

A SINGLE-LAYER SEMI-SUPERVISED FEED FORWARD NEURAL NETWORK CLUSTERING METHOD

RoyaAsadi¹, Sameem Abdul Kareem², Mitra Asadi³, Shokoofeh Asadi⁴

^{1,2}Department of Artificial Intelligence, Faculty of Computer Science and Information Technology, University of Malaya, Kuala Lumpur, Malaysia.

³Department of Research, Iranian Blood Transfusion Organization, Tehran, Iran.

⁴Department of Agricultural Management Engineering, Faculty of Ebne-Sina, University of Science and Research Branch, Tehran, 930277941, Iran.

Email: ¹ royaasadi @siswa.um.edu.my, ²sameem@um.edu.my, ³m.asadi@ibto.ir, ⁴shokoofeh.ame@gmail.com

ABSTRACT

The aim of this research is to develop and propose a single-layer semi-supervised feed forward neural network clustering method with one epoch training in order to solve the problems of low training speed, accuracy and high time and memory complexities of clustering. A code book of non-random weights is learned through the input data directly. Then, the best match weight (BMW) vector is mined from the code book, and consequently an exclusive total threshold of each input data is calculated based on the BMW vector. The input data are clustered based on their exclusive total thresholds. Finally, the method assigns a class label to each input data by using a K-step activation function for comparing the total thresholds of the training set and the test set. The class label of other unlabeled and unknown input test data are predicted based on their clusters or trial and error technique, and the number of clusters and density of each cluster are updated. In order to evaluate the results, the proposed method was used to cluster five datasets, namely the breast cancer Wisconsin, Iris, Spam, Arcene and Yeast from the University of California Irvin (UCI) repository and a breast cancer dataset from the University of Malaya Medical center (UMMC), and their results were compared with the results of the several related methods. The experimental results show the superiority of the proposed method.

Keywords: *Artificial Neural Network, Feed Forward Neural Network, Supervised and Unsupervised learning, Semi-supervised Clustering, Real weight.*

1.0 INTRODUCTION AND RELATED WORK

An artificial neural network has its roots in mathematics, statistics, numerical analysis, biology and psychology, and is one of the numerous algorithms used in machine learning and data mining [1-5]. A feed-forward neural network (FFNN) is a software version of the human brain. In this network, data processing has only one forward direction from the input layer to the output layer without any cycle or backward movement [6-8]. Learning is a necessary feature of the neural network in machine learning, and learning rules are categorized broadly under supervised learning, unsupervised learning, and reinforcement learning [6, 9-11]. Supervised learning is similar to unsupervised training in the sense that the training set is provided. However, in supervised training the desired output is provided and the weight matrix is applied based on the difference between the predicted output and the actual output of the neural network. Most approaches to unsupervised learning in machine learning are statistical modelling, compression, filtering, blind source separation, and clustering. Unsupervised learning or self-organized learning finds symmetries in the data represented by input instances with unlabeled data. However to assess the performance of unsupervised learning, there is no error or reward signal. In this study, the clustering aspect of unsupervised neural network classification is considered [10-12]. Unsupervised FFNN (UFFNN) clustering generally exhibits several advantages. These advantages include an inherent distributed parallel processing architecture, as well as capabilities to adjust the interconnection weights to learn and describe suitable clusters, process vector quantization prototypes and distribute similar data without the need for class

labels, control noisy data, cluster unknown data, and learn the types of input values on the basis of their weights and properties [6, 9, 12-14]. UFFNN clustering methods often use Hebbian learning, competitive learning, or competitive Hebbian learning. Hebb [15] developed the meaning of the first learning rule and proposed the Hebbian learning. Fig. 1 illustrates the single layer UFFNN as a simple topology with Hebbian learning [16]. Hebb described a synaptic flexibility mechanism in which the synaptic connection between two neurons is strengthened, and neuron j becomes more sensitive to the action of neuron i if the latter is close enough to stimulate the former while repeatedly contributing to its activation.

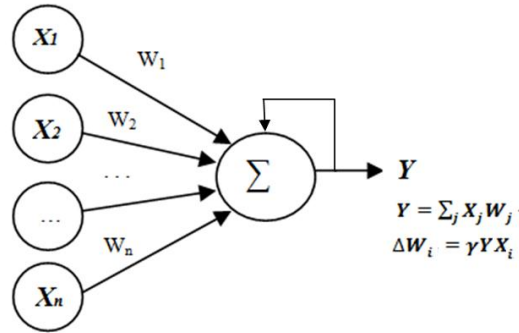


Fig.1: Single layer Unsupervised Feed Forward Neural Network with the Hebbian learning.

The Hebbian rule is shown in Equation (1):

$$(1) \quad \Delta W_i = \gamma Y X_i$$

where X is the input vector; Y is the output vector and γ is learning rate, where $\gamma > 0$ is used to control the size of each training iteration. The competitive learning network is a UFFNN clustering based on learning the nearest weight vector to the input vector as the winner node according to the computing distance, such as Euclidean. Fig. 2 shows a sample topology of an unsupervised neural network with competitive learning [17, 18]:

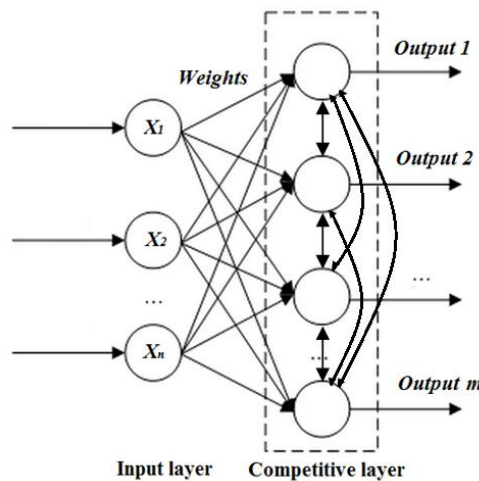


Fig.2: A sample topology of the competitive clustering.

The similarities between Hebbian learning and competitive learning include unsupervised learning without error signal, and are strongly associated with biological systems. However, in competitive learning, only one output must be active; such that only the weights of the winner are updated in each epoch. The updating of weights only considers the learning rate and input data from the input layer. By contrast, no constraint is enforced by neighbouring nodes in Hebbian learning, and all weights are updated at each epoch. The updating of weights considers the learning rate, input data from the input layer, and output data. In the case of competitive Hebbian learning, the neural network method shares some properties of both competitive learning and Hebbian learning [16, 19, 20]. The growing neural gas (GNG) [21] method is an example which uses the competitive Hebbian learning, in which the connection between the winning node and the second nearest node is created or updated in each training cycle. Competitive learning can apply vector quantization (VQ) [22] during clustering. VQ [22], *K*-means [2] and some UFFNN clustering methods such as Kohonen's self-organizing map (SOM) [23], neural gas (NG) [24] and GNG [21], are generally considered as the fundamental patterns in the current unsupervised feed-forward neural network clustering methods in stationary and online dynamic environments [25]. Linde et al. [22] introduced an algorithm for VQ design to obtain a suitable code book of weights for input data nodes clustering. VQ is based on the probability density functions by distribution of vectors of the weights. This method is often used for data compression. VQ divides a large set of the data (vectors) into clusters, each of which is represented by its centroid node, as in the *K*-means and some other clustering algorithms. VQ is a powerful tool for use in large and high-dimensional data. Data points are represented by the index of their closest centroid, such that commonly occurring data are more accurate. *K*-means [2] is a partitioning clustering method that uses a centroid-based technique similar to VQ. The main problem of partitioning methods lies in defining a special number of clusters and initializing the steps before the clustering tasks [26, 27]. Although *K*-means clustering is inefficient when applied to large datasets, this method can efficiently be applied if the initialization of steps is well defined [13, 27]. NG [24] is based on VQ and data compression. NG dynamically partitions itself in a manner similar to gas and describes the number of clusters. The weight vectors are randomly initialized. The NG algorithm is faster and results in a more accurate clusters when compared with other algorithms. However, NG fails to control the network of nodes by either deleting or adding a node dynamically during clustering. The GNG [21] method can follow dynamic distributions by adding nodes and deleting them in the network during clustering by using the utility parameters. First, two random nodes from the input data are selected, after which network competition is initiated for the highest similarity to the input pattern. During learning, related data nodes are classified as similarities within clusters, whereas unrelated data nodes are identified as dissimilarities within clusters. The disadvantages of the GNG include the increase in the number of nodes to obtain the input probability density and requirement for predetermining the maximum number of nodes and thresholds [28-30]. SOM [23] maps multi-dimensional data onto lower dimensional subspaces, with the geometric relationships between points indicates their similarity. SOM generates subspaces with unsupervised learning neural network training through a competitive learning algorithm. The weights are adjusted based on their proximity to the "winning" nodes, that is, the nodes that most closely resembles a sample input [31-34]. Currently, the UFFNN clustering methods suffer from the major problems of the low clustering speed, accuracy and an effective memory complexity [25, 29, 35, 36]. Briefly, two sources of these problems are the structure and features of the data, and the topology and algorithm of the current UFFNN clustering methods. Clustering of some kinds of data is so difficult because of their character and structure. For example, clustering the medical data sets is difficult because of limited observation, information, diagnosis and prognosis of the specialist; incomplete medical knowledge; and lack of enough time for diagnosis [37]. The current UFFNN clustering methods generally use random weights, thresholds and parameters for controlling tasks during clustering, such as the SOM that creates the code book of weights by selection of the input data randomly. Random initialization of weights results in the paradox of low accuracy and high training time [13, 36, 38]. The clustering process is considerably slow because the weights have to be updated in each epoch during learning. Utilizing suitable weights and parameters is necessary because the neural network relies on the "garbage-in, garbage-out" principle. Therefore, the problem also affects memory usage [6, 11, 39-41]. The parameter values are often selected by trial and error after several executions of the clustering model, and the clustering method often uses many parameters to manage clustering performance, such as measuring the distance threshold, controlling the spread of the neighbourhood, controlling the growth of the number of clusters and size of the network, learning rate, and steps of learning time [35, 38, 42]. On the other hand, relearning during several epochs takes time and clustering is considerably slow. Relearning affects the clustering accuracy, and time and memory complexities for clustering. [30, 43-46]. There is a technique of converting clustering method for semi-clustering by considering some constraint or user guides as feedback from users. Several literatures devoted to improve the UFFNN clustering methods by using constraints such as class labels. The constraints of class labels are based on the knowledge of the experts and the user guide as partial supervision for better controlling the tasks of clustering and desired results [47]. The UFFNN clustering methods have the capability to develop into

