

Logarithmic Spiral-based Construction of RBF Classifiers

Mohamed Wajih Guerfala

Laboratory of research in Automatic Control, ENIT
University of Tunis El Manar,
National Engineering School of Tunis
BP 37, Le Belvédère 1002 Tunis,
Tunisia

Amel Sifaoui

Laboratory of research in Automatic Control, ENIT
University of Tunis El Manar,
National Engineering School of Tunis
BP 37, Le Belvédère 1002 Tunis,
Tunisia

Afef Abdelkrim

Laboratory of research in Automatic Control, ENIT
University of Carthage, National Engineering School of Carthage (ENICarthage),
BP 37, Le Belvédère 1002 Tunis,
Tunisia

Abstract—Clustering process is defined as grouping similar objects together into homogeneous groups or clusters. Objects that belong to one cluster should be very similar to each other, but objects in different clusters will be dissimilar. It aims to simplify the representation of the initial data. The automatic classification recovers all the methods allowing the automatic construction of such groups. This paper describes the design of radial basis function (RBF) neural classifiers using a new algorithm for characterizing the hidden layer structure. This algorithm, called k-means Mahalanobis distance, groups the training data class by class in order to calculate the optimal number of clusters of the hidden layer, using two validity indexes. To initialize the initial clusters of k-means algorithm, the method of logarithmic spiral golden angle has been used. Two real data sets (Iris and Wine) are considered to improve the efficiency of the proposed approach and the obtained results are compared with basic literature classifier

Keywords—Radial Basis Function neural network; classification; k-means; validity index of Davis Bouldin; Mean Squared Error; Mahalanobis distance; Logarithmic spiral; golden angle; golden ratio

I. INTRODUCTION

Clustering is one of the most useful tasks in data mining process for discovering groups and identifying interesting distributions and patterns in the underlying data. The problem with Clustering is about partitioning a given data set into groups (clusters) such that the data points in a cluster are more similar to each other than points in different clusters [1].

In recent years, a number of clustering algorithms has been proposed and is available in literature. The radial basis function (RBF) neural network is one of the most used in data classification.

RBF neural networks consist of three layers: an input layer, a hidden layer and an output layer. The input layer corresponds to the input vector feature space and the output layer corresponds to the pattern classes [2]. So the whole architecture is fixed only by determining the hidden layer and the weights between the middle and the output layers [3].

Its training procedure is usually split into two successive steps: training in the hidden layer followed by training in the

output layer [4]. First, the centers of the hidden layer (HL) neurons are selected by clustering algorithms such as k-means [5], [6], support vector machine (SVM) [7] or hierarchical clustering [8]. Second, the weights connecting the hidden layer with the output layer are determined by Singular Value Decomposition (SVD) or by Least Mean Squared (LMS) algorithms.

One of the used techniques to find the optimal number of this HL is the logarithmic spiral which has seen a significant amount of research on nature-inspired optimization techniques such as neuro-computing in the past 25 years, evolutionary and genetic algorithms, particle swarm optimization. Most recently, a new multipoint meta-heuristics research method has emerged for 2-dimensional continuous optimization problems based on the analogy of spiral phenomena in nature, called 2-dimensional spiral optimization first proposed by Tamura and Yasuda in 2010 [9].

Focused spiral phenomena are approximated to logarithmic spirals, which frequently appear in nature, such as whirling currents, nautilus shells and arms of spiral galaxies. A Two-dimensional spiral optimization uses the feature of Logarithmic Spirals [LS] [9].

In this paper, a new learning algorithm is proposed for the construction of the radial basis function networks solving classification problems. It determines the proper number of hidden neurons automatically and calculates the centers values of radial basis functions. After the selection of the hidden neurons, the widths of nodes are determined by the P-nearest neighbors heuristic, and the weights between the hidden layer and the output layer are calculated by the pseudo-inverse matrix.

The aim of this approach consists in transforming the problem of determining the number of hidden layer neurons to a clustering problem. In order to determine the number of clusters in the data of each class, the k-means algorithm is combined with two different validity indexes (the validity index of Davis Bouldin for the first classifier and Mean Square Error for the second classifier).

In k-means algorithm, the used distance corresponds to the Mahalanobis distance. A solution is also given to overcome

the problem of setting the start values for the initial centers needed to start this algorithm using the proposed method "The logarithmic spiral golden angle". Two different real databases are used in order to evaluate the two proposed classifier performances.

Next section presents the problem of the construction of the hidden layer for RBF neural networks. Section 3 describes in detail the logarithmic spiral golden angle. Section 4 is devoted to the elaborated solution to overcome the problem of the k-means algorithm. Section 5 describes the construction of the two proposed RBF classifiers. Experiments and discussions are presented in Section 6, followed by concluding remarks in Section 7.

