# Adaptation of Multilayer Perceptron Neural Network to unsupervised Clustering using a developed version of k-means algorithm

HARCHLI Fidae, JOUDAR Nour-eddine, Es-SAFI Abdelatif, ETTAOUIL Mohamed

Scientific Computing and Computer Sciences, Engineering Sciences
Modeling and Scientific Computing Laboratory
University of SIDI MOHAMED IBN ABDELAH
Faculty of sciences and techniques
FES, MOROCCO

fidae.harchli@gmail.com, nour.jdr@gmail.com, essafiabdelatifi@hotmail.com,
mohamedettaouil@yahoo.fr

*Abstract:* - Cluster analysis plays a very important role in different fields and can be mandatory in others. This fact is due to the huge amount of web services, products and information created and provided on the internet and in addition the need of representation, visualization and reduction of large vectors. So in order to facilitate the treatment of information and reducing the research space, data must be classified. In other words, the needless of having a good technique of clustering is continually growing. There exist many clustering algorithms (supervised and unsupervised) in the literature: hierarchical and non hierarchical clustering methods, k-means, artificial neural networks (RNAs)…. All of these methods suffer from some drawbacks related to initialization issues, supervision or running time. For instance, the classes' number, initial code vectors and the choice of the best learning set in k-means and Multi Layer Perceptron (MLP) affect seriously the clustering results. To deal with these problems, we develop a new approach of unsupervised clustering. This later consists of using a developed version of k-means algorithm which determines the number of clusters and the best learning set in order to train the MLP in an unsupervised way. The effectiveness of this approach is tested on well-known data sets and compared to other classifiers proposed by recent researches.

*Key-Words:* - MLP Neural Network, Retro-propagation, Supervised and Unsupervised Learning, Cluster Analysis, k-means algorithm, parameters initialization, Assessment of Classification

## 1 Introduction

Cluster analysis is an important tool and a main task in data mining. It consists of partitioning the data set into clusters in such way that objects in the same cluster are more similar to each other than those in other clusters. It has many advantages that facilitate many tasks in different problems among them we find:

- Reduction of time: the consuming time to access to the desired information is reduced and facilitated. This favor is more important when the size of data set is great.
- Reduction of dimension: classification gives many solutions to reduce the dimension of vectors
- Representation and visualization of large vectors: elements which belong to the same cluster can be represented by the center of the cluster

Clustering techniques are used in many fields, including machine learning, pattern recognition, image analysis, information retrieval, text mining, recommendation and bioinformatics. A large category of methods proposed in the literature are sacrificed to cluster analysis. This variety of methods is divided into two groups: unsupervised and supervised methods. In the first group, any information about the examples is known or used by the method. In this case the data is called unlabeled data. For the second group, some information about data are known like the number of classes and definition of a training set in which each example is a pair consisting of an input object and a desired output value.

Artificial neural networks have proved their effectiveness as supervised classifiers thanks to their supervised learning. The multi layer perceptron is a famous artificial neural network which is widely

used in different fields thanks to his ability to classify, predict, approximate and so one. The MLP is known by its supervised learning so, to use this neural network in cluster analysis, it's compulsory to know the number of classes and the group of labeled data. What sets our contribution in this work apart from others is the use of the MLP neural network in unsupervised learning by proposing an adding step before the learning procedure. This previous step is very important to know the structure of data, have an idea about the number of clusters and construct a number of labeled objects of the data. It consists of conducting a nested series of partitions generated by a new version of k-means algorithm. The partition which maximizes the homogeneity within each class and the heterogeneity between classes represents the best. To check out this later, all generated partitions are examined by an index of classification. The proposed adding step is inspired from our work figured in [1]. Based on this later and MLP neural network, we develop a new unsupervised clustering method without having any prior information even the number of classes.

The paper proceeds as follows: Section 2 is divided into three overview parts: the first one and the second are reserved to an introduction of the MLP neural network and K-means algorithm. The third part contains a description of unsupervised classification. Section 3 discusses some related work in the areas of classification. Section 4 details our proposed approach and implementation. In Section 5, we expose the simulation and validation of this contribution. Finally, Section 6 concludes the paper and discusses possible future extensions and suggestions for improvement.

# 2 Literature overview

## 2.1 MLP neural network

Artificial neural networks (ANNs) are an exited computing paradigm which is completely inspired from the biological neurons. ANNs was initially configured as an attempt to imitate the human brain and nervous systems which are primarily made up of neurons [2] [3]. This attempt takes into account that the brain is the highly complex, nonlinear and parallel information processing system.

Multilayer perceptron neural network (MLPNN) is the most common and popular type of neural networks in use today. It belongs to a class of neural networks called feed forward neural networks. MLPNN represents a generic function approximator and classifier. It's capable to approximate generic classes of functions, including continuous and integrable functions [4]. Also it can distinguish data that are not linearly separable. Thanks to these properties the MLP neural network has been widely used in a variety of microware modeling and optimization problems. This famous system was first introduced by M. Minsky and S. Papert in 1969 [3].

### 2.1.1 Architecture

MLPNN is a feed forward network with one or more layers (hidden layers) of units (hidden neurons) between the input and output layers. The neurons of the previous layer are always connected with those of the next one while the neurons in the same layer are not interconnected (fig. 1).

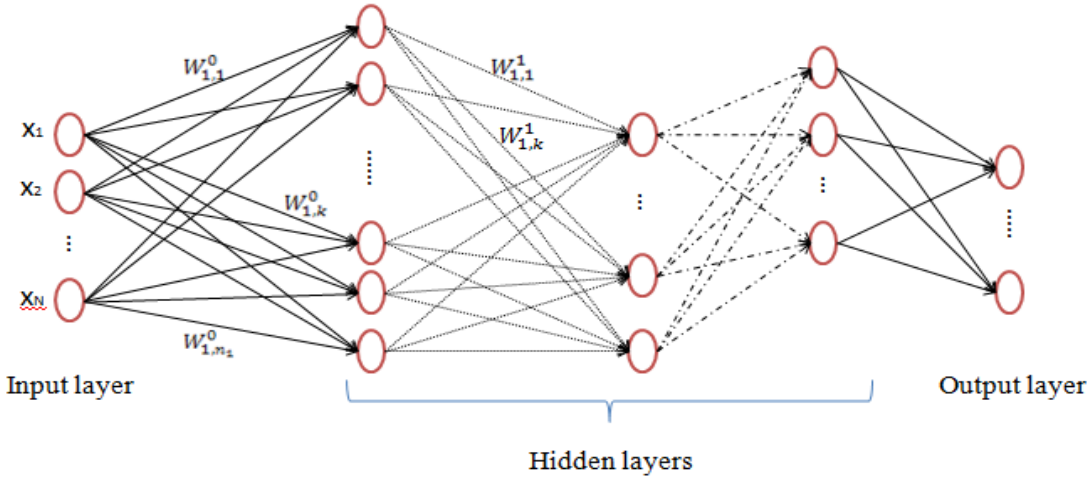


Fig. 1.Feed-forward neural network architecture

The number of hidden layers and hidden neurons are generated randomly. Many efforts [5][6][7][2] are employed to look for the optimal parameters of this layout of neurons grouped in layers. The number of

neurons in the input layer equals the dimension of observations while the number of neurons in the output layer is fixed a priori according to the purpose of the problem at hand.
The input layer is reserved to present the features to the fist hidden layer of the perceptron model. Hidden layers propagate the information from the inputs to the outputs (outputs of each hidden layer) along the network. The neurons of the output layer represent a hyper plane in the space of the input patterns.

