), which measures the test's ability to discriminate positive and negative findings, may be calculated from the tangent line's slope at any position on the ROC curve. The likelihood ratio is sensitivity / (1-specificity). The cutoff point does not add diagnostic value if the ratio is 1. A ratio larger than 1 enhances the chance of finding real positive instances, whereas a ratio less than 1 implies a lowered estimate of the positive condition's incidence. Diagnostic tests' accuracy is measured by the area under the ROC curve (AUC). The bigger the area, the better the test in distinguishing groups. An AUC of 1 implies flawless performance, whereas 0.5 suggests random guessing [24].

$$A \cup C = \int_0^1 ROC\ tdt \tag{32}$$

## 6. System model

The proposed system model is illustrated in Figure 2, This model takes raw data, extracts key variables from it, preprocesses them, and then uses them as a training dataset for the machine learning algorithm.

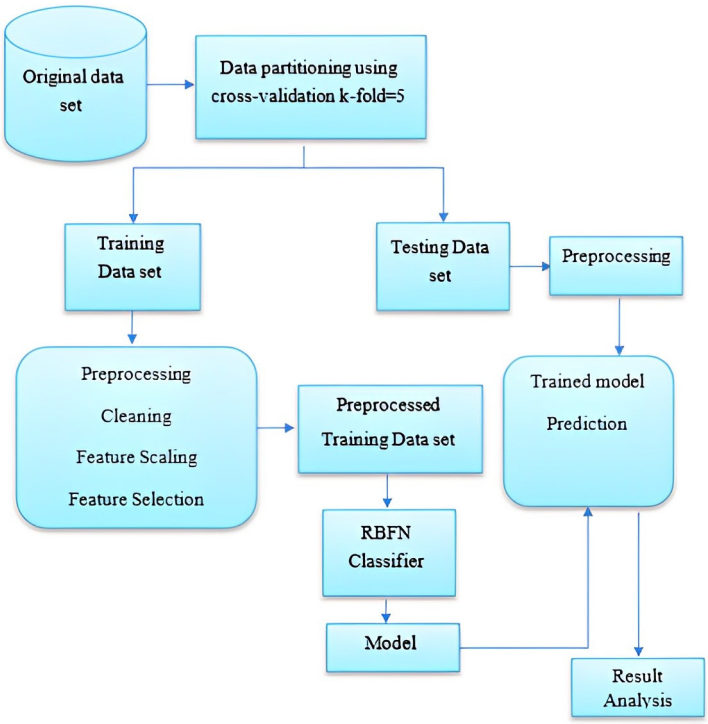



Figure 2. System model diagram.

In neural networks, a training dataset is fed into the neurons, which are activated or deactivated based on the inputs. During the training process, the connections between the neurons change continuously, after training is complete, the final model is obtained from the trained network, The performance of this network is tested using new data that was not used in the training process.

### *6.1. Dataset description*

In this study, the Thalassemia patient dataset consists of 280 observations and 13 variables. Among the 280 observations, 149 (53.21%) had moderate thalassemia, while 131 (46.78%) had major thalassemia. Table 2 summarizes the variables of the thalassemia patient dataset. We used cross-validation to split the dataset with k-fold=5 into training and testing, The characteristics from the thalassemia patient dataset that were utilised for training and assessing the RBF model are displayed in Table 2, We captured the demographic data and converted it into a numerical format using automatic label encoding, enabling the model to process the numerical data efficiently.

Table 2. Recoding of Thalassemia patient data variables

| No. | Variable | Description |
|---|---|---|
| 1 | Gender | (1= Males, 0= Females) |
| 2 | Age | The age of a person |
| 3 | Splenic enlargement | (0= Normal, 1= enlargement, 2= Splenectomy) |
| 4 | Heart disease | (0= NO, 1= Yes) |
| 5 | The growth is delayed | (0= NO, 1= Yes) |
| 6 | Osteoporosis | (0= NO, 1= Yes) |
| 7 | Blood transfusion | (0= NO, 1= Yes) |
| 8 | HB | Males; HB <13<br>Females; HB <11.5 |
| 9 | Mean cell volume | MCV<80 |
| 10 | HBA1 | Intermediate 50% to 70%, Major 0% |
| 11 | HBA2 | Intermediate 3% to 8% , Major 3% to 8% |
| 12 | HBF | Intermediate 20% to 40% , Major>80% |
| 13 | Diagnostic | (1=Thalassemia major, 0=Thalassemia Intermediate) |