II. PROBLEM STATEMENT OF USE

The construction of the hidden layer of RBF neural networks is by clustering algorithms such as k-means. K-means clustering algorithm is one of the best-known algorithms used in clustering.

However, it still has some problems one of which is in its initialization step which is generally done randomly by users. Another disadvantage of k-means is that it converges to local optimum, depending on its random initialization.

The k-means algorithm classifies objects to a pre-defined number of clusters, which is given by the user (assume k clusters). The idea is to choose random cluster centers, one for each cluster. These centers are preferred to be as far as possible from each other. The Starting points affect the clustering process and results [10], "Fig. 1".

Each boot (initialization) is a different solution (local optimum) which can in some cases be far from the optimal solution (global optimum) [11]. A simple solution to this problem is to run the algorithm several times with different initialization and retain the best combination found.

The use of this solution is limited because of its cost and because of the possibility of finding better results in a single execution [12].

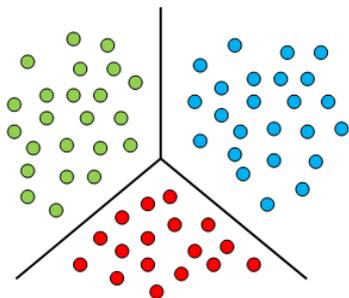


Fig. 1. Clustering on a set of 2D points data, 3 clusters

The main idea of this work is to improve the performance of the k-Means clustering algorithm by fixing its weaknesses. Randomness is one of the techniques wildly used in initializing K-means algorithm that is why it is considered as the main point of weakness that should be solved.

However, because of the sensitivity of k-means to its initial points, two solutions have been proposed to this problem. The first one is to initialize the centers of k-means algorithm using the circle method [13]. The obtained results of this algorithm are the initial centers positions C_k represented by the "Fig. 2".

The second solution is to initialize the centers of k-means algorithm using the logarithmic spiral golden angle in order to improve the clustering performance. In this paper, the second solution will be explained.

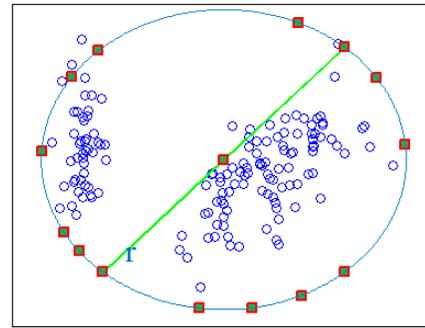


Fig. 2. Tracing of initialization of the k centers maximum on the outline of the circle of radius r and spaced by the angle θ

III. THE LOGARITHMIC SPIRAL GOLDEN ANGLE

The logarithmic spiral golden angle is a specific case of the logarithmic spiral. It represents a plane curve centered in a starting point and parameterized by the radius r , the angle θ and the Golden Ratio φ .

A. The logarithmic spiral

A Logarithmic Spiral is a plane curve for which the angle between the radius vector and the tangent to the curve is a constant [14]. Such spirals can be approximated mathematically defined by the following equation on the 2-dimensional polar coordinate system (r, θ) as [9]:

$$r = a e^{b\theta} \tag{1}$$

Where a and b are positive real with $a > 0$ and $b \neq 0$.

Equation (1) can be transformed into Cartesian coordinates as follows:

$$\begin{cases} x(\theta) = r(\theta) \cos(\theta) = a e^{b\theta} \cos(\theta) \\ y(\theta) = r(\theta) \sin(\theta) = a e^{b\theta} \sin(\theta) \end{cases} \tag{2}$$

In this work, the factor b of the logarithmic spiral has been set to zero ($b = 0$), it goes back to simplify the polar radius as follows: $r = a e^{b\theta} = a e^{0 \times \theta} \Rightarrow r = a$

The following equation of the logarithmic spiral golden angle is obtained:

$$\begin{cases} x(\theta) = r \cos(\theta) = a \cos(\theta) \\ y(\theta) = r \sin(\theta) = a \sin(\theta) \end{cases} \quad (3)$$

B. The Golden ratio

The irrational number, golden ratio is also known as golden section by the ancient Greeks, golden proportion, divine proportion or golden number [15].

The golden ratio φ , has many properties which people are eager to know. It is a number that is equal to the reciprocal of its own with the addition of 1: $\varphi = \frac{1}{\varphi} + 1$.

Likewise, the ratio of any two consecutive Fibonacci numbers converges to give approximates of 1.618, or its inverse, 0.618. This shows the relationship between Fibonacci numbers and golden ratio [16].

If the possibility of dividing a line in such a way that the ratio of the whole length to the length of the longer segment happens to be equal to the ratio of the length of the longer segment to the length of the shorter segment, then it could be said that the ratio is a golden ratio [15], "Fig. 3".