semi-clustering method by obtaining the feedback of users [47-49]. Generally, in order to improve the UFFNN method, the users manage and correct the number of clusters and density of each cluster by inserting and deleting the data nodes and clusters. However, the judgment of users can be wrong or they may make mistakes during the insertion, deletion or finding the link between nodes and assigning the class label to each disjoint sub-cluster. Commonly after clustering, the semi-UFFNN methods such as semi-SOM method [50] assigns a class label to the nearest node or winning node and consequently assigned the same class labels to its neighbour nodes in its cluster. Each cluster must have a unique class label, if the data nodes of a cluster have different class labels, the cluster can be divided into different sub-clusters, and move some far nodes in the cluster with weak link to other clusters. However, assigning the class labels of the data nodes between the clusters can be somewhat vague [47, 49, 50]. In this paper, a single-layer semi-supervised feed forward neural network clustering method is developed and proposed to overcome the mentioned problems.

2.0 METHODOLOGY

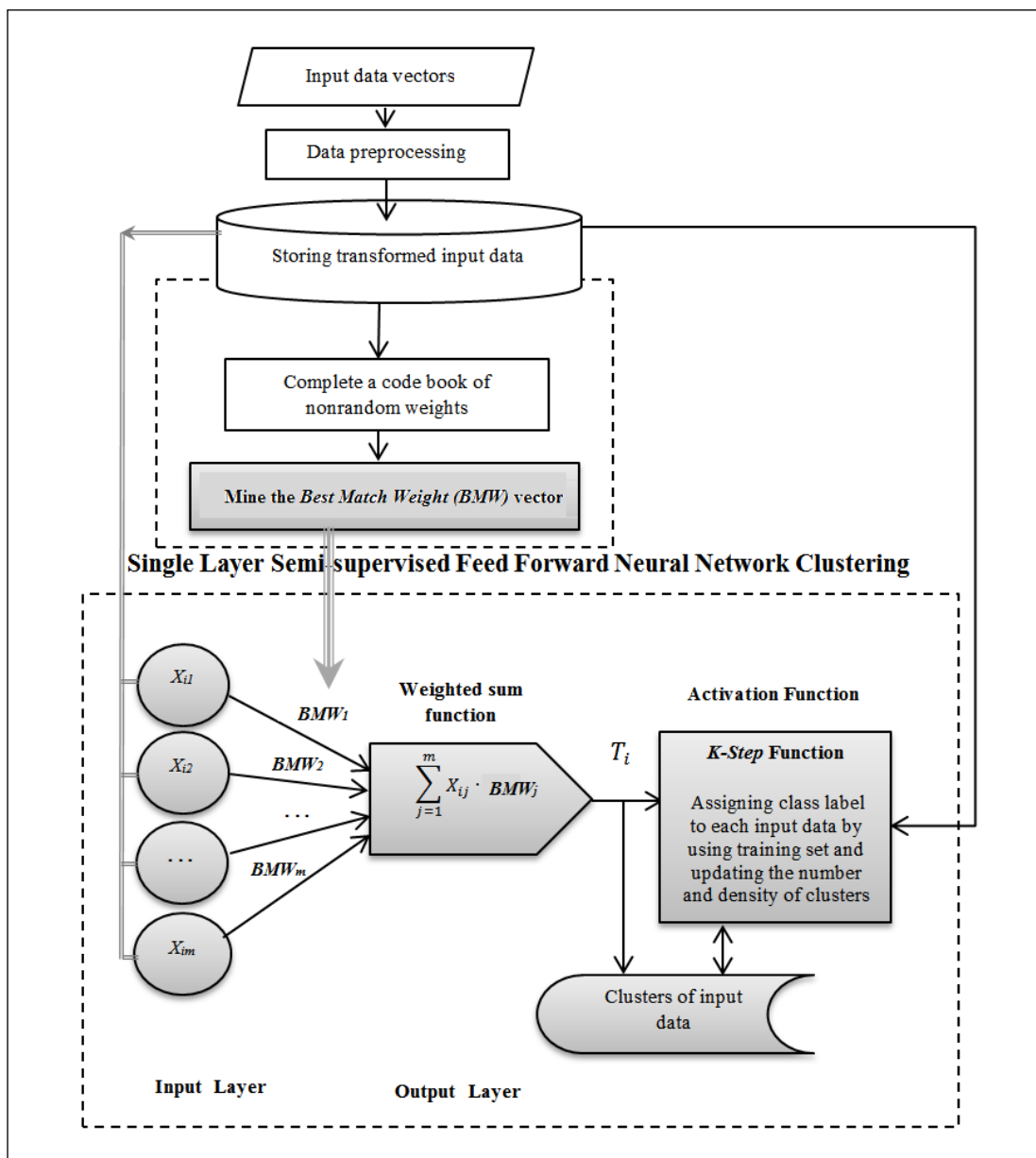


Fig.3: The design of the semi-supervised feed forward neural network clustering method.

We developed and proposed an efficient single-layer semi-supervised feed forward neural network (SFFNN) clustering method with only one epoch training time in order to overcome the problems of the low speed and accuracy of clustering and high memory complexity.

The SFFNN method computed a code book of real weights by using values of input data directly without using any random values. Consequently, the threshold of each input data was computed based on the real weights without using any class label or constraint. Finally, the input data are clustered based on related thresholds. Then, the method assigns a class label to each input data by using K-step activation function for comparing the total thresholds of the training set and the test set. The class label of other unlabeled and unknown input test data are predicted based on their clusters, or trial and error technique. Finally, the number of clusters and density of each cluster are updated. In the next section, we will explain the stages of the SFFNN clustering method and how it solves the clustering problems. Fig. 3 shows the design of the SFFNN model for clustering.

The SFFNN clustering method involves several phases:

2.1. Data preprocessing

Commonly preprocessing is the donating feature in developing efficient techniques for low training time and high accuracy of feed forward neural network clustering [51-53]. In the SFFNN model, the MinMax normalization technique is used to transform an input value of each attribute to fit in specific range such as [0,1] [38, 51-53]. The input matrix of values consists of every single value with individual measurement unit type and range. The fundamental enterprise of the proposed method is that no missing values exist and every value is acceptable. For this purpose, other data preprocessing techniques such as data cleaning are valuable [51].

2.2. Creating a code book of non-random weights and mining the best match weight (BMW) vector

After data preprocessing phase, the SFFNN model applies the matrix of the normalized input values. Table 1 shows the matrix of the input dataset X after preprocessing.

Table 1: The dataset of X .

Input data vector X_i	Attribute ₁	Attribute ₂	...	Attribute _m	μ_i	σ_i
X_1	X_{11}	X_{12}	...	X_{1m}	μ_1	σ_1
X_2	X_{21}	X_{22}	...	X_{2m}	μ_2	σ_2
...
X_n	X_{n1}	X_{n2}	...	X_{nm}	μ_m	σ_m

The SFFNN method creates a code book of weights by using the *standard normal distribution (SND)* [54] of input data. Each normalized attribute value of the input data X_{ij} is considered as the weight W_{ij} for that value. Equations (1) and (2) shows this matter. The method receives other input values of the data and computes the code book of all weights of input data values. This phase can be processed in parallel.

$$W_{ij} = (X_{ij} - \mu_i) / \sigma_i$$

(1)

$$W_{ij} = SND(X_{ij}) \quad i = 1, 2, \dots, n; \quad j = 1, 2, \dots, m;$$

(2)

The X_{ij} is the j_{th} attribute value of input data i , μ_i and σ_i are the mean and standard deviation of the input data record. As a result, each W_{ij} shows the distance of each attribute value of input data from the mean of the input data. Therefore, the initialization of weights is not at random. The matrix of the weights of the input data values is used as the code book of non-random weights, as shown in Table 2.