### *6.2. Data cleaning*

Considering that the model that is being utilized is not capable of generalizing very effectively, this is an essential stage in which data that is identical or missing is checked and processed in its entirety. Consequently, it is possible for duplicates to be present in both the training set and the testing set, The missing values in the data are depicted in the accompanying Figure 3, which shows that there are a total of 18 missing values that are spread out over the variables (Splenic enlargement = 4, HB = 6, and HBF = 9).

```
Gender                  0
Age                     0
Splenic enlargement     4
Heart disease           0
The growth is delayed   0
Osteoporosis            0
Blood transfusion       0
HB                      6
MCV                     0
HBA1                    0
HBA2                    0
HBF                     9
Diagnostic              0
type: int64
```

Figure 3. Missing values in the data set.

Where duplicate or similar data was removed upon its appearance in the data under study. Also, missing values were addressed by completely deleting the patient's data and excluding them from the sample to avoid negatively impacting the training and testing process of the models.

### *6.3. Variables scaling*

The z-score is a method that is utilized to standardize data. This approach involves transforming the scale of data values by utilizing the mean and standard deviation of each single variable. Assuring that each variable has a consistent scale and reducing the impact of extreme values are the two goals that will be accomplished via the implementation of this standardization. Implementing a standardized z-score helps to enhance the uniformity of data interpretation while also increasing the stability of the outcomes of the study. Using the formula that is presented below, the z-score procedure is applied to each variable that is included in the data ($x_i$). As in the following formula:

$$z = \frac{x - \mu}{\sigma} \tag{33}$$

The standard score standardizes the data by transforming its distribution to a mean of 0 and a standard deviation of 1 [12]. By standardizing the data using z-score, the variables have the same importance and the same scale, and thus one variable does not overwhelm another. Negative values in the matrix of numbers below indicate that they are below the average, while positive values indicate that they are above the average.

```
array([[-1.03637545, -0.98421062,  0.35653702, ..., -0.62474508,
        -0.86789424,  0.66195784],
       [-1.03637545,  0.10312422,  0.35653702, ..., -1.01201364,
        -1.29929915,  1.06505161],
       [-1.03637545, -0.13850575,  0.35653702, ...,  1.42515764,
         1.90542306, -1.46421988],
       ...,
       [ 0.96490128,  1.79453396, -1.30730239, ...,  1.08156599,
        -0.74463569, -1.00719102],
       [-1.03637545, -0.50095069, -1.30730239, ...,  1.30286231,
        -0.55974787, -1.23144742],
       [-1.03637545,  2.51942385,  0.35653702, ...,  0.28955812,
        -0.80626497, -0.23222905]])
```

Figure 4. The results matrix for the data after using standardization.

## 7. Cross validation (k-fold)

The effectiveness of machine learning models is evaluated using this method. It facilitates understanding the predictive capability of the model when applied to new data, providing a more accurate assessment of the model's functions. The dataset is divided into K equal parts using this method, with one iteration using each part as test data and the remaining part as training data. The procedure is repeated until every component has been evaluated, and the average performance of the model across all iterations is often used to determine the final result [13]. In this study, the k-fold cross-validation method was used, where $k = 5$, thus dividing the dataset consisting of 280 observations into 5 subsets, with each subset containing 224 observations in the training set and 56 observations in the test set, resulting in the training and testing process being conducted 5 times. Figure 5 below illustrates the division of the dataset in the case of using cross-validation.