### 2.1.2 Learning technique

Learning has undoubtedly attracted the most interest as a relevant contribution of neural networks to the sphere of information processing systems.
There are three basic classes of learning, each corresponding to a particular abstract learning task. These are supervised learning, unsupervised learning and reinforcement learning. These classes allow for [8]:

- Implementing a process with no need to know the mechanism, model, that lies beneath.
- Using the same model of neural network to deal with different tasks. Therefore an ANN can be regarded as a class of universal approximators that implements a nonlinear input-output mapping of a general nature [9].
- Adapting the system to changes in the surrounding environment.

The MLPNN utilizes a supervised learning technique. The goal of any supervised learning algorithm is to find a function that best maps a set of inputs to its correct output. It consists of adjusting the weights which are initially set by random values over a small range. Updating weights continue until the output units of the net for each set of data inputs are as close as possible to the desired outputs.
Mathematically speaking, the learning technique searches to minimize the following error function:

$$e_i = d_i - y_i \qquad (1)$$

Where $d_i$ is the desired response and $y_i$ is the calculated output by the network in response to the input $x_i$.
There are various methods that are proposed for training the MLPNN, but the most common and successful one is the Back-propagation [10] [9]. This later as an abbreviation for "backward propagation of errors" is used in conjunction with an optimization method such as gradient descent.
This method requires a desired output for each input data set value in order to calculate the output units of the network. For this issue, it is usually considered to be a supervised learning method.
The back-propagation is divided into two phases: forward and backward.
In the first phase the training pattern's input propagate through the neural network. In this phase the propagation's output activations are generated as it is shown in figure 2. It finishes with the computation of the error function (1).
During the second phase, the error signal $e_i$ is propagated through the network in order to generate the deltas of all output and hidden neurons. The name of the back-propagation is due to the backward direction in this phase. The weights are adjusted so as to minimize the error function $e_i$.

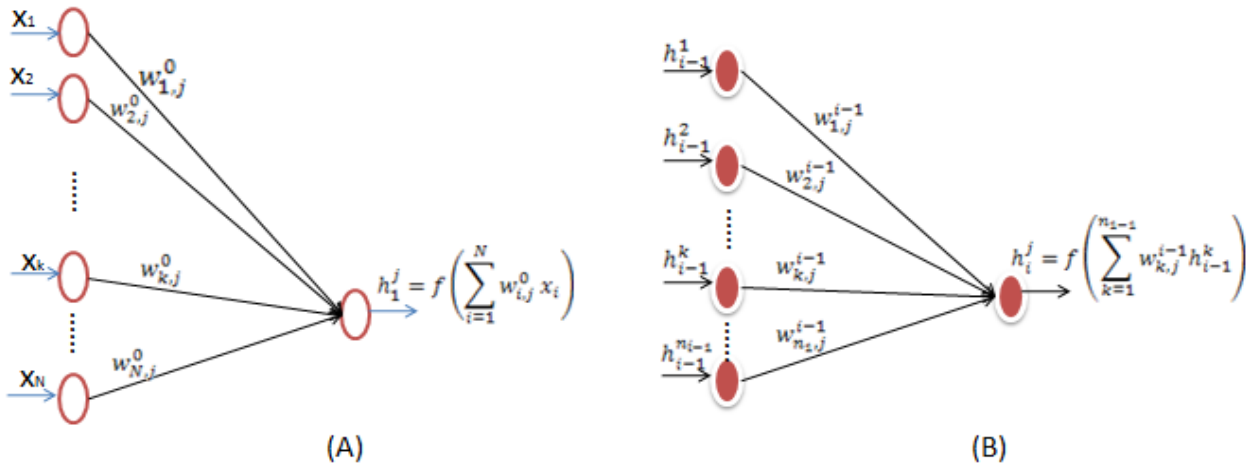


Fig. 2. Calculation of outputs for the first hidden layer and the others employing inputs, weights and the activation function: A) calculation of neuron's output in the first layer, B) calculation of neuron's output in the $i^{th}$ layer

## 2.2 K-means algorithm

The k-means algorithm is one of the most frequently used algorithms for classification using a squared error criterion [11]. It is a partitional and deterministic algorithm that attempts to partition the given data set into k different clusters by minimizing the distances between objects and cluster's centers.

Given a set of objects $\{x_1, \ldots\ldots\ldots\ldots, x_n\}$, where each object is represented by a d-dimensional vector, the k-means clustering algorithm partitions these n objects into k sets $\{s_1, \ldots\ldots, s_k\} / (k \leq n)$ such that the centroids $\{\mu_1, \ldots\ldots, \mu_k\}$ of provided classes are the minima of the following objective function:

$$min_{\{\mu_1,\mu_2,\ldots,\mu_k\}} \sum_{i=1}^{k} \sum_{x_j \in S_i} \|x_j - \mu_i\| \qquad (2)$$

This formula can be transformed as mathematic program which is defined as follow:

$$\begin{cases} \underbrace{min}_{x_{ij}} \sum_{j=1}^{j=m} \sum_{i=1}^{i=n} x_{ij} \left\| x_i - \frac{\sum_{i=1}^{i=n} x_{li}\, s_l}{l = \sum_{l=1}^{n} x_{li}} \right\|^2 \\ Sc \\ \sum_{j=1}^{j=m} x_{ij} = 1 \; (j = 1, \ldots\ldots, m) \\ \sum_{i=1}^{i=n} x_{ij} \geq 1 \; (i = 1, \ldots\ldots, n), \\ x_{ij} \in \{0, 1\} \; (j = 1, \ldots\ldots, m; \; i = 1, \ldots\ldots, n) \end{cases} \qquad (3)$$

Finding an exact solution to the function (2) is NP-hard. So a number of algorithms use heuristic approach to solve this problem. The k-means algorithm belongs to the set of those methods. Indeed The basic algorithm is composed of the following steps [12] [13]:

- Randomly select the number k and the coordinates of centroids.
- Assign each pattern to the group that has the closest centroid by determining the distance between each pattern and the centroids of classes.
- Once all patterns have been assigned, recalculate the new centroids of the k classes.
- If there is no further change, end the clustering task; otherwise return to step 2.