Table 2: Design the code book of the non-random weights in the SFFNN.

Weight vector of X_i	Attribute ₁	Attribute ₂	...	Attribute _m
Weight vector of X_1	$W_{11} = SND(X_{11})$	$W_{12} = SND(X_{12})$...	$W_{1m} = SND(X_{1m})$
Weight vector of X_2	$W_{21} = SND(X_{21})$	$W_{22} = SND(X_{22})$...	$W_{2m} = SND(X_{2m})$
...
Weight vector of X_n	$W_{n1} = SND(X_{n1})$	$W_{n2} = SND(X_{n2})$...	$W_{nm} = SND(X_{nm})$

Consequently, the SFFNN method computes the *best match weight (BMW)* vector by training the real weights in the code book. The *BMW* vector is the extraction of the code book of real weights as a unique weight vector for clustering the input data of the dataset globally. The *BMW* is the essential feature of the SFFNN model. The *BMW* consists of the components BMW_j for the attributes which is computed based on Equation (3). The parameter n is the number of input data, i is the current number of the node of input data; m is the number of attributes and j is the current number of the attribute of input data. The Equation (3) and (4) show these relationships.

$$BMW_j = (\sum_{i=1}^n W_{ij}^2)^{1/2} \tag{3}$$

$$\overrightarrow{BMW} = (BMW_1, BMW_2, \dots, BMW_m) \tag{4}$$

Table 3 illustrates the process of computing for the *BMW* vector.

Table 3: The process of mining the *BMW* vector from the code book of weight vectors.

The code book of weight vectors				
Weight vector of X_i	Attribute ₁	Attribute ₂	...	Attribute _m
Weight vector of X_1	W_{11}	W_{12}	...	W_{1m}
Weight vector of X_2	W_{21}	W_{22}	...	W_{2m}
...
Weight vector of X_n	W_{n1}	W_{n2}	...	W_{nm}
\overrightarrow{BMW}	BMW_1	BMW_2	...	BMW_m

The learning of the SFFNN method does not require computing any error function such as the mean square errors and updating weights in any training cycle, therefore, resulting in a reduced training time. The next phases will show how the thresholds are computed and the dataset of input data is clustered easily based on just the *BMW* vector. Tables 4 and Table 5 show an example of the computed *BMW* components by the SFFNN clustering method on the breast cancer Wisconsin and the Iris datasets.

Table 4: The computed *BMW* vector of the Breast cancer Wisconsin (Original) dataset by the SFFNN method.

BMW_1	BMW_2	BMW_3	BMW_4	BMW_5	BMW_6	BMW_7	BMW_8	BMW_9
0.1051841	0.11762797	0.11613390	0.11607615	0.096447262	0.12462520	0.10280549	0.11876218	0.10233766
4	6	5	5	8	4	7	3	8

Table 5. The computed *BMW* vector of the Iris dataset by the SFFNN method.

<i>BMW1</i>	<i>BMW2</i>	<i>BMW3</i>	<i>BMW4</i>
0.2337617343	0.2433078972	0.2512679332	0.2716624351

2.3. A Single Layer Semi-supervised Feed Forward Neural Network clustering

The topology of the SFFNN is very simple, as illustrated in Fig. 3, one input layer with n nodes which is the same as the number of the attributes, and an output layer with just one node.

- Clustering:** The units of the input layer are fed by the transformed data values from the data pre-processing phase of the SFFNN method. Each unit of node has a related weight component BMW_j . The output layer has one unit with a weighted sum function for computing the actual desired output. The threshold or output is computed by using normalized values of input data and the BMW vector based on the torque vector [55] definition. Each X_{ij} by using its arm W_{ij} creates a torque vector ratio to the gravity centre of the training dataset. Fig. 4 shows an example of the normalized data value vector X_{ij} which creates its own torque vector ratio to the global mean or the gravity centre of the training dataset.

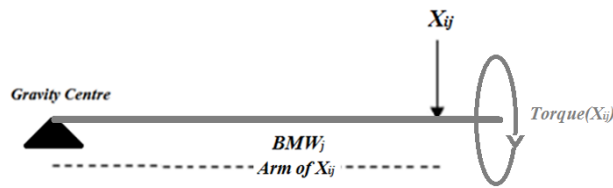


Fig.4: Distribution of normalized data attributes and their distances from the gravity centre of a training data set.

As shown in Fig. 4, each X_{ij} by using its arm BMW_j , which shows the distance of X_{ij} from the gravity centre of the matrix, creates a torque vector. Equation (4) shows the torque vector of each X_{ij} by using the related BMW_j which is equal the threshold T_i for each unit:

$$T_{ij} = X_{ij} \cdot BMW_j \tag{5}$$

As shown in Fig. 5, the all torque vectors of the attribute values of the input data are evaluated together and eventually will reach the equilibrium of the ratio to the gravity centre of the dataset. Therefore, after equivalence, the total torque vector is computed which is considered as the *total threshold* (TT_i) of each input data X_i . The total torque as exclusive TT_i of the actual output is computed by the weighted sum function. Equation (6) and (7) show the real TT_i for each input data vector X_i :

$$TT_i = \sum_{j=1}^m X_{ij} \cdot BMW_j \tag{6}$$

$$\text{OR } TT_i = \sum_{j=1}^m T_{ij} \tag{7}$$

Fig. 5 shows how the input data are clustered. As explained in Fig. 4, each TT_i is the total torque vector of each input data ratio to the gravity centre of the dataset. Therefore, each vector value of the X_i takes place its own position on the torque axis. Therefore, the input data based on their exclusive total thresholds lay on the torque axis respectively. Each input data vector X_i has an exclusive and unique threshold. The SFFNN considers the input data with near total thresholds

into one cluster. Needless to say, the clustering process is performed without using any class label or the initialization of the number of clusters.

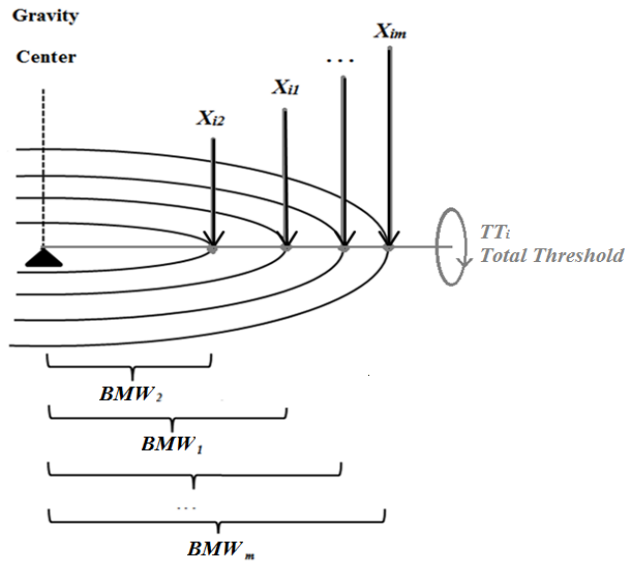
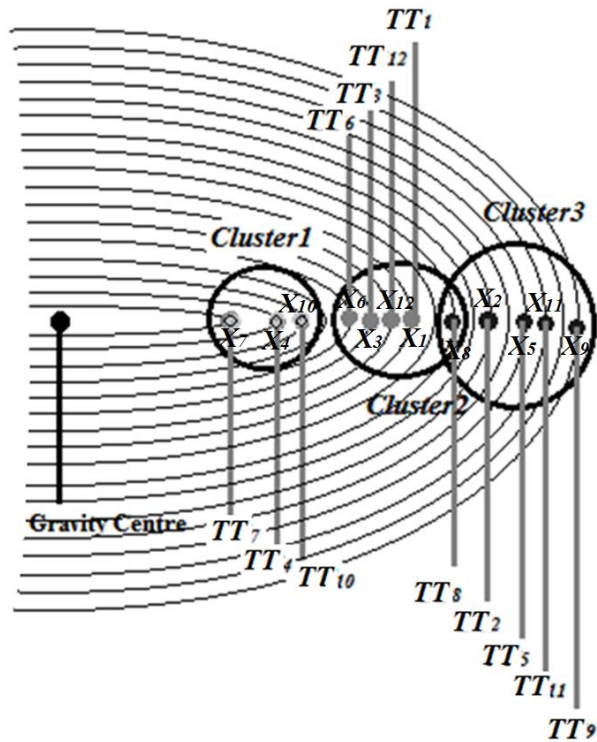


Fig. 5: Distribution of the normal input data attributes and their distances from the gravity centre of the dataset.

Fig. 6 is an example of clustering a dataset to three clusters which shows 12 data points with their special total thresholds and orders.



$$TT_7 < TT_4 < TT_{10} < TT_6 < TT_3 < TT_{12} < TT_1 < TT_8 < TT_2 < TT_5 < TT_{11} < TT_9$$

Fig. 6: An example of clustering a dataset to three clusters which shows seven data points with their special total thresholds and their orders by the SFFNN.