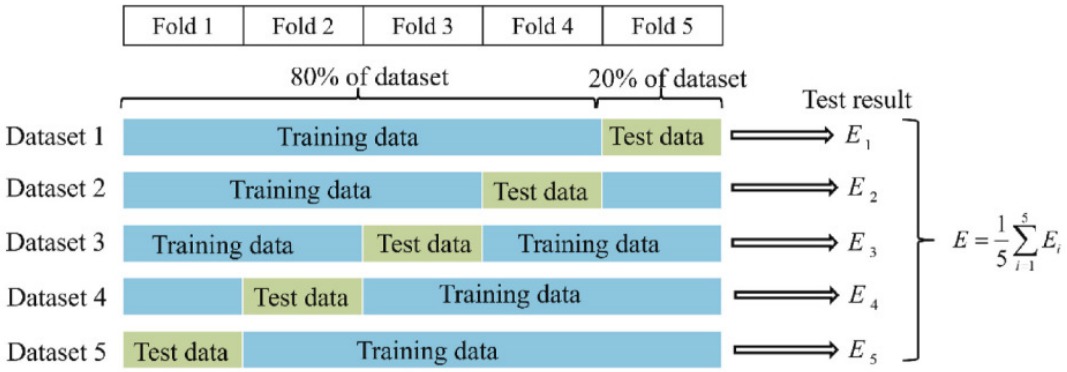



Figure 5. Illustrates the division of the dataset in the case of cross-validation based on [14].

## 8. Result and discussion

The current RBFClassifier is implemented in Python3 while the library used is the Scikit-learn library from Python. This means that when training the RBF network on the patient data under study, we can observe the behavior of the training process at different k-fold splits by adding the cross-validation result of training the radial basis function network. Table 3 consists of six columns: the first column represents the number of k-fold splits; the second and third columns show the training and validation accuracy respectively; the fourth and fifth columns represent the errors observed during the testing and training phases; and the last column indicates the learning rate. As follows:

Table 3. Consists of the training process information of RBFN with specific learning rate of 0.1 using Adam optimization method

| K-Fold | Training accuracy | Val accuracy | Training loss | Val loss | Learning rate |
|---|---|---|---|---|---|
| 1 | 0.9688 | 0.9464 | 0.0312 | 0.0536 | 0.1 |
| 2 | 0.9509 | 0.9821 | 0.0491 | 0.0179 | 0.1 |
| 3 | 0.9732 | 0.9464 | 0.0268 | 0.0536 | 0.1 |
| 4 | 0.9598 | 0.9643 | 0.0402 | 0.0357 | 0.1 |
| 5 | 0.9643 | 0.9643 | 0.0357 | 0.0357 | 0.1 |
| Average | ± 0.9643<br>0.0077 | ± 0.9607<br>0.0134 | 0.0366 | 0.0393 | |

In the table, the training accuracy is (0.9509-0.9732) and the results show that the model has a good ability to learn the training data. Regarding the validation accuracy, the accuracy ranges from 0.9464 to 0.9821 to justify the correct evaluation of the model. Regarding the training loss, it ranges from (0.0268-0.0491). On the other hand, the validation accuracy of the model to generalize unseen data ranges from (0.0179-0.0536). The standard deviation of the training accuracy is ± 0.0077, while the validation accuracy is ± 0.0134, as there is no significant variation between the sections, which indicates the stability of the model. We conclude from this table that the model does not suffer from overfitting.

Figure 6 represents the training and validation loss for the average of all splits in the network for RBF where the y-axis is the cost function and the x-axis is the epoch. From the graph below, it can be seen that the loss curves for training as well as testing start decreasing in a similar manner where the percentage reaches 10% in training and in testing it is 11% at 60 training epochs, so it can be concluded that the model trains well without suffering from generalization problem.

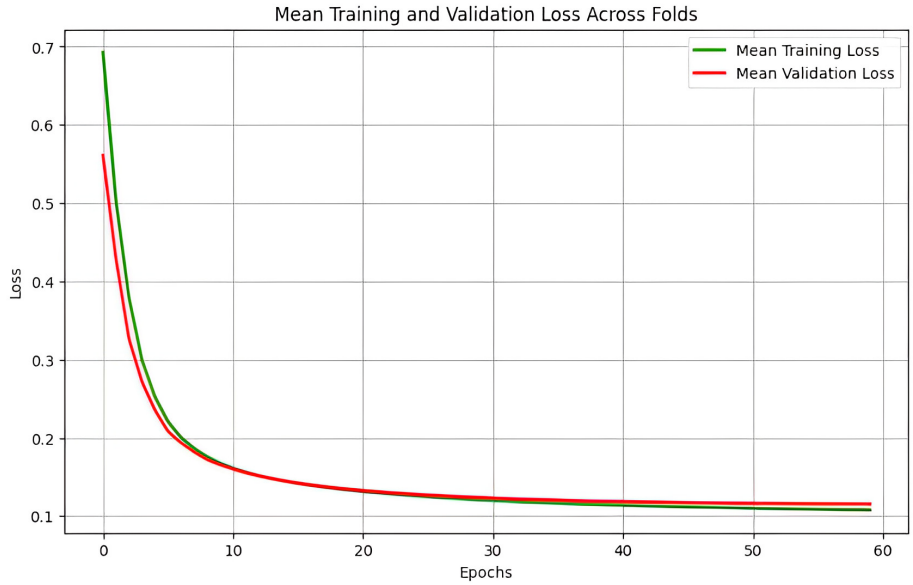



Figure 6. The training and testing loss curves for the average of all splits of the radial basis function network.