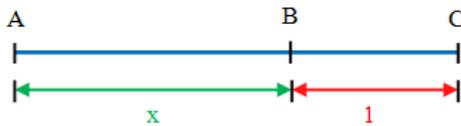


Fig. 3. Dividing of a whole length AC into two segments AB and BC

This gives mean ratio if $\frac{AB}{BC} = \frac{AC}{AB}$. If the value of AB is set to be x , and use 1 to represent the length of BC , then $\frac{x}{1} = \frac{1+x}{x}$ is obtained. Then the irrational number is the only

positive solution of the equation $x^2 - x - 1 = 0$, so $x = \frac{1+\sqrt{5}}{2}$

Its value is: $\varphi = \frac{1+\sqrt{5}}{2} \approx 1,6180339887$. Where the Greek letter phi (φ) represents the golden ratio.

C. The Golden angle

In geometry, the golden angle is created by dividing the circumference of a circle c in two sections, a longer arc of length a and a smaller arc of length b such that: $c = a + b$

and $\varphi = \frac{a+b}{a} = \frac{c}{a} = \frac{a}{b}$, "Fig. 4".

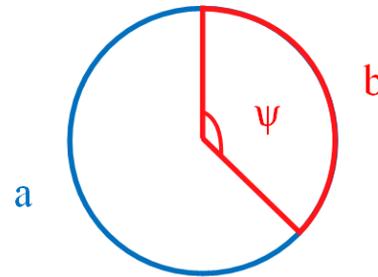


Fig. 4. Golden angle measurement

The angle formed by the arc b of circle c is called the golden angle ψ . It derives from the golden ratio φ .

$$\psi = 2\pi - \left(\frac{2\pi}{\varphi} \right) = 2\pi \frac{(\varphi-1)}{\varphi} \approx 2,391 \text{ rad} \approx 137,5^\circ$$

IV. PROPOSED INITIALIZATION OF THE K-MEANS ALGORITHM WITH LOGARITHMIC SPIRAL GOLDEN ANGLE

The k-means algorithm aims to minimize the distance between the object and the center of its group. In this section, the k-means algorithm based on the Mahalanobis distance and its proposition for initialization of the centers are presented.

A. The k-means algorithm Mahalanobis distance specifications

There are different types of distances such as: Minkowski distance, the average, the family of metrics, Euclidean Weighted and the Euclidean distance which is the most used, e.g. applied in the RBF Networks. [17]

Moreover, the Mahalanobis distance is a distance measure and its utility is a way to determine the similarity between two multidimensional random variables. It differs from Euclidean distance, because the Mahalanobis distance takes into account the correlation between random variables, [17]. The Mahalanobis distance is defined by:

$$d(x, y) = \sqrt{(x - y) \times Cov(X)^{-1} \times (x - y)^T} \quad (4)$$

With $Cov(X)$ is the covariance matrix.

If the elements x and y are independent, the covariance matrix is the identity and the Mahalanobis distance is equal to the Euclidean distance. The proposed algorithm based on the combination of Mahalanobis distance with k-means is described by the following steps:

Algorithm: Function *Kmeans_distance_Mahalanobis* (KDM)

Begin
Input: - The database $X = \{x_1, x_2, \dots, x_N\} \in R^d$.
 - The position of each center $C = \{c_1, c_2, \dots, c_k\} \in R^d$.
Output: - The new position of each center $C^* = \{c_1^*, \dots, c_k^*\} \in R^d$
Step 1:
 - Determine the size N of the data base of X .
 - Determine the number k of centers to be used in the observation space C .
 - Initialize the vector of the new positions of the centers C^* to zero.
Step 2: - Determine the covariance matrix $Cov(X)$ with the following equation:

$$Cov(X) = \frac{1}{N-1} \sum_{i=1}^N (X_{ij} - \bar{X}_j)(X_{ij} - \bar{X}_j)^T$$
 with $X_{ij} \in X$; $i = 1, \dots, n$ and $j = 1, \dots, p$.
 Where $\bar{X}_j = \sum_{i=1}^n X_{ij}$ with \bar{X}_j : arithmetic averages.
While: The new centers do not have, a significant displacement **Do:**
Step 3: - Assign each observation (dot) group nearest center $c_j : x_i \in c_j$ according to the Mahalanobis distance formula:

$$d(i, j) = \sqrt{(x_i - c_j) \times Cov(X)^{-1} \times (x_i - c_j)^T}$$
 with $i = 1, \dots, N$ and $j = 1, \dots, k$.
Step 4: - Recalculate the position of each new center:

$$c_j^* = \frac{1}{N_j} \sum_{x_i \in c_j} x_i$$
 with N_j = the set of points belonging to the center c_j and $j = 1, \dots, k$.
End While
End

To increase the performance of the k-means algorithm Mahalanobis distance, a solution is proposed to initialize the k-means algorithm using the logarithmic spiral golden angle parameterized by the radius r , the angle θ and the Golden ratio ϕ and the Golden angle ψ [18]. This solution is divided into several steps:

The first step is to calculate the maximum distance between two individual points (a, b) belonging to the database, then to define the middle ground G between these two individual points and determine the radius $R = \overline{Gb}$ "Fig. 5".