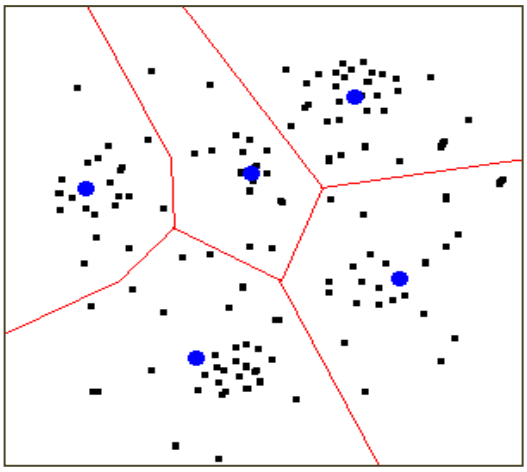

Fig. 3. The final step of k-means algorithm

The power of k-means algorithm has been demonstrated over the years by their successful use in many problems as a good tool of classification. Its main advantages can be summarized as follow:

- Rapidity: The comparison is conducted only between the observations and the center of classes
- Easiness: the process of this algorithm is very easy to implement
- Detection of outliers: It detects and isolates the outliers which disturb the process of classification
- Efficiency in dealing with large amounts of data

Notwithstanding the above, this method suffers from some shortcomings. Among them we present the following:

- The choice of initial parameters: the number of classes and the initial centers are set randomly.
- The choice of similarity Function to measure the similarity between the objects: The Euclidian distance is the useful measure but in certain case it is not the most appropriate.
- It can deal only with spherical and symmetrical point distribution

## 2.3 Unsupervised classification

The classification task consists of partitioning a data set E into an ensemble of classes $C_i$. This partition must satisfy two objectives:

- Maximize the similarity of objects belonging to the same cluster.
- Minimize the similarity among objects in different clusters.

There exist two types of classification: supervised and unsupervised.
In this work we are interested in the unsupervised classification because it is the most frequently used. Unsupervised classification aims to partition the given data without having any prior knowledge about the structure or the number of clusters (in an unsupervised way). It has been addressed in many contexts and by researchers in many disciplines. Because the majority of problems are unsupervised, it has a broad appeal and usefulness as one of the steps in exploratory data analysis.
The more information we know about data, the easier the classification becomes. However, clustering huge amounts of data is a difficult task since the goal is to find a suitable partition in an unsupervised way. Many different clustering techniques have been defined in order to solve the problem [1][14][15].

## 3 Related work

The MLP neural network has been used in many types of problems with different degrees of complexity and in different fields of application [16][17][18][19][20]. Since the literature on this topic is extensive, we concentrate only on some recent works.
Youssef Safi and Abdelaziz Bouroumi [21] present and discuss the application of the feed forward neural network to the real-world problem of predicting forest fires. In this article, the authors aim to predict, in function of all parameters involved in forest fires, the total surface of a forest area that might be burned if nothing is done in order to prevent the catastrophe. The authors of [22] use a MLP to label every pixel of multispectral satellite images in order to identify regions of interest such as water, soil, forest, land, etc. In order to train the MLP, they used either a labeled data set called: round truth data or the output of a K-means clustering program or both. The study presented in this work shows that the application of the MLP leads to a more accurate and faster classification of multispectral image data. Also the authors of [23] used the output of K-means algorithm as an input of MLP neural network (MLPNN). This later is introduced as a diagnostic decision support mechanism in the epilepsy treatment.
Another article [24] presents a new application of MLPNN based classification by classifying the blood characteristics for thalassaemia screening. Authors aim to classify eighteen classes of thalassaemia abnormality and one control class by inspecting data characterized by a complete blood count (CBC) and Haemoglobin typing. To have a good classification in a short time, authors of [5] propose a mathematical model to find the optimal number of hidden layers and hidden neurons of the MLPNN.
The presented works in the literature have proved the effectiveness of the famous MLPNN since it requires little computation time and satisfactory result.

## 4 Description of the proposed method

In this section we present the proposed method with more details.
The present paper aims to realize the following objectives:

- Develop a package of parameterless clustering algorithm.
- Determine automatically the appropriate number of classes and the leaning set.
- Use the multilayer perceptron in unsupervised case in order to obtain the rule of classification.

To achieve those goals, we perform an iterative process which contains three interesting techniques:

- New proposed heuristique
- K-means algorithm
- Multi layer perceptron

The proposed method consists of two phases: In the first one, we divide the set of observations into a suitable number of classes, then, in the second one, from each class, we choose properly a subset of observations. Using those later as a learning set, we apply the multilayer perceptron. More precisely, this method can be described in the following steps:

**First Phase:**

1. Among the set of observations E, we look for the two most distant observations $x_1, x_2$ in E.
2. Using the principle of the nearest neighbor to $x_1$ and $x_2$ , we form two subsets E1, E2.
3. We apply K-means algorithm using the centroids of E1 and E2.
4. We identify the most heterogeneous class from the set of classes obtained in the previous step.
5. We divide the most heterogeneous class into two subsets by applying the technique presented in the first and second step.
6. We apply K-means algorithm to the barycenters of the obtained classes.

**By using two specific criteria (4), the first phase will be arrested when the best number of classes is obtained which corresponds to specific values of these criteria.**

**Second phase:**

7. We choose from each obtained class a subset of observations whose silhouette criteria (5) is great than a certain threshold. Those observations are used as a learning set to apply the multilayer perceptron.
8. We classify the data using the conventional MLPNN.

We present a summarization of the proposed method with more details in figure 4, 5 and 6 to make it clearer.

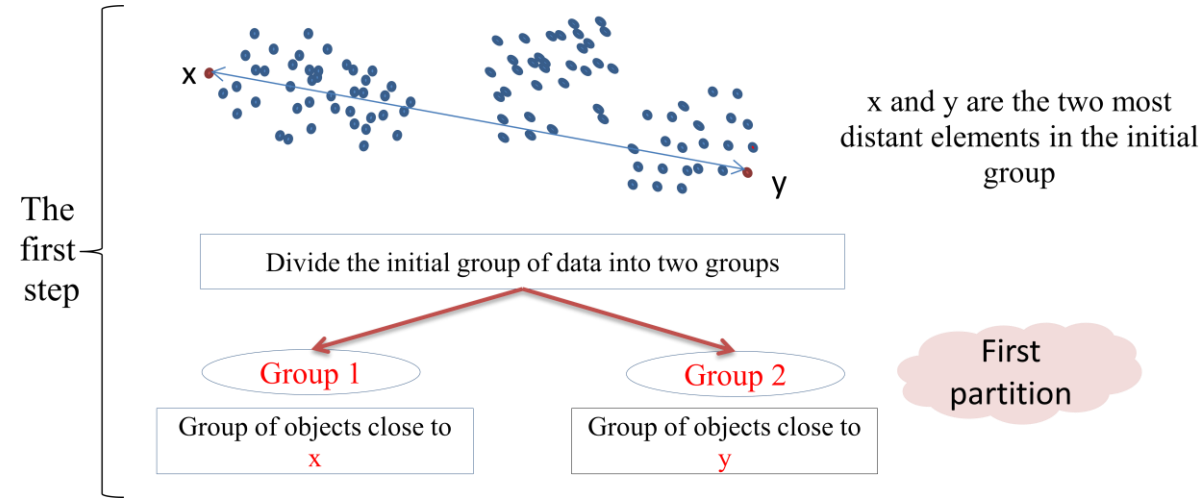


Fig. 4. Description of the main idea in the first step which will be used in all steps of the first phase of the proposed method