For the accuracy of the RBF network, the Figure 7 represents the training and testing accuracy curves, It should be remembered that this accuracy is low at the beginning of the learning process for both training and testing, the accuracy percentage gradually increases and reaches 96.34% in training and 96.07% in testing at 60 training cycles, so, this means that the data recognition by the RBF network is enhanced with the training process.

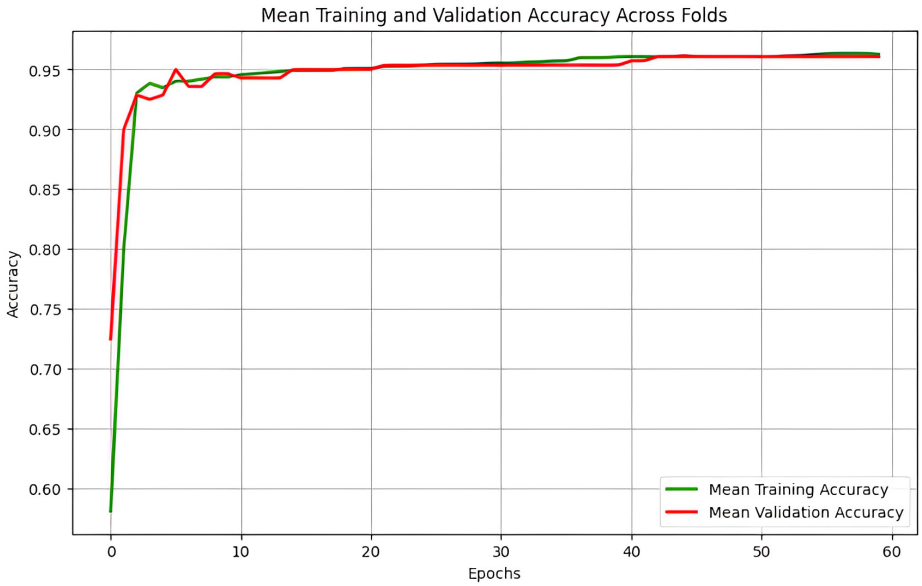
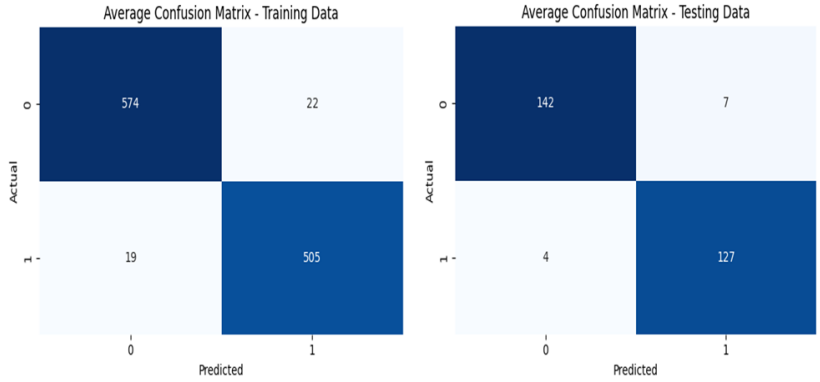



Figure 7. The accuracy curves for training and testing of the radial basis function network.