The second step is to calculate the golden number ϕ by applying the following formula: $\phi = \frac{1 + \sqrt{5}}{2} \approx 1,6180339887$

The third is to initialize the values of the logarithmic spiral golden angle on the polar coordinate system (r, θ) : the radius $r = a = a_0 = 0$ and the angle $\theta = \theta_0 = 0$. The angle θ increases by the factor $d\theta = \psi = 2\pi \frac{(\phi-1)}{\phi}$ and the radius r increases by the factor $da = \frac{R}{k_{max}} = \frac{\overline{Gb}}{k_{max}}$.

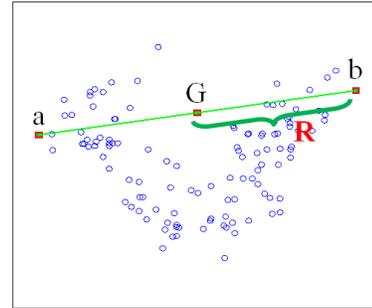


Fig. 5. Tracing of the two most distant individuals (a, b) and their medium G

To determine k_{max} , the suggestion of Bezdek was adopted [19] as follows : $k_{max} = \sqrt{N}$, (N is the size of the database).

The fourth step is to determine the positions of the centers of the logarithmic spiral golden angle with center G and radius $r = a$ "Fig. 6". The calculation of the center position C_k is performed by applying the following formula:

$$\begin{cases} C_{kx} = G_x + a \times \cos(\theta) \\ C_{ky} = G_y + a \times \sin(\theta) \end{cases} \quad (5)$$

With $\theta = \theta + d\theta$; $a = a + da$ and $k = 1, \dots, k_{max}$.

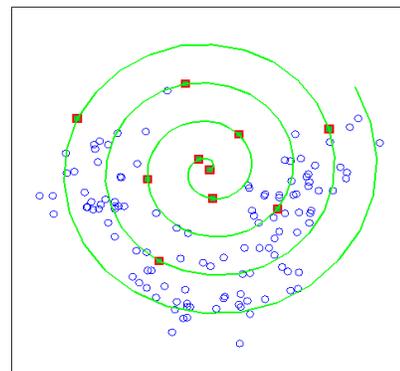


Fig. 6. Tracing of the initialization of the k centers maximum on the outline of the logarithmic spiral golden angle

Thereafter, the positions of these k_{max} centers in the variable $C_k = \{c_1, \dots, c_{k_{max}}\}$ will be saved.

The basic principle of the adopted strategy is summarized in the following algorithm:

Algorithm: *Init_Centres_Kmeans_Logarithmic_Spiral*

Begin
Input: - The database $X = \{x_1, x_2, \dots, x_N\} \in R^d$.
 - The maximum number of centroids k_{\max} .
Output: - The position of each center $C = \{c_1, \dots, c_k, \dots, c_{k_{\max}}\} \in R^d$
Step 1: - Calculate the maximum distance D between two points belonging to the base X .
 - Calculate the center G of D and the radius $R = \overline{Gb}$.
Step 2: - Calculate the golden ratio φ by applying the following formula: $\varphi = \frac{1 + \sqrt{5}}{2}$
Step 3: - Initialize the values of the logarithmic spiral golden angle on the polar coordinate system (r, θ) : the radius $r = a = a_0 = 0$ and the angle $\theta = \theta_0 = 0$.
 - Fix the increment of the angle θ by the factor $d\theta = \psi = 2\pi \frac{(\varphi - 1)}{\varphi}$
 - Fix the increment of the radius r by the factor $da = \frac{R}{k_{\max}} = \frac{\overline{Gb}}{k_{\max}}$
Step 4: - Determine the positions of the centers belonging to the logarithmic spiral golden angle with center G and radius $r = a$ according to the following formula:

$$\begin{cases} C_{kx} = G_x + a \times \cos(\theta) \\ C_{ky} = G_y + a \times \sin(\theta) \end{cases}$$
 With $\theta = \theta + d\theta$; $a = a + da$ and $k = 1, \dots, k_{\max}$.
Step 5: - Save the positions of the centers found in $C = \{c_1, \dots, c_k, \dots, c_{k_{\max}}\} \in R^d$.
End

B. Evaluation Measures

Using an unsupervised clustering algorithm, such as k-means algorithm, requires the determination of the number k of groups leading to the execution of the algorithm repeatedly for different values of this parameter.

For optimal number of groups, a criterion should be used to evaluate the result of the algorithm. This criterion is known as the validity index [1], [20]-[22], [15, 16, 17, and 18] name based on the notions of compactness and separation.