Fig. 7 and Fig. 8 are two examples of clustering by SFFNN method. In Fig. 7 the Breast cancer Wisconsin from UCI repository is clustered to two clusters based on a unique total threshold of each data point by the SFFNN method.

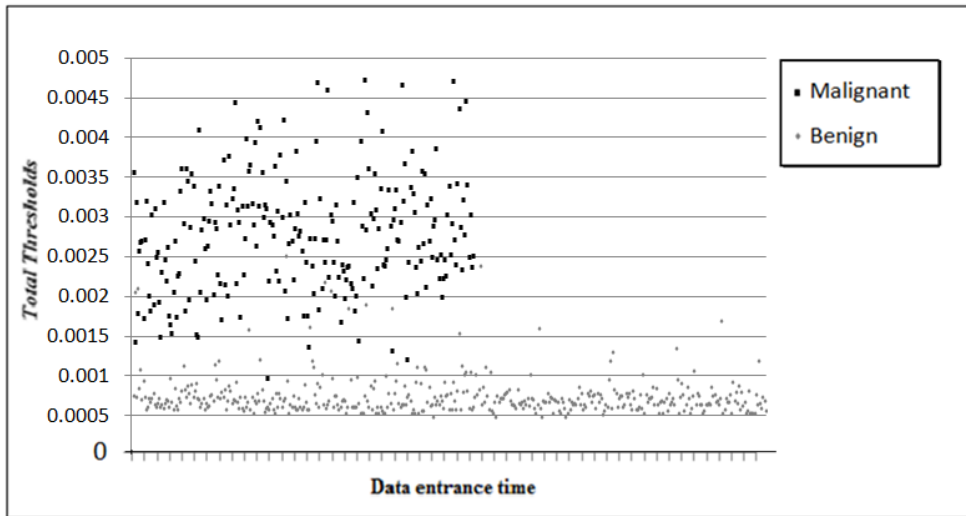


Fig. 7: The outlook of clustering the Breast cancer and Iris dataset by SFFNN.

In Fig. 8 the Iris data from UCI repository is clustered to three clusters based on a unique total threshold of each data point by the SFFNN method.

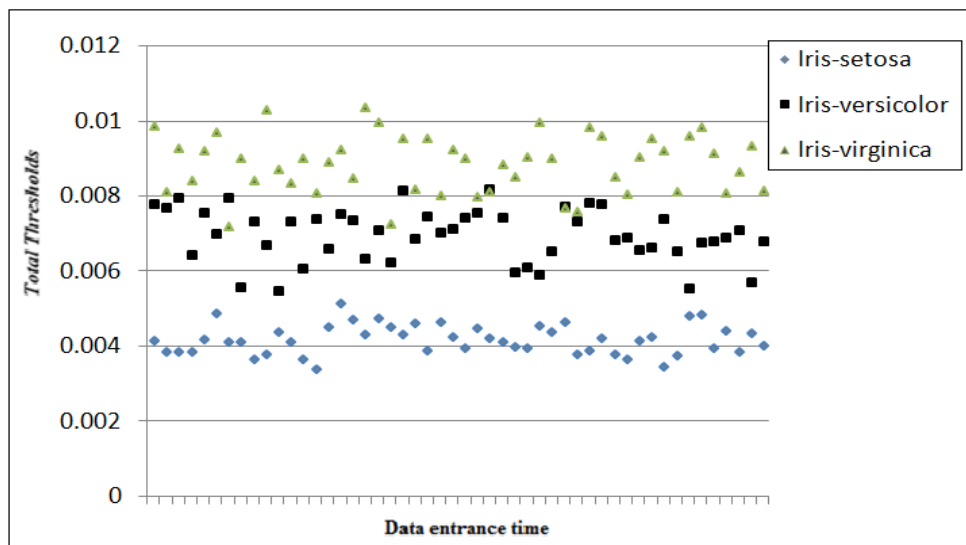


Fig.8: The outlook of clustering the Iris dataset by SFFNN.

After clustering, each cluster will be assigned to the special class which is most frequent in the cluster.

- Utilizing K -step activation function:** The K -step function [56] or threshold function is a linear activation function for transformation of input values. This kind of function as shown in Equation (8) is limited with K values based on the number of classes of the dataset, and each limited domain of thresholds refers to the special output value of the K -step function. For example, the *Binary-step* function is a branch of the K -step function for two data classes 0 and 1. It is often used in single

layer networks. The function $g(TT_i)$ is K -step activation function for the transformation of TT_i where the output will be 0 or 1 based on the threshold TT_i as shown in Fig. 9 and Equation (8).

$$g(TT_i) = \begin{cases} 1 & \text{If } (X \geq TT_i) \\ 0 & \text{If } (X < TT_i) \end{cases} \quad (8)$$

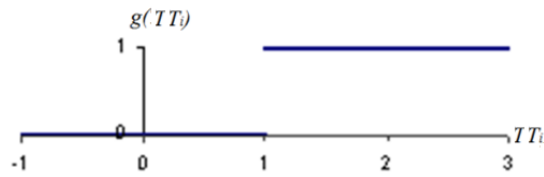
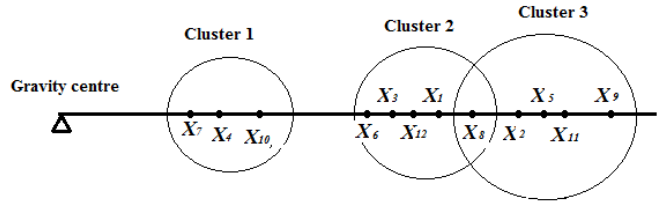


Fig.9: The *Binary-step* function.

- Semi-supervised clustering of the input data:** In this phase of the SFFNN clustering method, the method assigns the class label to each input data based on the training data set. Therefore, by using the K -step activation function, the model links the exclusive threshold of each input data to a related class. Consequently, by considering K class labels and their related exclusive thresholds in the training set, the proposed method expects K clusters and for each cluster considers a domain of thresholds. The model considers the clustering results of the last phase, if there is an input data with a related threshold in each cluster but without a class label (unknown data), the method moves this input data to a related cluster. In order to predict the class labels for unobserved data, there are several techniques. The authors often consider two unsupervised and supervised neural network models such as a combination of the SOM and back propagation network (BPN) [57] for the prediction of related class labels of the unobserved data [52]. The BPN uses gradient-based optimization methods in two basic steps: to calculate the gradient of the error function, and to employ the gradient. Usually, Bagging and Boosting methods are used in several models to find the upper vote or the weight of the mentioned class label [58]. We proposed a “Trial and Error” technique. The K -step function signs and predicts the class label of each unknown observation, the related cluster and thresholds domain of the cluster for the input instance. When the SFFNN clustering accuracy is measured by F -measure function with 10 folds of the test set, the accuracy shows the validation of the prediction. Hence, the method updates the number of clusters and the density of each cluster by using class labels through the feedback of users. This phase affects the result of clustering and improves the accuracy of clustering. Fig. 10 shows an example for clustering a dataset (part A) to two clusters, and improving the result by semi-supervised feed forward neural network clustering (part B).

A. Clustering by SFFNN method

Input data	Exclusive total threshold
X9	0.8014544
X11	0.6999856
X5	0.6499858
X2	0.5978536
X8	0.5198623
X1	0.4415853
X12	0.4157691
X3	0.3645489
X6	0.3085632
X10	0.1254876
X4	0.0648593
X7	0.0014578



B. Semi-clustering by SFFNN method

Input data	Exclusive total threshold	Class labels based on the training set
X9	0.8014544	3
X11	0.6999856	3
X5	0.6499858	3
X2	0.5978536	3
X8	0.5198623	2
X1	0.4415853	2
X12	0.4157691	2
X3	0.3645489	2
X6	0.3085632	2
X10	0.1254876	1
X4	0.0648593	1
X7	0.0014578	1

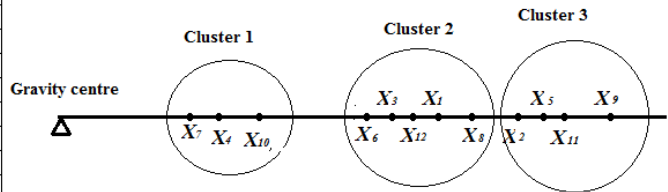


Fig.10: The outlook of improving the result of clustering by considering class label in the SFFNN.