By using the data of thalassemia patients, we can see the performance of the RBF network by comparing the classification results with the real classification, which is by applying the rabbit matrix (confusion matrix) to the training data and testing thalassemia patients. The application results represent two rows and two columns clearly

with a binary classification problem. It contains the Figure 8 representing the rank matrix of the training data, where the activity of thalassemia intermedia is among them and is expressed in the matrix (TN). The nutritional elements in (the first row of the second column) contain 22 values, indicating the number of observations that were incorrectly classified as patients with thalassemia major while they are infected with thalassemia intermedia and are expressed in the matrix (FP). The antiviral activity in (the second row of the first column) contains 19 values and indicates the observations that were illegally classified as the two elements infected with thalassemia intermedia while they are infected with thalassemia major and are expressed in the matrix (FN). The number of vital elements in (the second row of the second column) was examined on 505 values and indicates the number of observations that were correctly classified into two thalassemia major contacts and expressed in the matrix (TP). Figure 8b is the confusion matrix relating to the test data. The value 142 refers to the correct number of cases classified as having thalassemia intermedia 7 cases reported as having thalassemia major when in fact they have intermedia. 4 cases were reported to have thalassemia intermedia when in actual sense they have thalassemia major. The value 127 means the number of observations that fall in the thalassemia major class and was correctly classified. It is shown in the following diagram:




(a) (b)

Figure 8. A confusion matrix of the radial basis network in the training data set, (a) and the validation data set, (b) is presented.

Since it defines a measure of accuracy, it is important to remember that when we have a dataset with imbalanced classes, this measure can be somewhat misleading. In certain instances, it is also required to make use of additional metrics, such as precision, recall, and F1 score, in order to acquire extra information on the performance of our model. Also, in the figure, we have the RBF measures on the classification points of the training and test data on binary cases, especially each thalassemia case. From the RBF network classification report, it is found that the accuracy of the model for classifying thalassemia major and intermediate patients is 97% for class 0 and 95% for class 1. Also, the recall of class 0 is at a very high level of 95% and the recall of class 1 is 97% which states that most of the true cases are identified in all classes. Regarding the last F1 score parameter which is combined precision and recall, it performed very well in both categories with 96% in category 0 and 95% in category 1. Exist refers to the number of actual results in each category of prediction, and the RBF model was found to be very accurate and stable by providing this prediction support system for thalassemia diagnosis. The following table shows the radial core network classification data report as follows:

Table 4. The radial basis function network classification report across the average of all divisions

| | Classification Report (Average Across All Folds) | | | |
|---|---|---|---|---|
| | Precision | Recall | F1-score | Support |
| 0 | 0.97 | 0.95 | 0.96 | 149 |
| 1 | 0.95 | 0.97 | 0.96 | 131 |
| accuracy | | | 0.96 | 280 |
| macro avg | 0.96 | 0.96 | 0.96 | 280 |
| Weighted avg | 0.96 | 0.96 | 0.96 | 280 |

When using the ROC (Receiver Operating Characteristic) curve, ”AUC” or ”Area Under the Curve” is used to evaluate the model’s performance at different thresholds. The ROC curve measures the relationship between the true positive rate and the false positive rate. Points on the AUC curve represents the performance of the classification model, where the area under the curve indicates the model’s accuracy in distinguishing between the two classes. The higher the AUC value for a given model, the better its performance in correctly classifying the classes. In this context, an AUC value of 0.995 indicates excellent performance, suggesting that the classification model is nearly ideal in distinguishing between different classes.

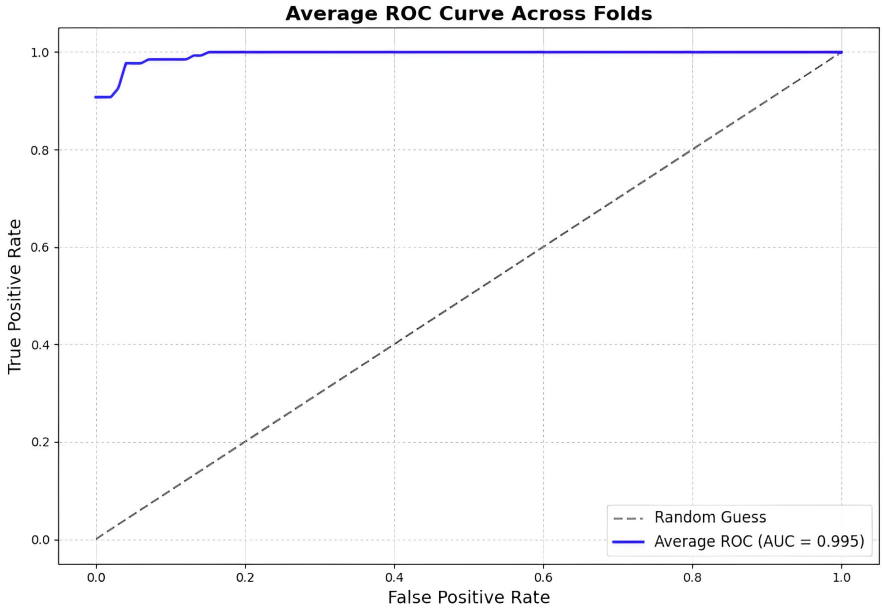



Figure 9. The ROC curve for the radial basis function neural network model.