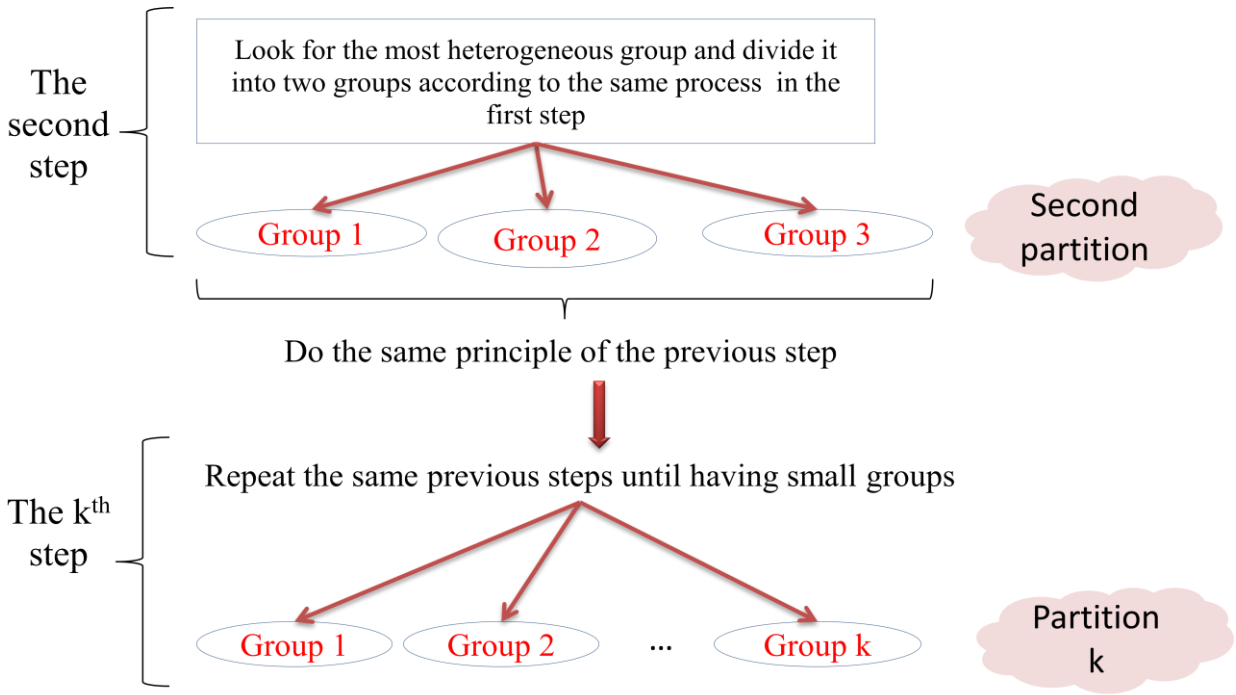


Fig. 5. Division of the most heterogeneous group in each step according to the process of the first step until having small groups

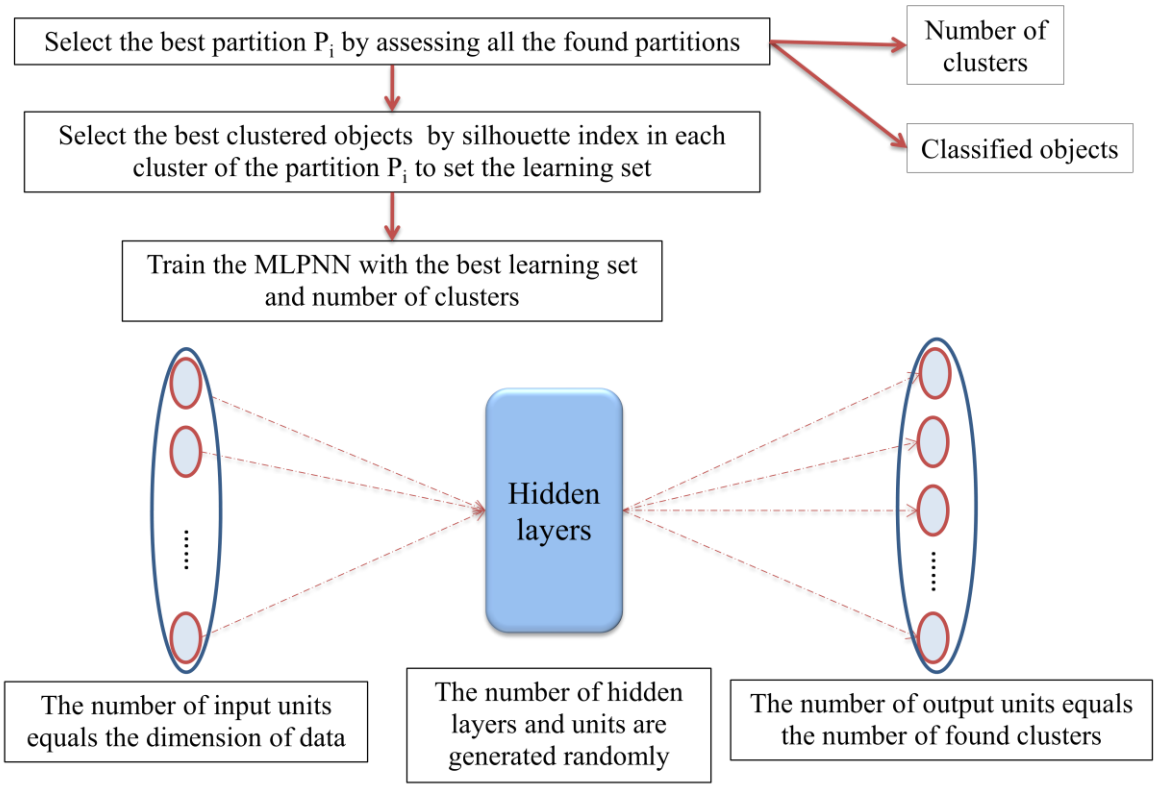


Fig. 6. Training of the constructed MLPNN with the obtained learning set and number of clusters to be adapted to the clustering problem

In this step we define the indexes used to select the learning set and the number of clusters.
Thanks to $R^2$ and $F$, we evaluate the validity of the number of classes. Those criteria are defined as follows:

$$R^2 = \frac{I_B}{I} \qquad F = \frac{R^2/k-1}{1-R^2/n-k} \tag{4}$$

The number of classes which realizes the first small difference between two successive values of $R^2$ is retuned as the suitable number of classes. More precisely, this index finds the suitable number when it satisfies a compromise between the following two conditions:

- The value of $R^2$criterion corresponding to this number of classes must be closer to 1 without having a lot of classes.
- Difference between two successive values of $R^2$ have to be so large.