2.4. The algorithm of the SFFNN clustering method

Fig. 11 shows the algorithm of the semi-supervised feed forward neural network (SFFNN) clustering method:

```

SFFNN ( )
Input : Dataset  $N$ ; Output: Clusters of dataset;
Begin
Let  $X$ :Data node set;
Let  $n$  : Number of nodes;
Let  $m$  : Number of attributes;
Let  $i$  : Current number of the node;
Let  $j$  : Current number of the attribute;
Let  $X_i$ : Current input data of dataset;
Let  $W_{ij}$  : The weight of attribute  $j$  of input data $X_i$ ;
Let  $BMW$ : Best match weight vector;
Let  $BMW_j$  :  $j_m$ Component of the BMW vector;
Let  $TT_i$ : Threshold of input data of  $X_i$ ;
{
Forall  $i=1$  to  $n$ 
{
Input a new vector  $X_i$ ;
//Data preprocessing phase:
 $X_{ij} = MinMax(X_{ij})$ ;
// Compute  $W_{ij}$  as the weight of  $X_{ij}$  the standard normal distribution of each input data value of  $X_i$  based on  $\mu_i$  and  $\sigma_i$  are mean and standard deviation of the input data
    
```

```

Forall  $j=1$  to  $m$ 
{

$$W_{ij} = (X_{ij} - \mu_i) / \sigma_i ;$$

}
}
// Generate the code book of real weights and the best match weight vector (BMW),
Forall  $j=1$  to  $m$ 

$$BMW_j = (\sum_{i=1}^n W_{ij}^2)^{1/2} ;$$

// The BMW includes  $BMW_j$ 

$$\overrightarrow{BMW} = (BMW_1, BMW_2, \dots, BMW_m) ;$$

// Process of a single layer unsupervised feed forward neural network for clustering of input data
// Compute the exclusive threshold of each input data of  $X_i$ 
Forall  $i=1$  to  $n$ 

$$TT_i = \sum_{j=1}^m X_{ij} \cdot BMW_j ;$$

// Semi-clustering by using K-Step activation function
{
Group the data points of input data with similar thresholds ( $TT_i$ ) in one cluster;
Learn and generate optimized number of clusters and their densities;
}
{
Assign the class label to each input data by using training set;
Predict the class label to unlabelled input data;
Update the number of clusters and density of each cluster
}
}

```

Fig.11: Semi-supervised feed forward neural network clustering algorithm.

The SFFNN method is able to update the clusters and their densities based on assigning a class label to each input data by utilizing the *K-step* activation function in just one iteration.

3.0 EXPERIMENTAL RESULTS AND COMPARISON

In this section, the performance of the SFFNN clustering is evaluated and compared with other related models. All the experiments were implemented in Visual C#.Net under Microsoft Windows 7 Professional operating system with 4 GHz Pentium processor. To evaluate the performance of the proposed method a series of experiments on several related methods and datasets were used.

3.1 Datasets from UCI Repository

The Breast Cancer-Wisconsin, Iris, Spambase, Arcene and Yeast datasets from the UCI Repository [59] are selected for evaluation of the proposed method as shown in the Table 6:

Table 6: The information of selected datasets in this study from the UCI Repository.

Dataset	Dataset Characteristics	Attribute Characteristics	Number of Data	Number of Attributes	Classes
Breast Cancer Wisconsin (Original)	Multi variable	Integer	699	10	Two classes: benign and malignant
Iris	Multi variable	Real	150	4	Three classes: Iris Setosa, Iris Versicolour and Iris Virginica
Spambase	Multi variable	Integer-Real	4601	57	Two classes: Spam and Non-Spam
Arcene	Multi variable	Real	900	10000	Two classes: Cancer patients and Healthy patients
Yeast	Multi variable	Real	8	1484	Ten classes

For experimentation, the speed of processing was measured by the number of epochs. The accuracy of the methods is measured through the number of clusters, and the quantity of *correctly classified nodes (CCN)* which shows the total nodes and density with the correct class in the correct related cluster in all clusters. The *CCN* is the same as the true positive and true negative nodes. Also the accuracy of the proposed method is measured by *F-measure* function with 10 folds of the test set. The precision of computing was considered with 15 decimal places, because of generating the different values of the exclusive thresholds.

3.1.1. Breast Cancer Wisconsin Dataset

Breast cancer Wisconsin (Original) dataset is selected from the UCI Repository. The collected dataset is from the university of Wisconsin Hospitals, Madison, and as reported by Dr. William H. Wolberg through his clinical cases [59, 60]. The SFFNN method was performed on the Breast Cancer Wisconsin dataset in one epoch in 8.7262 milliseconds. Table 7 shows the speed of clustering process based on the number of epochs and the accuracy based on the density of the *CCN* in the Breast Cancer Wisconsin dataset by the SFFNN model.

Table 7: The *Correctly Classified Nodes* and the number of epochs of clustering the Breast Cancer Wisconsin dataset by the SFFNN method versus other methods.

The clustering method	<i>CCN</i>	Density of <i>CCN</i> %	<i>F-measure</i> %	Epoch
<i>K-means</i>	657	96.19	-	20
Neural Gas	657	96.19	-	20
GNG	477	69.84	-	5
SOM	660	96.63	98.06	20
Semi-SOM	672	98.39	98.75	20
Clustering by SFFNN	661	96.78	98.74	1
Semi-clustering by SFFNN	683	100	100	1

In Table 7 based on the outcomes of the experiment, the *CCN* of the *K-means* and the NG methods are 657 after 20 epochs [61]. The *CCN* of the GNG method is 477 after 5 epochs [46]. Camastra and Verri reported, the SOM produced 660 *CCN* after 20 epochs [61]. The SOM clustering result was improved by considering class labels regarding the semi-SOM method, and the accuracy was changed to 672 *CCN*. The *CCN* of the proposed SFFNN clustering method after one epoch was 661, and the accuracy of the proposed SFFNN clustering was computed

by using the *F-measure* with 10 folds of the test set for this dataset which was 98.74% after just one epoch of training. The *CCN* of the semi-clustering by the proposed SFFNN method after one epoch is 683, and the accuracy by using the *F-measure* was 100% during just one epoch of training similar its density of the *CCN*. While for the BPN, the accuracy by *F-measure* is 99.28% after 1000 epochs of training.

3.1.2. Iris Dataset

The Iris plants dataset was created by Fisher [59, 62]. The Iris can be classified into Iris Setosa, Iris Versicolour and Iris Virginica. The SFFNN method clustered the Iris dataset in one epoch after 4.1744 milliseconds. Table 8 shows the speed of clustering based on the number of epochs and the accuracy based on the density of the *CCN* for the Iris dataset.

Table 8: The *Correctly Classified Nodes* and the number of epochs of clustering the Iris dataset by the SFFNN method versus other methods.

The clustering method	<i>CCN</i>	Density of <i>CCN</i> %	<i>F-measure</i> %	Epoch
<i>K-means</i>	134	89.33	-	20
NG	139	92.67	-	20
GNG	135	90.00	-	10
SOM	128	85.33	85.22	140
Semi-SOM	139	92.67	92.66	140
Clustering by SFFNN	142	94.67	94.67	1
Semi-clustering by SFFNN	150	100	100	1

In Table 8, the *CCN* of the *K-means* and the NG methods are 134 and 139 after 20 epochs respectively [61]. The *CCN* of the GNG method is 135 after 10 epochs [63]. The SOM produced 128 *CCN* and 85.22% accuracy by using the *F-measure* with 10 folds of test set after 140 epochs. The SOM clustering result was improved by considering class labels regarding the semi-SOM method, and the accuracy was improved to 139 *CCN*. The *CCN* of the proposed SFFNN clustering method after one epoch was 142, and the accuracy was computed by using the *F-measure* with 10 folds of test set which was 94.67% during just one epoch of training. The *CCN* of the proposed SFFNN clustering method after one epoch was 150. Also, the accuracy of the SFFNN clustering was computed by using the *F-measure* for this dataset which was 100% during just one epoch of training. While in the BPN, the accuracy by *F-measure* is 94% after 140 epochs of training.

3.1.3. Spambase Dataset

The Spambase E-mail dataset is created by Mark Hpkins, Erik Reeber, George Forman, Jaap Suermondt [59]. The Spambase dataset can be classified into Spam and Non-Spam. The SFFNN method clustered the Spambase dataset in one epoch taking 337.1057 milliseconds. Table 9 shows the speed of processing based on the number of epochs and the accuracy based on the density of the *CCN* in the Spambase dataset by the SFFNN method.

Table 9: The *Correctly Classified Nodes* and the number of epochs of the Spambase dataset by the SFFNN method versus other methods.

The clustering method	CCN	Density of CCN %	F-measure%	Epoch
K-Means	1083	23.54	-	20
Neural Gas	1050	22.82	-	20
GNG	967	21.02	-	5
SOM	2662	57.84	65.21	140
Semi-SOM	3528	76.67	80.74	140
Clustering by SFFNN	2731	59.36	66.46	1
Semi-clustering by SFFNN	4597	99.90%	99.89	1

In Table 9, based on the results of the experiment, the *Correctly Classified Nodes* of the K-means and the Neural Gas methods were 1083 and 1050 respectively after 20 epochs of training [46]. The *Correctly Classified Nodes* of the GNG method was 967 after 5 epochs [61]. The SOM produced 2662 CCN after 140 epochs. However, Camastra and Verri reported the accuracy of the SOM, was 1210 CCN after 20 epochs [61]. The SOM clustering result was improved by considering class labels based on the semi-SOM method, and the accuracy was improved to 3528 CCN. The CCN of the proposed SFFNN clustering after one epoch of training was 2731, while the *F-measure* with 10 folds of the test set for this dataset was 66.46%. The CCN of the semi-clustering by the SFFNN after one epoch was 4597 and its density of CCN was 99.90% and the accuracy of the SFFNN clustering was computed by using the *F-measure* which was 99.89% after just one epoch of training. While the BPN accuracy by *F-measure* was 79.50% after 2000 epochs of training.