The importance of traits in determining the type of thalassemia depends on their impact on the genetic index, where the variable with the lowest genetic index is considered the most important. Figure 10 illustrates the importance of different traits in distinguishing between major and intermediate thalassemia using a random forest model. The horizontal axis displays the relative importance of each trait, while the vertical axis represents the traits that were analyzed.

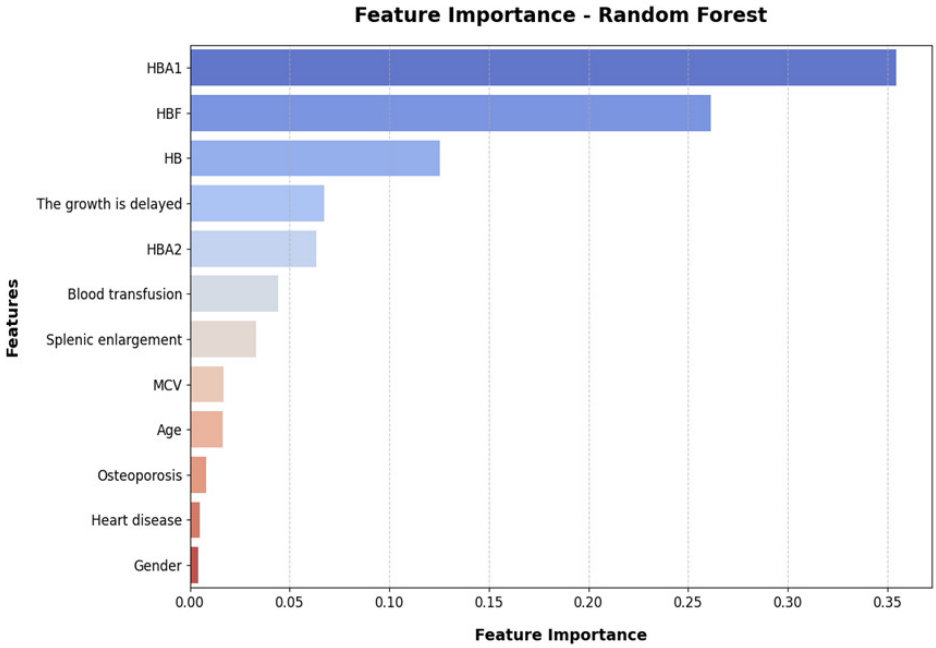



Figure 10. Importance analysis of features in diagnosis using random forest.

The most important variables in this model are HBA1 (adult hemoglobin) and HBF (fetal hemoglobin), with HBA1 showing the highest relative importance (0.3461), indicating that the level of HBA1 is the most prominent indicator in distinguishing between major and intermediate thalassemia. In major thalassemia, the level of HBA1 is very low, while in intermediate thalassemia it decreases to a lesser extent. HBF (0.2667) follows as the second most important variable, with its levels rising significantly in major thalassemia (over 80%) while being lower in intermediate thalassemia. HB (total hemoglobin) ranks third (0.1410) and is an important indicator of the severity of anemia in major thalassemia, where it is very low compared to intermediate thalassemia. Growth retardation (0.0755) comes in fourth place, being a common manifestation in major thalassemia, but its impact is less than that of HBA1 and HBF. Next is HBA2 (0.0677), which is important in distinguishing between types of thalassemia but is less significant than the previous variables. Variables such as blood transfusion (0.0365) and splenomegaly (0.0255) indicate that these factors play a role in diagnosing major thalassemia, but they are less important in distinguishing between types. While MCV (0.0180) shows a minimal effect compared to the other features, it is low in major thalassemia. Age (0.0127) and osteoporosis (0.0070) also have a weak effect on diagnosis, while heart diseases (0.0010) and gender (0.0024) show less impact in determining the type of thalassemia.