In literature, there are a lot of validity indexes, most of them are based on the notions of compactness within different groups and the separability between these different groups. In this article, the Davies-Bouldin index and the Mean Squared Error will be used as two validity indexes of neural classifiers.

C. Davies-Bouldin Index

This index takes into account both of the compactness and the separability of groups [23]. Its value is much lower than

the groups are compact and well separated. It promotes hyperspherical groups and is, therefore, particularly well-suited for a use with the k-means algorithm. The I_{DB} index is defined by the following expression:

$$I_{DB} = \frac{1}{K} \sum_{i=1}^k \max_{i \neq j} \frac{\{d_c(c_i) + d_c(c_j)\}}{D_{cc}(c_i, c_j)} \quad (6)$$

Where $d_c(c_i)$ is the average distance between an object and its group c_i following the center and $D_{cc}(c_i, c_j)$ is the distance between the centers of groups c_i and c_j with:

$$d_c(c_i) = \frac{1}{N_i} \sum_{l=1}^{N_i} \|x_l - c_j\| \quad (7)$$

$$D_{cc}(c_i, c_j) = \|c_i - c_j\| \quad (8)$$

D. Mean Squared Error

The Mean Squared Error is frequently used to assess the risk of an estimator. It is also useful to relay the concepts of bias, precision, and accuracy in statistical estimation. In this work, the MSE was used for groups' compactness measure for each centroid [24]; it is notably equivalent to the Euclidean function of the k-means algorithm:

$$E = \sum_{l=1}^N \sum_{j=1}^k \delta_{il} \|x_l - c_j\|^2 \quad (9)$$

$$\text{With: } \delta_{il} \begin{cases} 1 & \text{if } x_l \in c_i \\ 0 & \text{else} \end{cases}$$

V. NEW ALGORITHMS FOR THE CONSTRUCTION OF THE HIDDEN LAYER OF THE RBF CLASSIFIER

Two new algorithms were used to characterize the hidden layer classifier i.e. to determine the number of centers of different Gaussian and the value of each center.

In what follows, the principle of the proposed classifiers is presented. It is also explained how the two validity indexes (IDB and MSE) are combined with the k-means algorithm Mahalanobis distance to determine automatically the number k of groups.

However, it is necessary to fix a maximum number of centroids k_{\max} . The k_{\max} value can be defined by the user if he/she knows the structure of his database. Given that it is not always the case, the Bezdek [19] suggestion is adopted, so the $k_{\max} = \sqrt{N}$ (N is the size of the database) is chosen.

Applying these algorithms to all classes and summing the number of the obtained groups, the number of neurons in the hidden layer is determined. A neuron is then assigned to each group. For this RBF classifier, the database was partitioned $X = \{x_1, x_2, \dots, x_N\} \in R^d$ in individual blocks according to the

number of output classes $\Omega_j = 1, 2, \dots, m$. The database $X_d = (X_{\Omega_1} \ X_{\Omega_2} \ \dots \ X_{\Omega_m})^T$ is obtained.

Then, we apply the Principal Component Analysis (PCA) to the data base X_d in order to reduce it to a new basis two-dimension $X_d^* = \{x_1, x_2\} \in R^2$.

Principal component analysis (PCA) is a widely used statistical technique for unsupervised dimension reduction. K-means clustering is a commonly used data clustering for performing unsupervised learning tasks [25].

The PCA is based on the calculation of averages, variances and correlation coefficients. The main basis of dimension reduction is that PCA picks up the dimensions with the largest variances.

The PCA allows, in the same time, a reduction of data and an easier interpretation in the treated domain, as the new dimensions are often very significant [26]. In this case, the two largest variances were chosen to represent the new database.

The next step is to determine the number of centers and the center position of each class $C_{\Omega_1} = \{c_1, \dots, c_k\} \in R^d$ through the classifier based on the k-means algorithm with Mahalanobis distance.

The centers of each class $\Omega_j = 1, \dots, m$ found in the $C = \begin{pmatrix} C_{\Omega_1} \\ \cdot \\ \cdot \\ C_{\Omega_m} \end{pmatrix}$ matrix are grouped and the k-means algorithm Mahalanobis distance is applied to the new positions of the centers $C^* = \{c_1^*, c_2^*, \dots, c_k^*\} \in R^d$.

To complete the construction of the hidden layer classifier, there is a second parameter to consider in the neurons, which is the width factor σ_j for each centroid c_j ($j = 1, \dots, k$). This factor is calculated using the following formula:

$$\sigma_j = \frac{1}{N_a \sqrt{8}} \sum_{i=1}^{N_a} \|x_i - c_j\| \quad (10)$$

Witch N_a represents the training data

A. Construction of the RBF classifier KMD-LS-IDB

The proposed algorithm (KMD-LS-IDB) based on the k-means algorithm with Mahalanobis distance combined with the Davies-Bouldin validity index. This classifier determining

the number and the centers values of the hidden layer for each class of the database is described below:

Begin
Input:
 - The block database $X_{\Omega_j} = \{X_{\Omega_1}, \dots, X_{\Omega_m}\} \in R^d$ of one class of data base X_d , taking the case of the block. (The same approach for other classes).
Output: - The position of each center $C_{\Omega_1} = \{c_1, \dots, c_k\} \in R^d$.