3.1.4. Arcene Dataset

The Arcene dataset was collected from two different sources: the national cancer institute (NCI) and the eastern Virginia medical school (EVMS) [59]. All data were obtained by merging three mass-spectrometry datasets to create training and test data as a benchmark. The training and validation instances include patients with cancer (ovarian, prostate cancer), and healthy patients. Each dataset of training and validation contains 44 positive samples and 56 negative instances with 10,000 attributes. We considered the training data set and validation dataset with 200 total instances together as one set. The Arcene dataset can be classified into cancer patients and healthy patients. Arcene's task is to distinguish cancer versus normal patterns from mass-spectrometric data [59]. This dataset is one of 5 datasets of the NIPS 2003 feature selection challenge [64, 65]. The SFFNN method clustered the Arcene dataset in one epoch taking 5 seconds and 386.139 milliseconds. Table 10 shows the speed of processing based on the number of epochs and the accuracy based on the density of the CCN in the Arcene dataset by the SFFNN method.

Table 10: The *Correctly Classified Nodes* and the number of epochs of the Arcene dataset by the SFFNN method versus other methods.

The clustering method	CCN	Density of CCN %	F-measure%	Epoch
K-means	118	59.00	-	10
SOM	106	53.00	58.04	140
Semi-SOM	128	64.00	67.86	140
Clustering by SFFNN	120	60.00	64.26	1
Semi-clustering by SFFNN	200	100	100	1

In Table 10, based on the results of the experiment, the *CCN* of the *K*-means method was 118 after 10 epochs [66]. The SOM produced 106 *CCN* and 58.04% accuracy by using the *F-measure* with 10 folds of the test set for this dataset after 140 epochs. The SOM clustering result was improved by considering class labels based on the semi-SOM method, and the accuracy was improved to 128 *CCN*. The *CCN* of the proposed SFFNN clustering method after one epoch was 120, and the accuracy was computed by using the *F-measure* with 10 folds of the test set for this dataset which was 64.26% during just one epoch of training. The *CCN* of the semi-clustering by the SFFNN after one epoch was 200. The *F-measure* and the density of *CCN* of semi-clustering by SFFNN clustering were 100% after just one epoch of training. Recently, Veenu Mangat and Renu Vig [67] reported classification of the Arcene dataset by several classification methods such as *K-NN*. *K-Nearest Neighbour (K-NN)* is a supervised classifier that is able to learn by analogy and performs on n-dimensional numeric attributes [1]. Given an unknown instance, *K-NN* finds *K* instances in the training set that are closest to the given instance pattern and predicts one or average of class labels or credit-rates. Unlike BPN, *K-NN* assigns equal weights to the attributes. The *K-NN* (*K*=10) was able to classify the Arcene dataset with 77.00% accuracy by *F-measure* after several epochs and 10 times running the method.

3.1.5. Yeast Dataset

The Yeast dataset is obtained from the UCI Repository. The collected dataset is reported by Kentai Nakai from Institute of Molecular and Cellular Biology, university of Osaka [59]. The aim is to predict the cellular localization sites of proteins. The Yeast dataset contains 1484 samples with 8 attributes. The classes are Cytosolic, Nuclear, Mitochondrial, Membrane protein: no N-terminal signal, Membrane protein: uncleaved signal and Membrane protein: cleaved signal. Extracellular, Vacuolar, Peroxisomal and Endoplasmic reticulum lumen [59]. In this research, we cluster the Yeast data by using the SFFNN clustering method taking 684.833 milliseconds in one epoch of training. Table 11 shows the speed of processing based on the number of epochs and the accuracy based on the density of the *CCN* in the Yeast dataset by the SFFNN method.

Table 11: The *Correctly Classified Nodes* and the number of epochs of the Yeast dataset by the SFFNN method versus other methods..

The clustering method	<i>CCN</i>	Density of <i>CCN</i> %	<i>F-measure</i> %	Epoch
SOM	483	32.55	24.02	140
Semi-SOM	597	40.23	34.05	140
Clustering by SFFNN	494	33.29	27.43	1
Semi-clustering by SFFNN	1484	100	100	1

In Table 11 based on the results of the experiment, the SOM produced 483 *CCN* and 24.02% accuracy by using the *F-measure* with 10 folds of the test set for this dataset after 140 epochs. The SOM clustering result was improved by considering class labels regarding the semi-SOM method, and the accuracy was improved to 597 *CCN*. The *CCN* of the proposed SFFNN clustering method after one epoch was 494, and the accuracy was computed by using the *F-measure* with 10 folds of the test set for this dataset which was 27.43% during just one epoch of training. The *CCN* of the semi-clustering by the SFFNN after one epoch was 1484. The *F-measure* and the density of *CCN* of semi-clustering by SFFNN clustering were 100% after just one epoch of training. Several literature reported the difficulty of clustering or classification of the Yeast dataset. As, Longadge et al. [68] reported classification of the Yeast dataset by several classification methods such as *K-NN*. The *K-NN* (*K*=3) was able to classify the Yeast dataset with 0.11% accuracy by *F-measure* after several epochs and times running the method. Also, Ahirwar [69] reported the *K*-means was able to classify the Yeast dataset with 65.00% accuracy by *F-measure* after several epochs.

3.2. Breast cancer Dataset from the University of Malaya Medical Center (UMMC)

The dataset was collected by the University of Malaya Medical Center (UMMC), Kuala Lumpur from 1992 until 2002 [70]. As shown in Table 12, the dataset was divided into 9 subsets based on the interval of survival time: 1st year, 2nd year, ..., 9th year.

Table 12: The 9 subsets of observed data of the breast cancer from the UMMC based on the interval of survival time.

Year Of treatment	1 st year	2 nd year	3 rd year	n th year	8 th year	9 th year
1993	Data from 1993 to 1994	Data from 1993 to 1995	Data from 1993 to 1996	...	Data from 1993 to 2001	Data from 1993 to 2002
1994	Data from 1994 to 1995	Data from 1994 to 1996	Data from 1994 to 1997	...	Data from 1994 to 2002	
1995	Data from 1995 to 1996	Data from 1995 to 1997	Data from 1995 to 1998		
...			
2000	Data from 2000 to 2001	Data from 2000 to 2002				
2001	Data from 2001 to 2002					

As shown in Table 13, the breast cancer dataset contains 13 attributes. The number of input data instances in the dataset is 827, the number of Attributes is 13 continuous and one attribute for showing the binary class in two cases of alive or dead. This breast cancer dataset has class labels of '0' for alive and '1' for dead as constraints.

Table 13: The information of the UMMC breast cancer dataset attributes.

Attributes	Attribute Information
<i>AGE</i>	Patient's age in year at time first diagnosis
<i>RACE</i>	Ethnicity (Chinese, Malay, Indian and Others)
<i>STG</i>	Phase (how far the cancer has spread anatomically)
<i>T</i>	Tumour type (the extent of the primary tumour)
<i>N</i>	Lymph node type (amount of regional lymph node involvement)
<i>M</i>	Metastatic (presence or absence)
<i>LN</i>	Number of nodes involved
<i>ER</i>	Estrogen receptor (negative or positive)
<i>GD</i>	Tumour grade
<i>PT</i>	Primary treatment (type of surgery performed)
<i>AC</i>	Adjuvant Chemotherapy
<i>AR</i>	Adjuvant Radiotherapy
<i>AT</i>	Adjuvant Tamoxifen

Also, Fig. 12 shows the sample of breast cancer dataset from the UMMC:

AGE	RACE	STG	T	N	M	LN	ER	GD	PT	AC	AR	AT
66	I	1	1	0	0	0	X	X	M	N	Y	Y
66	I	2b	2	1	0	1	N	X	M	N	N	Y
43	I	2a	2	0	0	0	X	X	L	N	Y	Y
30	M	2b	2	1	0	2	X	X	L	Y	Y	N
65	C	4	2	x	1	X						
41	M	3a	3									
55	C	2b										
36	C											

Fig. 12: The sample of the breast cancer from the UMMC dataset.

We considered 9 subsets for 1st year, 2nd year, ..., 9th year. The SFFNN model was implemented on each dataset by considering the class labels. Table 14 shows the results of the implementation of the proposed model. The number of data of each subset; CPU Time usage per second for training each subset during one epoch; and the accuracy of the semi-clustering of each subset of the breast cancer dataset based on the *F-measure* with 10 folds of test data by using the SFFNN clustering method are shown.

Table 14: The results of implementation of the SFFNN for each subset of the breast cancer.