The silhouette index is a measure of how appropriately the data has been clustered. The silhouette index of each datum varies between -1 and 1. An $s(i)$ close to one means that the datum is appropriately clustered. In contrast if it is close to negative one, the datum is not appropriately clustered. And an $s(i)$ close to zero means that the datum is on the border of two clusters. The silhouette index is defined as follow:

$$\forall\, i \in X; s(i) = \frac{b(i)-a(i)}{max\{a(i), b(i)\}} \tag{5}$$

Where:

$$\begin{cases} a(i) = \dfrac{1}{|C_i|-1} \displaystyle\sum_{\substack{x_j \in C_i \\ i \neq j}} d(x_i, x_j), & x_i \in C_i \\ b(i) = \underbrace{\min}_{\substack{C_i \in P \\ C_i \neq C_j}} \{d(x_i, C_j)\}, & x_i \in C_i \\ d(x_i, C_j) = \dfrac{1}{|C_j|} \displaystyle\sum_{x_j \in C_j} d(x_i, x_j) \end{cases}$$

## 5 Computer & Simulation

In this section we aim to validate the performance of the proposed method and show the capacity of the adapted MLPNN to deal with unlabeled data sets. To do that we run the proposed method on five

benchmark data sets token from UCI machine repository [25].

| Data sets | Total Number of patterns | Number of attributes | Number of classes |
|---|---|---|---|
| **Iris** | **150** | **4** | **3** |
| **Wine** | **178** | **13** | **3** |
| **Soybean** | **47** | **35** | **4** |
| **Hayes-Roth** | **132** | **5** | **3** |
| **Glass** | **214** | **10** | **6** |

Table 1.Description of the used data sets

In Table 1, we give some information about the number of patterns, number of attributes and the number of classes in each used data set. The used data sets are from different types. They have different sizes and number of classes to make sure that our method can treat many types of data. To have more details about the nature of these sets, we refer the reader to [25].

The first experiment aims to show the process of the proposed method to obtain the appropriate number of classes with the variation of the obtained number of classes and the two indexes $R^2$ and F.

| Data set | Number of classes | $R^2$ | $F$ |
|---|---|---|---|
| **IRIS** | **1** | **0.1194** | **9. 8936** |
| | **2** | **0 . 1315** | **5 . 4518** |
| | **3** | **0.1362** | **3. 7317** |
| | **4** | **0.1357** | **2. 7468** |
| | **5** | **0.1381** | **2. 2102** |
| | **6** | **0.1390** | **1. 8292** |
| | **7** | **0.1399** | **1. 5566** |
| | **8** | **0.1420** | **1. 3649** |
| | **9** | **0.1424** | **1. 1995** |
| | **10** | **0.1426** | **1. 0644** |
| | **11** | **0.1430** | **0.9560** |
| | **12** | **0.1431** | **0.8630** |

Table 2.Variation of $R^2$ and $F$ according to the obtained number of classes for data IRIS

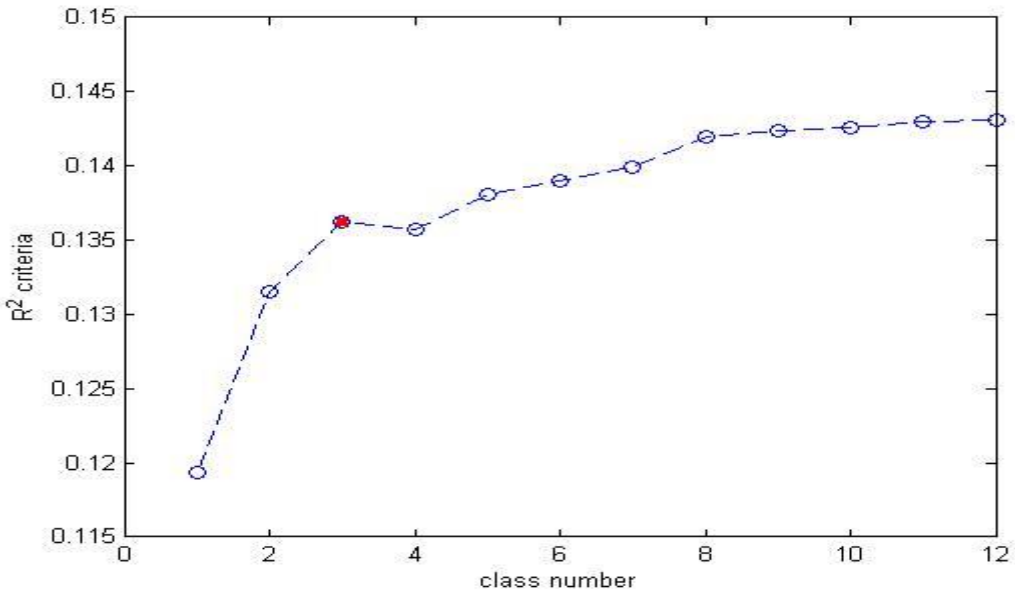


Fig. 7.Variation of $R^2$ according to the obtained number of classes (IRIS dataset)

| Data set | Number of classes | $R^2$ | $F$ |
|---|---|---|---|
| WINE | 1 | 0.0521 | 4. 0084 |
| | 2 | 0.0742 | 2. 8857 |
| | 3 | 0.0796 | 2. 0474 |
| | 4 | 0.0773 | 1. 4666 |
| | 5 | 0.0787 | 1. 1795 |
| | 6 | 0.0798 | 0.9823 |
| | 7 | 0.0797 | 0.8286 |

Table 3. Variation of $R^2$ and $F$ according to the obtained number of classes for data WINE