Step 1: - Determine the size n and the number of characters (the attributes) p of the data base X_{Ω_1} .
Step 2: - Initialize the minimum number of centroids $K_{\min} = 2$ and then look for the maximum number of centroids by $K_{\max} = \sqrt{n}$.
 - Initialize the variables $a = k = K_{\min}$ and $d = 1$.
Step 3: - Apply Algorithm *Init_Centres_Kmeans_Logarithmic_Spiral* which initializes the centers for kmeans algorithm of the data base X_{Ω_1} .
Step 4: - Repeat the following steps until $k = K_{\max}$.
Step 4.1: -If $k < K_{\max} + 1$ Then:
 - Take the following centers positions $C = \{c_1, \dots, c_a\} \in R^d$.
 - Deduce the number of the centers k .
End If
Step 4.2: - Apply k-means algorithm with Mahalanobis distance to determine the new positions of the $C^* = \{c_1^*, \dots, c_a^*\} \in R^d$ centers.
Step 4.3: - Calculate the compactness and separability of groups with the Davies- Bouldin Index: I_{DB} .
Step 4.4: - Save I_{DB} variable in the table called *Tab_IDB*.
Step 4.5: - Increment variables $a = a + 1, d = d + 1$.
Step 5: - Determine β : the I_{DB} lowest index of the table *Tab_IDB*.
 - Take the following centers positions $C_{\Omega_1} = \{c_1, \dots, c_{\beta+1}\} \in R^d$ as the optimal number required classifying Ω_1 class centers.
End

B. Construction of RBF classifier KMD-LS-MSE

The proposed algorithm (KMD-LS-MSE) based on the k-means algorithm with Mahalanobis distance combined with the Mean Squared Error validity index. This classifier determining the number and the centers values of the hidden layer for each class of the database is described below:

Begin

Input:

- The block database $X_{\Omega_j} = \{X_{\Omega_1}, \dots, X_{\Omega_m}\} \in R^d$ of one class of data base X_d , taking the case of the block. (The same approach for other classes).

Output: - the position of each center $C_{\Omega_1} = \{c_1, \dots, c_k\} \in R^d$.

Step 1: - Determine the size n and the number of characters (the attributes) p of the data base X_{Ω_1} .

Step 2: - Initialize the minimum number of centroids $K_{\min} = 2$ and then look for the maximum number of centroids by $K_{\max} = \sqrt{n}$.

- Initialize the variables $a = k = K_{\min}$ and $d = 1$.

Step 3: - Apply Algorithm *Init_Centres_Kmeans_Logarithmic_Spiral* which initializes the centers for kmeans algorithm of the data base X_{Ω_1} .

Step 4: - Repeat the following steps until $k = K_{\max}$

Step 4.1: -If $k < K_{\max} + 1$ Then:

- Take the following centers positions $C = \{c_1, \dots, c_a\} \in R^d$.
- Deduce the number of centers k .

End If

Step 4.2: - Apply **k-means** algorithm with **Mahalanobis** distance to determine the new positions of the $C^* = \{c_1^*, \dots, c_a^*\} \in R^d$ centers.

Step 4.3: - Calculate the compactness and separability of groups with the Mean Squared Error Index: M_{SE} .

Step 4.4: -Save M_{SE} variable in the table called *Tab_MSE*.

Step 4.5: - Increment variables $a = a + 1, d = d + 1$.

Step 5: - Determine β : the M_{SE} lowest index of the table *Tab_MSE*.

- Take the following centers positions $C_{\Omega_1} = \{c_1, \dots, c_{\beta+1}\} \in R^d$ as the optimal number required classifying Ω_1 class centers.

End

C. Calculation of synaptic weight

After determining the parameters of the proposed classifier hidden layer, the learning is finished by the calculation of the synaptic weight w_{ij} , connecting the hidden layer neurons to those of the output layer.

The linearity property of the outputs $y_j(x_i)$ of the network is used. The expression of each of the m outputs is written as:

$$y_j(x_i) = h_j(x_i) \times w_{ij} \tag{11}$$

The global output of the network is written as follows:

$$Y = H \times W \tag{12}$$

The objective is to determine the matrix W that minimizes an error function, chosen as the square of the sum of classification errors.