Year	CCN	Density of CCN (%)	The number of data in each subset	Epoch	CPU Time usage (Milliseconds)	<i>F-measure</i> Accuracy of the SFFNN (%)
1 st year	818	98.91	827	1	42.00	99.43
2 nd year	665	98.81	673	1	33.00	98.69
3 rd year	551	98.22	561	1	31.50	98.93
4 th year	428	97.27	440	1	30.50	98.14
5 th year	354	99.72	355	1	28.20	99.99
6 th year	270	100	270	1	14.90	100
7 th year	200	100	200	1	13.50	100
8 th year	124	100	124	1	12.70	100
9 th year	56	100	56	1	12.50	100

Table 14 shows that the training process for each sub-set of the breast cancer dataset took one epoch between [12.5, 42] milliseconds of CPU time; and the accuracies of the SFFNN for the Breast Cancer sub-datasets were between [98.14% - 100%]. For comparison with other similar methods in the scope of this research, we implemented the SOM using the BPN as a hybrid method. The SOM clustered each subset of breast cancer dataset and found the *BMW* vector of each data after 20 epochs. The BPN method fine-tuned the codebook of weights of unfolding SOM method instead of random weights. The training process in the BPN was 25 epochs. The results of the hybrid method of the SOM-BPN are shown in Table 15 for every subset.

The PCA [71] was considered as a preprocessing technique for dimension reduction and used by the BPN model. The PCA is a classical multivariate data analysis method that is useful in linear feature extraction and data compression. Table 13 shows the result of the PCA-BPN hybrid method for every subset of the breast cancer dataset of the UMMC. The PCA took the time of the CPU for dimension reduction and the BPN used the output of the PCA for classification after several epochs. The results of Table 15 shows the accuracies of implementation of the PCA-BPN method for the Breast Cancer dataset were between [63.00% - 99.00%], while that of the SOM-BPN method were between [71.00% - 99.00%].

Table 15: Comparing the accuracies of the hybrid methods of the PCA-BPN and the SOM-BPN with the SFFNN for each subset of the Breast Cancer dataset.

Year	PCA-BPN (%)	SOM-BPN (%)	SFFNN (%)
1st year	76.00	82.00	99.43
2nd year	63.00	72.00	98.69
3rd year	62.00	71.00	98.93
4th year	77.00	78.00	98.14
5th year	83.00	86.00	99.99
6th year	93.00	93.00	100
7th year	98.00	98.00	100
8th year	99.00	99.00	100
9th year	99.00	99.00	100

4.0 DISCUSSION

To evaluate the performance of the SFFNN clustering method, we compared the results of the proposed method with the results of other related methods on several data sets from UCI Repository and a real and original medical dataset from UMMC. Clustering the medical datasets is difficult because of limited observation, information, diagnosis and prognosis of the specialist; incomplete medical knowledge; and lack of enough time for diagnosis [37]. However, the developed SFFNN method has the capability to overcome some of the problems associated with clustering in the prediction of survival time of the breast cancer patients from the UMMC. As shown in the results obtained, the SFFNN method has superior results.

The SFFNN method has the successful actions and features:

- Training in one layer and just after one epoch resulting in fast training.
- Initialize a code book of real weights without the use of any random number or random parameter directly by learning through the input data values.
- Training of the SFFNN method without the need for relearning, updating weights or computation of an error function.
- The SFFNN method has two phases for semi-supervised feed-forward neural network clustering the input data. First, the proposed method predicts the number of clusters, the densities of the clusters, and subsequently clusters the dataset. Then, the method updates the clusters and their contents by using the class labels of the training set.

For computing times and memory complexities, we considered the parameters c , k , n , m , S_m as the number of epochs, clusters, nodes, attributes and size of each attribute. Table 16 shows the time and memory complexities of the SFFNN and some related methods. As well as that of the BPN as an example of the supervised feed forward neural network that depends mainly on the number of weighted functions in the hidden layers f_h and the number of iterations c , and K -NN that d is distance to one example.

Table 16: The time complexities and memory complexities of the SFFNN method and some related methods.

Method	Time Complexity	Memory Complexity
<i>K</i> -means	$O(c.k.n.m)$	$O((n+k).m.s_m)$
NG	$O(c.n^2.m)$	$O(c.n^2.m.s_m)$
GNG	$O(c.n^2.m)$	$O(c.n^2.m.s_m)$
SOM	$O(c.n.m^2)$	$O(c.n.m^2.s_m)$
BPN	$O(c.f_h)$	$O(c.f_h.s_m)$
<i>K</i> -NN	$O(c.k.n.d)$	$O(c.k.n.d.s_m)$
PCA	$O(m^2.n)+O(m^3)$	$O((m^2.n).s_m) + O((m^3).s_m)$
SFFNN	$O(n.m)$	$O(n.m.s_m)$

The SFFNN method is a linear SFFNN clustering method and has time complexity and memory complexity of $O(n.m)$ and $O(n.m.s_m)$.

5.0 Conclusion and Future work

We developed a single-layer semi-supervised feed forward neural network (SFFNN) clustering method to overcome the major problems of low speed and low accuracy of clustering and an effective memory complexity for only one epoch of training. The SFFNN can learn real weights and thresholds without using any random values and arbitrary parameters. Firstly, a code book of weights is trained by feeding input data directly to the network. Then, the *best match weight (BMW)* is mined from the codebook. Consequently, the exclusive threshold of each input data is computed. The input data are clustered based on their exclusive thresholds. The class label of each unlabelled input data will be predicted by considering a *K-step* activation function and the exclusive threshold. Finally the number of clusters and density of each cluster are updated. For experimental results, a series of experiments on several related methods and datasets were conducted. The SFFNN results were: 100% accuracy for the Breast Cancer, Iris, Arcene and Yeast, and 99.89% accuracy for the Spam dataset from the UCI Repository; and between [98.14% - 100%] accuracies for the breast cancer dataset from the University of Malaya Medical Center. The time and memory complexities of the SFFNN were $O(n.m)$ and $O(n.m.s_m)$ based on the number of nodes, attributes and size of the attribute. The experimental results show that the SFFNN clustering method demonstrates high speed and accuracy in performance with training in just one epoch and hence utilising low time and efficient memory complexity. For future work, an online dynamic semi-supervised feed-forward neural network clustering method is suggested by further developing the SFFNN model.

REFERENCES

- [1] Dasarthy, B.V., Nearest neighbor pattern classification techniques. 1990, IEEE Computer Society Press: Los Alamitos, CA.
- [2] Goebel, M. and Gruenwald, L., A survey of data mining and knowledge discovery software tools. ACM SIGKDD Explorations Newsletter, 1999. 1(1): p. 20-33.
- [3] Hegland, M., Data mining—challenges, models, methods and algorithms. Canberra, Australia: Australia National University, ANU Data Mining Group, 2003.
- [4] Kantardzic, M., Data mining: concepts, models, methods, and algorithms. 2011: New York: Wiley—Interscience.
- [5] Kemp, R.A., MacAulay, C., Garner, D. and Palcic, B., Detection of malignancy associated changes in cervical cell nuclei using feed-forward neural networks. Journal of the European Society for Analytical Cellular Pathology, 1997. 14(1): p. 31-40.

- [6] Andonie, R. and Kovalerchuk, B., *Neural Networks for Data Mining: Constrains and Open Problems*. Ellensburg, WA: Central Washington University, Computer Science Department., 2007.
- [7] Bose, N.K. and Liang, P., *Neural network fundamentals with Graphs, Algorithms, and Applications*. 1996: New York: McGraw–Hill.
- [8] McCloskey, S., *Neural Networks and Machine Learning*. 2000.
- [9] Bengio, Y., Buhmann, JM., Embrechts and M., Zurada, JM. , *Special Issue on Neural Networks for Data Mining and Knowledge Discovery*. 2000: Institute of Electrical and Electronics Engineers.
- [10] Han, J. and Kamber, M., *Data mining: concepts and techniques*. 2006: Morgan Kaufmann.
- [11] Kantardzic, M. and Press, I., *Data mining: concepts, models, methods, and algorithms*. 2003: Wiley-Interscience.
- [12] Hegland, M., *Data mining–challenges, models, methods and algorithms*. Publications of ANU Data Mining group, Draft, 2003.
- [13] Jain, A.K., *Data clustering: 50 years beyond K-means*. *Pattern Recognition Letters*, 2010. 31(8): p. 651-666.
- [14] Rougier, N. and Boniface, Y., *Dynamic self-organising map*. *Neurocomputing*, 2011.
- [15] Hebb, D.O., *The organization of behavior: A neuropsychological approach*. New York: Wiley, 1949. 1: p. 143-150.
- [16] Laskowski, K. and Touretzky, D., *Hebbian Learning, Principal Component Analysis, and Independent Component Analysis*. 15-486/782: *Artificial Neural Networks* 2006; Available from: <http://www.cs.cmu.edu/afs/cs/academic/class/15782-f06/slides/hebbpca.pdf>.
- [17] Haykin, S. and Network, N., *A comprehensive foundation*. *Neural Networks*, 2004. 2(2004).
- [18] Du, K.L., *Clustering: A neural network approach*. *Neural Networks*, 2010. 23(1): p. 89-107.
- [19] Fritzke, B., *Some competitive learning methods*. *Artificial Intelligence Institute, Dresden University of Technology*, 1997.
- [20] McClelland, J.L., Thomas, A.G., McCandliss, B.D. and Fiez, J.A., *Understanding failures of learning: Hebbian learning, competition for representational space, and some preliminary experimental data*. *Progress in brain research*, 1999. 121: p. 75-80.
- [21] Fritzke, B. A., *growing neural gas network learns topologies*. 1995: Citeseer.
- [22] Linde, Y., Buzo, A. and Gray, R., *An algorithm for vector quantizer design*. *IEEE Transactions on Communications*, 1980. 28(1): p. 84-95.
- [23] Kohonen, T., *Self-organizing maps (Springer Series in Information Sciences)*. Berlin: Springer–Verlag, 1997. 30: p. 22-25.
- [24] Martinetz, T.M., Berkovich, S.G. and Schulten, K.J., *Neural-gas' network for vector quantization and its application to time-series prediction*. *IEEE Transactions on Neural Networks*, 1993. 4(4): p. 558-569.