## 9. Conclusion

We conclude that the basic functional neural network based on the radial basis function performs exceptionally well when applied to data from thalassemia patients, with highly consistent results across various departments. The model shows perfect alignment between training accuracy and validation accuracy, indicating no issues of underfitting or overfitting. Experimental results demonstrate that using a radial basis function network with the Adam algorithm and a learning rate of 0.1 enhances performance and achieves excellent accuracy. Additionally, the experiment confirms that the optimal network configuration includes one hidden layer containing 16 neurons, which enhances the model's efficiency in classifying thalassemia in a binary manner. Furthermore, a random forest model was used to identify the most significant features influencing the differentiation between types of thalassemia. The results showed that the most important features are HBA1 (adult hemoglobin) and HBF (fetal hemoglobin),

which represent the main indicators for determining the type of thalassemia due to their significant impact on classification. This is followed by the HB (total hemoglobin) feature as a third important feature, and then growth delay and HBA2 with varying degrees of importance. These analyses helped identify the fundamental factors associated with the genetic and clinical differences between major thalassemia and intermediate thalassemia, contributing to enhancing the understanding of the precise classification of the disease and improving diagnostic and treatment strategies.

## 10. Recommendations

1. Increase the dataset size by collaborating with additional medical centers to improve the generalizability of the results.
2. Expanding the research study by using additional methods that include comparison with neural network models, including Convolutional Neural Networks (CNN).
3. Expanding the use of the Adam optimization algorithm to improve the results of methods in research and statistical studies due to its efficiency and flexibility compared to other methods.

REFERENCES


1. A.A. Ayrancı and B. Erkmen, *IoT-based fire detection: A comparative study of machine learning techniques*, Niğde Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi, vol. 13, no. 4, pp. 1–1, 2024.
2. M. A. Samani, F. Khazaee, and S. Ebadollahi, *Multi-Layer Perceptron and Radial Basis Function for Thyroid Disease Diagnosis*, Preprint, 05 Mar. 2024, available at Research Square, doi: https://doi.org/10.21203/rs.3.rs-3994161/v1.
3. M. Chen, T. Zhu, and J. Guan, *The evaluation on balance of economic benefit and water pollution of enterprise based on Kmeans-RBF neural network*, Proc. 4th Int. Conf. Informatization Economic Dev. and Manag. (IEDM 2024), Kuala Lumpur, Malaysia, 2024.
4. A. Veeramani and P. I. Khan, *Flight Delay Prediction and Error Analysis Using Machine Learning*, vol. 14, pp. 153–161, 2024.
5. Z. Kucukakcali, and I. B. Cicek, *Modeling the Factors Affecting Gleason Score with Artificial Neural Networks and Indirectly Determining Prostate Cancer Risk Factors*, PriMera Sci. Med. Public Health, vol. 5, no. 1, pp. 3–11, 2024.
6. A.Y. Abushawish, and A.B. Nassif, *Prediction of early-stage diabetes using machine learning*, 2023 Advances in Science and Engineering Technology International Conferences (ASET), 1–4, 2023.
7. Z. Jafari, A. M. Yousefi, and S. Rajebi, *Investigation on different pattern classification methods and proposing the optimum method with implementation on blood transfusion dataset*, International Journal on Technical and Physical Problems of Engineering, vol. 12, no. 2, pp. 66–70, 2020.
8. D. A. Tyas, S. Hartati, A. Harjoko, and T. Ratnaningsih, *Erythrocyte Classification using Multi-Layer Perceptron, Naïve Bayes Classifier, RBF Network, and SVM*, International Journal of Engineering and Advanced Technology, vol. 9, no. 2, pp. 2024–2028, 2019.
9. I. Ahmad, S. N. H. S. Abdullah, and R. Z. A. R. Sabudin, *Morphological features analysis for erythrocyte classification in IDA and Thalassemia*, International Journal of Advanced Computer Science and Applications, vol. 9, no. 12, pp. 274–280, 2018.
10. F. Yousefian, T. Banirostam, and A. AzarKeivan, *Prediction of Mellitus Diabetes in Patients with Beta-thalassemia using Radial Basis Network, and k-Nearest Neighbor based on Zafar Thalassemia Datasets*, Diabetes, vol. 19, pp. 1–10, 2017.
11. G. L. Masala, B. Golosio, R. Cutzu, and R. Pola, *A two-layered classifier based on the radial basis function for the screening of thalassaemia*, Computers in Biology and Medicine, vol. 43, no. 11, p. 1724–1731, 2013.
12. M. R. Firmansyah, and Y. P. Astuti, *Stroke classification comparison with KNN through standardization and normalization techniques*, Advance Sustainable Science, Engineering and Technology, vol. 6, no. 1, p. 02401012, 2024.
13. I. K. Nti, O. Nyarko-Boateng, and J. Aning, *Performance of Machine Learning Algorithms with Different K Values in K-fold CrossValidation*, Int. J. Inf. Technol. Comput. Sci., vol. 13, no. 6, pp. 61–71, 2021.
14. G. Wang, C. Li, F. Tang, Y. Wang, S. Wu, H. Zhi, F. Zhang, M. Wang, and J. Zhang, *A fully-automatic semi-supervised deep learning model for difficult airway assessment*, Heliyon, vol. 9, no. 5, p. e15629, 2023.
15. F. A. Ruslan, Z. M. Zain, and R. Adnan, *Modelling flood prediction using Radial Basis Function Neural Network (RBFNN) and inverse model: a comparative study*, in 2013 IEEE International Conference on Control System, Computing and Engineering, Penang, Malaysia, Nov. 2013, pp. 577–581.
16. Z. Gvishiani, and J. Dawidowicz, *Comparison of MLP and RBF neural networks in the task of classifying the diameters of water pipes*, Rocznik Ochrona Środowiska, vol. 24, pp. 505–519, 2022.
17. J. Liu, *Radial Basis Function (RBF) Neural Network Control for Mechanical Systems*, Springer: Berlin/Heidelberg, Germany, 2013.
18. D. Indarto, *Stimulasi sensorimotor sebagai layanan holistik integratif unggulan di Paud Terpadu Zaki's Club Gemolong Kabupaten Sragen*, Pengaruh pengguna pasta labu kuning (Cucurbita moschata) untuk substitusi tepung terigu dengan penambahan tepung angkak dalam pembuatan mie kering, vol. 15, pp. 165–175, 2016.
19. Y. Wu, H. Wang, B. Zhang, and K. L. Du, *Using radial basis function networks for function approximation and classification*, International Scholarly Research Notices, vol. 2012, no. 1, p. 324194, 2012.