The weight of the output layer can be calculated by the following matrix equation:

$$\underbrace{\begin{bmatrix} \varphi_{11} & \varphi_{12} & \dots & \varphi_{1M} \\ \varphi_{21} & \varphi_{22} & \dots & \varphi_{2M} \\ \dots & \dots & \dots & \dots \\ \varphi_{N1} & \varphi_{N2} & \dots & \varphi_{NM} \end{bmatrix}}_H \times \underbrace{\begin{bmatrix} w_1 \\ w_2 \\ \dots \\ w_M \end{bmatrix}}_W = \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_N \end{bmatrix}}_Y \tag{13}$$

With

H : The matrix of Radial Basis Functions

Y : The matrix of the output layer.

W : The weight matrix of centroids.

φ_{ij} : The Gaussian widths of the matrix H .

The above equation is giving by:

$$H \times W = Y \Rightarrow W = H^{-1} \times Y \tag{14}$$

Given that the H matrix is rarely square, the pseudo-inverse of the matrix H is applied according to equation (15):

$$W = [H^T \times H]^{-1} \times H^T \times Y \tag{15}$$

VI. EVALUATION OF RBF CLASSIFIERS

The purpose of this section is to evaluate the performance and efficiency of the proposed RBF neural classifiers. The performance of these RBF neural networks classifiers is tested with two different databases: Iris and Wine among the different data sets available from the machine learning community by the University of California at Irvine (UCI) [27].

The first test is carried out with the Wine database which includes the results of a chemical analysis of types of wine produced in different regions of Italy from different grapes varieties. The concentration of 13 components are indicated for each of the 178 types of wine (patterns) which are analyzed and divided into three classes (59 in Class 1, 71 in Class 2 and 48 in Class 3).

The second test is done with Iris database which is one of the most popular data set to examine the performance of novel methods in pattern recognition and machine learning. It is composed of three classes (i.e., iris Setosa, iris Versicolor and iris Virginica) each having 50 patterns with four features.

To evaluate the proposed classifiers performances, the holdout method is used. It consists on dividing the initial data into two independent sets: one for training and the other for testing the classifier performances.

The results given by the RBF classifier built with this algorithm are compared with those obtained with other neural

classifiers: the Learning Vector Quantization (LVQ) classifier proposed by Kohonen, the RBF neural network classifier for which the hidden layer is obtained using adaptive Pattern Classifier (APCIII) [28], the Multi-Layer Perceptrons classifier (MLP) and with a reference: the K nearest Neighbor (KNN).

The present comparative results of different classifiers over Wine and Iris are illustrated in *Table I* and *Table II*.

TABLE I. RESULTS OF THE RECOGNITION RATE OVER WINE DATABASE

Classification algorithms	Database: Wine
KMD-LS-IDB	98,88 %
KMD-LS- MSE	95,55 %
LVQ	66,14 %
APCIII	67,04 %
MLP	73,80 %
KNN	70,45 %

TABLE II. RESULTS OF THE RECOGNITION RATE OVER IRIS DATABASE

Classification algorithms	Database: Iris
KMD-LS-IDB	93,46 %
KMD -LS-MSE	93,46 %
LVQ	94,00 %
APCIII	93,33%
MLP	96,66 %
KNN	96,70 %

Considering Wine database, the best recognition rate is obtained by the KMD-LS-IDB proposed classifier and then the proposed classifier KMD-LS-MSE. For Iris database, the best recognition rate is given for the KNN classifier; however, the difference with the two proposed classifiers is not important.

Then, the proposed algorithms give good results in terms of recognition rate but the most powerful of them is the classifier KMD-LS-IDB.

VII. CONCLUSION

In this paper, new RBF neural networks classifiers has been designed to classify database. The proposed algorithms aim to deduce the centers of the hidden layer neurons and to calculate the number of the neurons in particular.

The basic idea of this approach is to select the training data from the database class by class and to decide about the optimal number of neurons in each class by using two different validity indexes (the validity index *IDB* and the *MSE*). This number is integrated in the k-means algorithm with the Mahalanobis distance.

A solution was also proposed to overcome the problem of initialization of centers necessary for the start of the K-means algorithm using the method of the logarithmic spiral golden angle.

The obtained classifiers results are satisfactory in comparison with other considered classifiers in literature for two real databases (Iris and Wine).

REFERENCES

[1] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On clustering validation techniques," *J. Intell. Inf. Syst.*, vol. 17, no. 2–3, pp. 107–145, 2001.

[2] J. K. Sing, D. K. Basu, M. Nasipuri, and M. Kundu, "Improved k-means algorithm in the design of RBF neural networks," *Conf. Converg. Technol. Asia-Pacific Reg. TENCON*, pp. 841–845, 2003.

[3] S. Song, Z. Yu, and X. Chen, "A novel radial basis function neural network for approximation," *Int. J. Inf. ...*, vol. 11, no. 9, pp. 46–53, 2005.