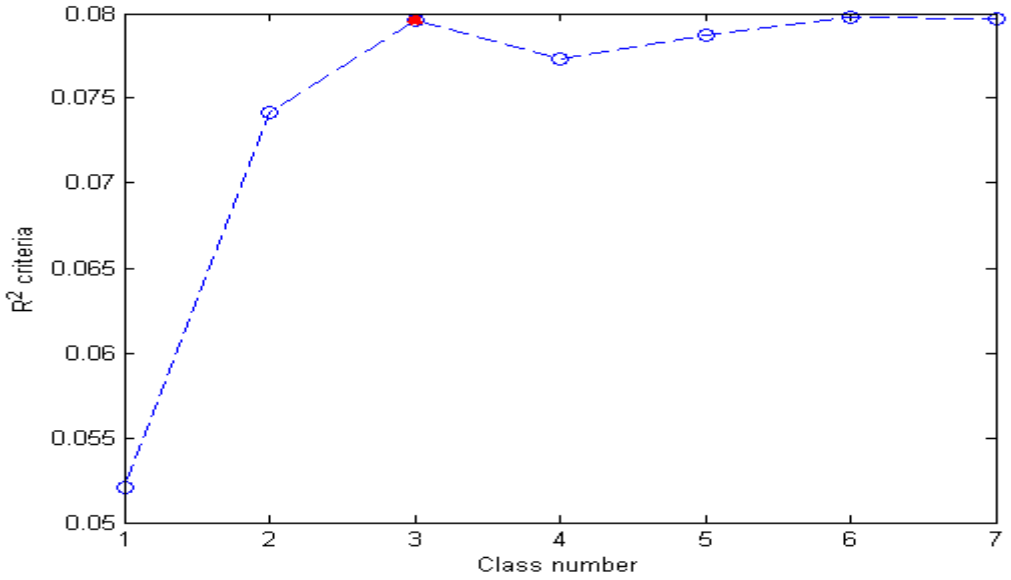


Fig. 8.Variation of $R^2$ according to the obtained number of classes (WINE dataset)

| Data set | Number of classes | $R^2$ | $F$ |
|---|---|---|---|
| SOYBEAN | 1 | 0.0920 | 7. 3948 |
| | 2 | 0.0954 | 3. 7946 |
| | 3 | 0.1040 | 2. 7458 |
| | 4 | 0.1098 | 2. 1579 |
| | 5 | 0.1045 | 1. 6112 |
| | 6 | 0.1034 | 1. 3077 |
| | 7 | 0.0987 | 1. 0482 |
| | 8 | 0.0910 | 0.8260 |
| | 9 | 0.0915 | 0.7278 |

Table 4. Variation of $R^2$ and F according to the obtained number of classes for data SOYBAN

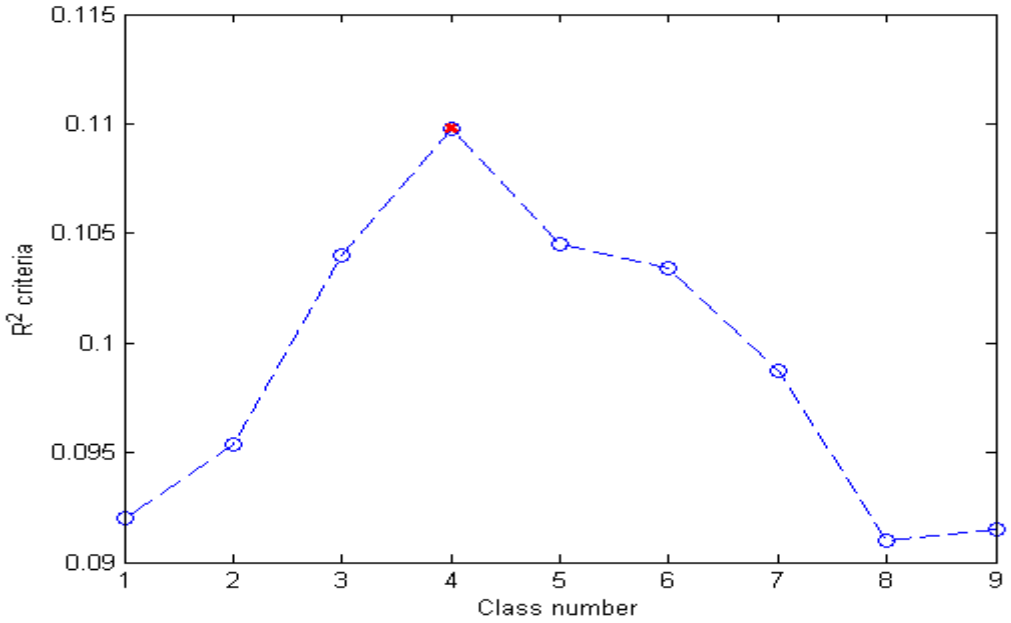


Fig. 9.Variation of $R^2$ according to the obtained number of classes (SOYBEAN dataset)

| Data set | Number of classes | $R^2$ | $F$ |
|---|---|---|---|
| **Hayes-Roth** | **1** | **0.0865** | **5.9665** |
| | **2** | **0.0999** | **3.9960** |
| | **3** | **0.1078** | **2.8604** |
| | **4** | **0.1094** | **2.1500** |
| | **5** | **0.1148** | **1.7889** |
| | **6** | **0.1066** | **1.3518** |
| | **7** | **0.1067** | **1.1427** |

Table 5. Variation of $R^2$ and F according to the obtained number of classes (Hayes-Roth dataset)

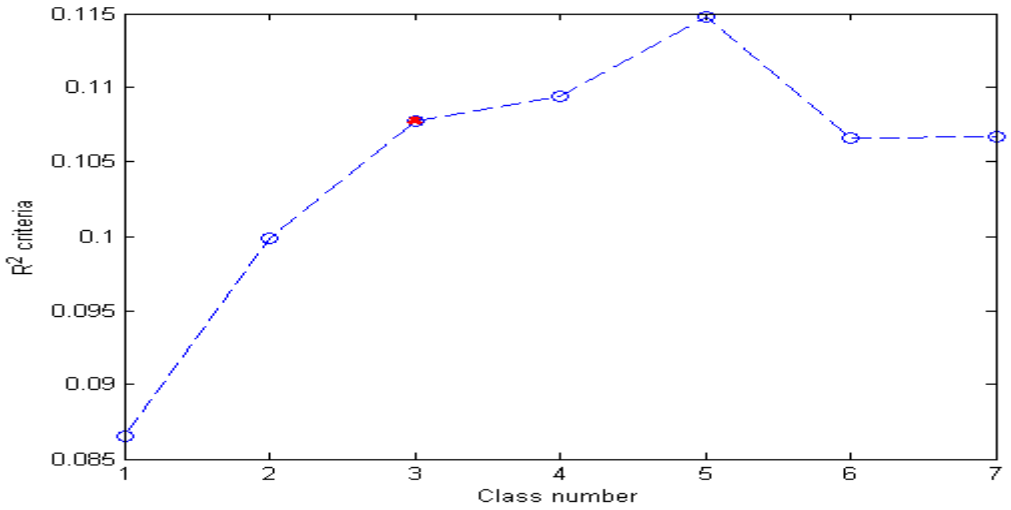


Fig. 10.Variation of $R^2$ according to the obtained number of classes (**Hayes-Roth** dataset)

| Data set | Number of classes | $R^2$ | $F$ |
|---|---|---|---|
| Glass | 1 | 0.1035 | 7.2694 |
| | 2 | 0.1107 | 4.4805 |
| | 3 | 0.1112 | 2.9623 |
| | 4 | 0.1098 | 2.1574 |
| | 5 | 0.1110 | 1.7233 |
| | 6 | 0.1098 | 1.3972 |
| | 7 | 0.1098 | 1.1809 |
| | 8 | 0.1048 | 0.9655 |
| | 9 | 0.1036 | 0.8350 |
| | 10 | 0.1029 | 0.7343 |
| | 11 | 0.1053 | 0.6741 |
| | 12 | 0.1034 | 0.5957 |
| | 13 | 0.0967 | 0.5022 |
| | 14 | 0.0973 | 0.4618 |
| | 15 | 0.0946 | 0.4109 |
| | 16 | 0.0950 | 0.3804 |
| | 17 | 0.0945 | 0.3499 |