20. O. Hospodarskyy, V. Martsenyuk, N. Kukharska, A. Hospodarskyy, and S. Sverstiuk, *Understanding the Adam optimization algorithm in machine learning*, CITI'2024: 2nd International Workshop on Computer Information Technologies in Industry 4.0, June 12–14, 2024, pp. 235–248.
21. Y. Ywema, *Learning from demonstration for isolating forest fires using convolutional neural networks*, Bachelor's Thesis, Artificial Intelligence, University of Groningen, Groningen, Netherlands, 2020.
22. A. Bakumenko, and A. Elragal, *Detecting anomalies in financial data using machine learning algorithms*, Systems, vol. 10, no. 5, p. 130, 2022.
23. T. J. Prins, *UCLA electronic theses and dissertations title*, 2019. Available from: https://escholarship.org/uc/item/0th2s0ss.
24. W. Zhu, N. Zeng, and N. Wang, *Sensitivity, specificity, accuracy, associated confidence interval and ROC analysis with practical SAS implementations*, NESUG Proc., vol. 19, p. 67, 2010.
25. V. F. Rodriguez-Galiano, B. Ghimire, J. Rogan, M. Chica-Olmo, and J. P. Rigol-Sanchez, *An assessment of the effectiveness of a random forest classifier for land-cover classification*, ISPRS Journal of Photogrammetry and Remote Sensing, vol. 67, no. 1, pp. 93–104, 2012.
26. K. Açıcı, Ç. B. Erdaş, T. Aşuroğlu, M. K. Toprak, H. Erdem, and H. Oğul, *A random forest method to detect Parkinson's disease via gait analysis*, Communications in Computer and Information Science, vol. 744, pp. 609–619, August 2017.
27. G. N. Ahmad, S. Ullah, A. Algethami, H. Fatima, and S. M. H. Akhter, *Comparative study of optimum medical diagnosis of human heart disease using machine learning technique with and without sequential feature selection*, IEEE Access, vol. 10, pp. 23808–23828, 2022.