- [25] Asadi, R., Sabah Hasan, H. and Abdul Kareem, S., Review of current Online Dynamic Unsupervised Feed Forward Neural Network classification, in Computer Science and Electronics Engineering (CSEE) 2014: Kuala Lumpur, Malaysia.
- [26] Andrews, N.O. and Fox, E.A., Recent developments in document clustering. Computer Science, Virginia Tech, Tech. Rep, 2007.
- [27] Mercer, D.P., Clustering large datasets. Linacre College, 2003.
- [28] Hamker, F.H., Life-long learning Cell Structures--continuously learning without catastrophic interference. Neural Networks, 2001. 14(4-5): p. 551-573.
- [29] Furoo, S., Ogura, T. and Hasegawa, O., An enhanced self-organizing incremental neural network for online unsupervised learning. Neural Networks, 2007. 20(8): p. 893-903.
- [30] Hebboul, A., Hacini, M. and Hachouf, F., An incremental parallel neural network for unsupervised classification. 2011: IEEE.
- [31] Germano, T., Self Organizing Maps. Accessed at <http://davis.wpi.edu/~matt/courses/soms>, 1999.
- [32] Honkela, T., Description of Kohonen's Self-Organizing Map. Accessed at <http://www.cis.hut.fi/~tho/thesis>, 1998.
- [33]
- [34] Kohonen, T., Self-Organization Maps (3rd ed.). 2000, Berlin: Springer-Verlag.
- [35] Ultsch, A. and Siemon, H.P., Kohonen's Self Organizing Feature Maps for Exploratory Data Analysis. Proc. Int. Neural Networks Conf., 1990: p. 305-308.
- [36] Hebboul, A., Hacini, M. and Hachouf, F., An incremental parallel neural network for unsupervised classification. in Proc. 7th Int. Workshop on Systems, Signal Processing Systems and Their Applications (WOSSPA). 2011: Tipaza, Algeria, May 9-11.
- [37] Rougier, N. and Boniface, Y., Dynamic self-organising map. Neurocomputing, 2011. 74(11): p. 1840-1847.
- [38] Melek, W.W. and Sadeghian, A., A theoretic framework for intelligent expert systems in medical encounter evaluation. Expert Systems, 2009. 26(1): p. 82-99.
- [39] Han, J. and Kamber, M., Pei. Data Mining Concepts and Techniques. 2011, The Morgan Kaufmann Series in Data Management Systems, Morgan Kaufmann Publishers.
- [40] Demuth, H., Beale, M. and Hagan, M., Neural Network Toolbox TM 6: User's Guide. Natick, MA: Math Works, 2008.
- [41] Jolliffe, I.T., Principal component analysis. 2002: Springer verlag.
- [42] Kasabov, N., ECOS: Evolving connectionist systems and the ECO learning paradigm. 1998: Citeseer.
- [43] Han, J. and Kamber, M., Data Mining, Southeast Asia Edition: Concepts and Techniques. 2006, San Francisco, CA.: Morgan kaufmann.
- [44] Pavel, B., Survey of clustering data mining techniques. San Jose, CA: Accrue Software, 2002.
- [45] Deng, D. and Kasabov, N., On-line pattern analysis by evolving self-organizing maps. Neurocomputing, 2003. 51: p. 87-103.

- [46] Hinton, G.E. and Salakhutdinov, R.R., Reducing the dimensionality of data with neural networks. *Science*, 2006. 313(5786): p. 504.
- [47] Bouchachia, A., Gabrys, B. and Sahel, Z., Overview of some incremental learning algorithms. in *Proc. Fuzzy Systems Conf. Fuzz-IEEE*. 2007: IEEE.
- [48] Prudent, Y. and Ennaji, A., An incremental growing neural gas learns topologies. in *Proc. IEEE Int. Joint Conf. Neural Networks, IJCNN'05*. 2005.
- [49] Kamiya, Y., Ishii, T., Furao, S. and Hasegawa, O., An online semi-supervised clustering algorithm based on a self-organizing incremental neural network. in *Neural Networks, 2007. IJCNN 2007. International Joint Conference on*. 2007: IEEE.
- [50] Shen, F., Yu, H., Sakurai, K. and Hasegawa, O., An incremental online semi-supervised active learning algorithm based on self-organizing incremental neural network. *Neural Computing and Applications*, 2011. 20(7): p. 1061-1074.
- [51] Herrmann, L. and A. Ultsch, Label propagation for semi-supervised learning in self-organizing maps. *Proceedings of the 6th WSOM*, 2007.
- [52] Asadi, R. and Abdul Kareem, S., Review of feed forward neural network classification preprocessing techniques. in *Proc. 3rd Int. Conf. Mathematical Sciences (ICMS3)*, pp. 567–573, Kuala Lumpur, Malaysia. 2014: AIP Publishing.
- [53] Larochelle, H., Mandel, M., Pascanu, R. and Bengio, Y., Learning algorithms for the classification restricted Boltzmann machine. *The Journal of Machine Learning Research*, 2012. 13: p. 643-669.
- [54] Oh, M. and Park, H.M., Preprocessing of independent vector analysis using feed-forward network for robust speech recognition. in *Proc. Neural Information Processing Conf.* 2011: Springer.
- [55] Ziegel, E.R., *Statistical inference*. *Technometrics*, 2002. 44(4).
- [56] Serway, R.A., *Physics for scientists and engineers*. 2012: Cengage Learning.
- [57] Alippi, C., Piuri, V. and Sami, M., Sensitivity to errors in artificial neural networks: A behavioral approach. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 1995. 42(6): p. 358-361.
- [58] Werbos, P., *Beyond regression: new tools for prediction and analysis in the behavioral sciences*. PhD Thesis. Harvard University, 1974.
- [59] Daffertshofer, A., Lamoth, C.J.C., Meijer, O.G. and Beek, P.J., PCA in studying coordination and variability: a tutorial. *Clinical Biomechanics*, 2004. 19(4): p. 415-428.
- [60] Asuncion, A. and Newman, D., *UCI Machine Learning Repository*. Irvine, CA: University of California, School of Information and Computer Science. Accessed at <http://www.ics.uci.edu/~mlern/MLRepository>, 2007.
- [61] Wolberg, W.H. and Mangasarian, O.L., Multisurface method of pattern separation for medical diagnosis applied to breast cytology. *Proceedings of the National Academy of Sciences*, 1990. 87(23): p. 9193-9196.
- [62] Camastra, F. and Verri, A., A novel kernel method for clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005. 27(5): p. 801-805.

- [63] Fisher, R., *The Use of Multiple Measurements in Taxonomic Problems: Contributions to Mathematical Statistics*. 1950, JNew York: Wiley.
- [64] Costa, J.A.F. and Oliveira, R.S., Cluster analysis using growing neural gas and graph partitioning. in *Proc. Int. Joint Conf. Neural Networks*. 2007: IEEE.
- [65] Guyon, I., Design of experiments of the NIPS 2003 variable selection benchmark. in *NIPS 2003 workshop on feature extraction and feature selection*. 2003.
- [66] Guyon, I. and Elisseeff, A., An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 2003. 3: p. 1157-1182.
- [67] Davidson, I. and Yin, K., Semi-lazy learning: combining clustering and classifiers to build more accurate models. in *International Conference on Machine Learning*. 2003.
- [68] Mangat, V. and Vig, R., Novel associative classifier based on dynamic adaptive PSO: Application to determining candidates for thoracic surgery. *Expert Systems with Applications*, 2014. 41(18): p. 8234-8244.
- [69] Longadge, M.R., Dongre, M.S.S. and Malik, L., Multi-Cluster Based Approach for skewed Data in Data Mining. *Journal of Computer Engineering (IOSR-JCE)*, 2013. 12(6): p. 66-73.
- [70] Ahirwar, G., A Novel K means Clustering Algorithm for Large Datasets Based on Divide and Conquer Technique. Pradnyesh. J. Bhisikar(IJCSIT) *International Journal of Computer Science and Information Technologies*, 2014. 5(1): p. 301-305.
- [71] Hazlina, H., Abdul Kareem, S., NurAishah, M. and Yip, C., Back propagation neural network for the prognosis of breast cancer: comparison on different training algorithms. in *Proc. 2nd. Int. Conf. Artificial Intelligence in Engineering & Technology (ICAIET)*. 2004.
- [72] Jolliffe, I.T., *Principal Component Analysis*, in *Springer Series in Statistics* 1986, Springer New York. p. 1-7.