Table 5. Variation of $R^2$ and $F$ according to the obtained number of classes (Glass dataset)

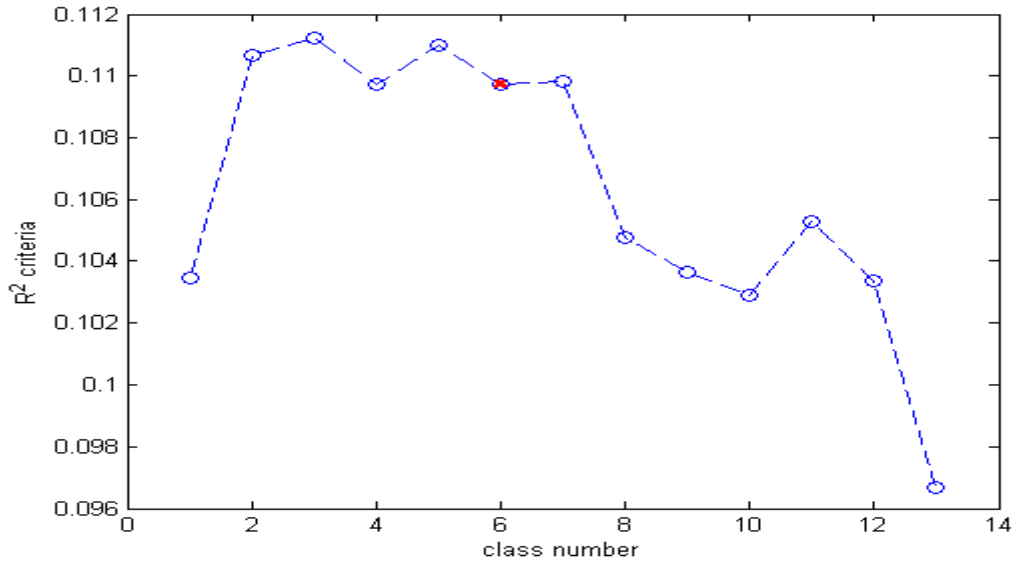


Fig. 10.Variation of $R^2$ according to the obtained number of classes (Glass dataset)

The previous Tables and curves show that the proposed method finds the exact number of classes for all used data. Indeed, the value of $R^2$ corresponding to the founded number of classes is 0.1357 for IRIS, 0.0796 for WINE, 0.1098 for SOYBEAN, 0.1078 for Hayes-Roth and 0.1098 for Glass. According to these values the number of clusters is found. This result is very interesting because the problem of finding the exact number of clusters for unlabeled data has attracted the attention of many authors in the past as nowadays.

The second experiment aims to choose the appropriate learning set to train the MLPNN. To do that, we vary a threshold representing the minimum of the silhouette of all objects. So we select the objects having a silhouette close to one which will participate to train the MLPNN.

| Data Set | Threshold of similarity | Rate of classification |
|---|---|---|
| IRIS | 0.65 | 82.66% |
| | 0.7 | 89.33% |
| | 0.8 | 89.33% |
| | 0.9 | 93.33% |
| WINE | 0.5 | 73.33% |
| | 0.6 | 74% |
| | 0.7 | 78.88% |
| | 0.8 | 78.88% |
| | 0.9 | 80% |
| | 0.94 | 80% |
| SOYBEAN | 0.4 | 34% |
| | 0.6 | 72% |
| | 0.7 | 84.5% |
| | 0.85 | 89.36% |
| Hayes-Roth | 0.3 | 81% |
| | 0.4 | 81,30% |
| | 0.6 | 84% |
| | 0.7 | 89.23% |
| | 0.8 | 93,84% |
| Glass | 0.4 | 79.43% |
| | 0.6 | 87.5% |
| | 0.8 | 91.2% |

Table 6.Variation of the recognition rate of five data according to the variation of similarity threshold

The results of this experience are stored in Table 3. The first column describes data sets; whereas the column 2 specifies the threshold and 3 presents the rate of classification. In this experiment we vary the threshold and measure the rate of the masticated observations. The process is arrested at the iteration which realizes the little rate. Examining the results, we conclude that the choice of threshold has a great impact on the quality of classification. Since when we increase the threshold we give to the network compact groups for training step which train the network suitably.

To state our method from other methods of the literature, the next table shows the performance of our method by comparing the obtained result with the conventional k-means algorithm and a recent work in this field which use Voronoi diagram (VD) to surmount the problem of initialization of k-means algorithm [26].

Table 6 shows that the proposed procedure of the MLPNN leads to satisfactory results. The proposed unsupervised version works well in the case of the IRIS and SOYBEAN data set over the existing techniques, while VD method overcomes our result for WINE data set. Must be noted that our result can be improved using the recent versions of k-means method and MLP neural network as the improved k-means and VD method. We reported that our method takes into account more important goals than VD method and works in an unsupervised way i.e. without knowing any prior knowledge about data.

| Data set | Rate of clustering | | |
|---|---|---|---|
| | K-means | Our method | VD |
| IRIS | 85.33% | 93.33% | 88.67% |
| WINE | 70.78% | 81.46% | 90.44% |
| SOYBEAN | 76.6% | 89.36% | 82.98% |

Table 7.Comparison of the proposed scheme with k-means and VD for Iris, Wine and Soybean

To give more details about the efficiency of our method, we conduct other comparison with two methods from the family of neural networks: LVQ and RBF. The last table of this section shows a detailed comparison of the results.

| Data sets | Rate of clustering | | |
|---|---|---|---|
| | Our method | LVQ | RBF |
| Iris data | 93.33% | 92.55% | 85.64% |
| Wine data | 81.46% | 68.9% | 67.87% |
| Hayes-Roth | 93,4% | 52.26% | 70.03% |
| Glass | 91.2% | 60.69% | 69.54% |

Table 7.Comparison of the proposed scheme with ANN methods as LVQ and RBF for IRIS, WINE, HYES-ROTH and GLASS

The comparative tests revealed that in comparison to LVQ and RBF neural networks, the adapted version of MLPNN outperforms the other for

dataset IRIS, WINE, HAYES-ROTH and GLASS by giving very satisfying rate of recognition.