[4] B. Mhamdi, T. Aguil, and K. Grayaa, "Radial Basis Function Neural Network to Shape Reconstruction of Conducting Objects," *2012 6th Int. Conf. Sci. Electron. Technol. Inf. Telecommun. SETIT 2012*, vol. 1, no. 2, pp. 628–633, 2012.

[5] J. Moody and C. J. Darken, "Fast Learning in Networks of Locally-Tuned Processing Units," *Neural Comput.*, vol. 1, pp. 281–294, 1989.

[6] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Trans. Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.

[7] M. Vogt, "Combination of Radial Basis Function Neural Networks with Optimized Learning Vector Quantization," *Proc. ICNN'93, Int. Conf. Neural Networks*, vol. III, pp. 1841–1846, 1993.

[8] M. Kubat, "Decision trees can initialize radial-basis function networks," *IEEE Trans. Neural Networks*, vol. 9, pp. 813–821, 1998.

[9] N. Siddique and H. Adeli, "Spiral Dynamics Algorithm," vol. 23, no. 6, pp. 1–24, 2014.

[10] B. Al-shboul and S. Myaeng, "Initializing K-Means using Genetic Algorithms," *Int. J. Comput. Electr. Autom. Control Inf. Eng.*, vol. 3, p. 6, 2009.

[11] Sifaoui, A. Abdelkrim, and M. Benrejeb, "On the Use of Neural Network as a Universal Approximator," *Int. J. Sci. Tech. Autom. Control Comput.*, vol. 2, no. July, pp. 386–399, 2008.

[12] Sifaoui, A. Abdelkrim, and M. Benrejeb, "On New RBF Neural Network Construction Algorithm for Classification," *SIC*, vol. 18, no. 2, pp. 103–110, 2009.

[13] M. W. Guerfala, A. Sifaoui, and A. Abdelkrim, "RBF Neural Network Construction Algorithm for Classification based on Mahalanobis distance," in *ACECS'15, 2015*, no. 1, pp. 1–5.

[14] R. H. Bacon, "Logarithmic Spiral: An Ideal Trajectory for the Interplanetary Vehicle with Engines of Low Sustained Thrust," *Am. J. Phys.*, vol. 27, no. 3, p. 164, 1959.

[15] G. Markowsky, "Misconceptions about the Golden Ratio," *Coll. Math. J.*, vol. 23, no. 1, pp. 2–19, 1992.

[16] L. D. G. Sigalotti and A. Mejias, "The golden ratio in special relativity," *Chaos, Solitons & Fractals*, vol. 30, no. 3, pp. 521–524, 2006.

[17] R. J. Praga-Alejo, L. M. Torres-Treviño, D. S. González-González, J. Acevedo-Dávila, and F. Cepeda-Rodríguez, "Analysis and evaluation in a welding process applying a Redesigned Radial Basis Function," *Expert Syst. Appl.*, vol. 39, no. 10, pp. 9669–9675, 2012.

[18] M. W. Guerfala, A. Sifaoui, and A. Abdelkrim, "Construction of an RBF Classifier Based on Logarithmic Spiral," in *ACECS'16, 2016*, pp. 1–6.

[19] J. C. Bezdek, R. J. Hathaway, M. J. Sabin, and W. T. Tucker, "Convergence Theory for Fuzzy C-Means: Counterexamples and Repairs," *Syst. Man Cybern. IEEE Trans.*, vol. 17, no. 5, pp. 873–877, 1987.

[20] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "Clustering algorithms and validity measures," *Proc. Thirteen. Int. Conf. Sci. Stat. Database Manag. SSDBM 2001*, pp. 3–22, 2001.

[21] M. Halkidi and M. Vazirgiannis, "Clustering Validity Assessment: Finding the optimal partitioning of a data set," in *In Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on, 2001*, pp. 187–194.

[22] M. Sassi, A. G. Touzi, and H. Ounelli, "Using Gaussians Functions to Determine Representative Clustering Prototypes," pp. 1–5, 2006.

[23] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 1, no. 2, pp. 224–227, 1979.

[24] H. Rady, "Reyni ' s Entropy and Mean Square Error for Improving the Convergence of Multilayer Backpropagation Neural Networks: A Comparative Study," no. October, 2011.

[25] Ding and X. He, "K-means Clustering via Principal Component Analysis," *Int. Conf. Mach. Learn.*, 2004.

- [26] S. Jaffali and S. Jamoussi, "Principal component analysis neural network for textual document categorization and dimension reduction," 2012 6th Int. Conf. Sci. Electron. Technol. Inf. Telecommun. SETIT 2012, vol. 1, no. 2, pp. 835–839, 2012.
- [27] L. Blake and C. J. Merz, "UCI Repository of machine learning databases," University of California. p. <http://archive.ics.uci.edu/ml/>, 1998.
- [28] Y. Hwang and S. Bang, "An Efficient Method to Construct a Radial Basis Function Neural Network Classifier," Neural Networks, 1997.