What we can infer from these remarks is that satisfying and competitive results can be obtained with our proposed method without having any prior idea about the data set. Then the proposed procedure is believed to be novel and advantageous in the unsupervised clustering and learning fields.

# 6 Conclusion

The clustering task is very important and a compulsory step in several disciplines. The MLPNN has shown a great success as a supervised classifier. Our contribution in this work was the adaptation of this famous neural network to the unsupervised classification. In this vein we faced the great problem of determining the learning set and the number of classes. This work focused on finding solution to this problem. For this issue, we proposed a novel unsupervised clustering method based on the MLPNN with the help of a developed version of k-means algorithm. The proposed method is experienced on five benchmark data sets from different types chosen from UCI machine repository. The provided results prove that our method realizes the envisaged objectives and gives satisfying and competitive results with other supervised method. So we can consider our proposed method as a total package of a parameterless clustering algorithm thanks to the automatic detection of classes' number and the learning set.

The finding results encourage us to improve this method by adjusting some parameters as the the architecture of MLPNN and using some developed models of MLPNN as hybrid MLPNN.

*References:*

[1] M. ETTAOUIL, A. ES-SAFI, F. HARCHLI, "Improving the Performance of K-means Algorithm using an Automatic Choice of Suitable Code Vectors and Optimal Number of Clusters", *International Journal of Theoretical and Applied Information Technology*, Vol. 56, No. 3, 2013, pp. 428-436.

[2] M. Ettaouil, Y. Ghanou, "Neural architectures optimization and Genetic algorithms", *Wseas Transactions on Computer*, Vol. 8, No 3, 2009, pp. 526-537.

[3] Cichoki, R. Unberhuen, "*Neural Network for Optimization and signal processing*", Jhon Wiley & Sons, New York, 1993.

[4] F. Scarselli, and A. C. Tsoi, "Universal Approximation Using Feed Forward Neural Networks: A Survey of Some Existing Methods, and Some New Results", *Neural Networks*, Vol. 11, 1998, pp. 15-37.

[5] M. Ettaouil, M. Lazaar, Y. Ghanou, "Architecture Optimization Model for the Multilayer Perceptron and Clustering", *Journal of Theoretical and Applied Information Technology*, Vol. 47, No. 1, 2013, pp. 64-72.

[6] P. A. Castillo, J. J. Merelo, A. Prieto, V. Rivas, G. Romero, "GProp: global optimization of multilayer perceptrons using Gas", *Neurocomputing*, Vol. 35, 2000, pp. 149–163.

[7] J. Arifovic, R. Gencay, "Using genetic algorithms to select architecture of a feed forward artificial neural network", *Physica A*, Vol. 289, 2001, pp. 574–594.

[8] Editorial, "Advances in artificial neural networks and machine learning," *Neurocomputing*, Vol. 121, 2013, pp. 1–4.

[9] X. Liang, "Removal of hidden neurons in multilayer perceptrons by orthogonal projection and weight crosswise propagation", *Neural Computer & Application*, Vol. 16, 2007, pp. 57-68.

[10] X. Zeng and D. S. Yeung, "Hidden neuron pruning of multilayer perceptrons using a quantified sensitivity Measure", *Neurocomputing*, Vol. 69, 2006, pp. 825-837.

[11] J. MCQUEEN, "Some methods for classification and analysis of multivariate observations", *In Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1967, pp. 281–297.

[12] J. A. Hartigan, M. A. Wong, "A k-means clustering algorithm", *Applied Statistics*, Vol. 28, No. 1, 1979, pp. 100–108.

[13] M. David, "*An example inference task: clustering Information theory, inference and learning algorithms (Chapter 20)",* Cambridge University Press, 2003.

[14] M. ETTAOUIL, A. ES-SAFI, F. HARCHLI, "Optimizing the architecture of Kohonen map and classification", *Journal of Computing*, Vol. 4, No. 7, 2012, pp. 31-37.

[15] F. HARCHLI, A. ES-SAFI, M. ETTAOUIL, "Optimization of the Organized KOHONEN Map by a New Model of Preprocessing Phase and Application in Clustering", *Journal of Emerging Technologies in Web Intelligence*, Vol. 6, No. 1, 2014, pp. 80-84.

[16] C. M. Bishop, "*Neural Networks for Pattern Recognition*", Oxford University Press, Oxford, 1995.

[17] H. A. Bourlard, N. Morgan, "*Connectionist Speech Recognition: A Hybrid Approach*", Kluwer Academic Publishers, Dordrecht, 1993.
[18] T. Masters, "*Signal and Image Processing with Neural Networks: a C++ Sourcebook*", Wiley, New York, 1994.
[19] P. Cortez, M. Rocha, J. Neves, "Evolving time series forecasting arma models", *Journal of Heuristics*, Vol. 10, No. 4, 2004, pp. 415–429.
[20] M. Dirst, A. S. Weigend, "*Time Series Prediction: Forecasting the Future and Understanding the Past*", Addison-Wesley, Reading, MA, 1994.
[21] Y. Safi and A. Bouroumi, "Prediction of Forest Fires Using Artificial Neural Networks", *Applied Mathematical Sciences*, Vol. 7, No. 6, 2013, pp. 271 – 286.
[22] Y. V. Venkatesh, S. K. Raja, "On the classification of multispectral satellite images using the multilayer perceptron", *Pattern Recognition*, Vol. 36, No. 9, 2003, pp. 2161–2175.
[23] U. Orhan, M. Hekim, M. Ozer, "EEG signals classification using the K-means clustering and a multilayer perceptron neural network model", *Expert Systems with Applications*, Vol. 38, No. 10, 2011, pp. 13475-13481.
[24] S. Damrongrit, P. Theera, W. Wongseree, Touchpong Usavanarong, Nuttawut Paulkhaolarn, Chompunut Kanjanakorn, Monchan Sirikong, Chanin Limwongse, Nachol Chaiyaratana, "Classification of complete blood count and haemoglobin typing data by a C4. 5 decision tree, a naïve Bayes classifier and a multilayer perceptron for thalassaemia screening", *Biomedical Signal Processing and Control*, Vol. 7, No. 2, 2012, pp. 202-212.
[25] A. Asuncion and D. J. Newman, UCI Machine Learning Repository, Irvine, CA: University of California, School of Information and Computer Science, (2007). Available: http://archive. ics. uci. edu/ml/datasets/Forest+Fires.
[26] R. Damodar, K. J. Prasanta, "Initialization for K-means clustering using Voronoi diagram", *Procedia Technology*, Vol. 4, 2012, pp. 395-400.