Applying of (SOM, HAC, and RBF) algorithms for data aggregation in wireless sensors networks

Ahmed Subhi Abdalkafor[1], Salah A. Aliesawi[2]
[1]Career Development Center, University of Anbar, Anbar, Iraq
[2]College of Computer Science and Information Technology, University of Anbar, Anbar, Iraq

Article Info	ABSTRACT
	Wireless sensor network (WSN) is one of the most promising technologies due to its size, cost-effective nature, and its ability to easily deploy in the target environment, as well as for its entry into many sensitive applications. However, making the most of the potential of this network is very difficult due to many issues, including the data received from the sensor nodes contains a huge amount of data redundant that negatively affects the overall network performance. Recent years have witnessed an increasing interest in data aggregation technology intending to eliminate redundant data from neighboring sensor nodes before transferring to the base station, thus improve performance efficiency and increasing the wireless sensor networks lifespan. This paper focused on applying three intelligent algorithms (SOM, HAC, and RBF) and describing the impact of data aggregation strategy on WSNs through the results obtained. As well as, an accurate description of the literature that applied these algorithms. A Competitive classification accuracy has been achieved when the proposed work is implemented and tested via the intel berkeley research lab dataset.

Corresponding Author:

Ahmed Subhi Abdalkafor
Career Development Center, University of Anbar, Anbar, Iraq
Email: ahmed.abdalkafor@uoanbar.edu.iq

1. INTRODUCTION

The new technological development has allowed more sensors to be associated with the internet of things (IoT). The number of devices connected to the IoT will exceed 20 billion by 2021, and the data generated from these devices will reach more than 2.5 quintillion bytes per day [1]-[3]. Moreover, people's desire to sense and collect data has increased exceedingly due to the effect of artificial intelligence (AI) technology on their ability to use data. Consequently, many sensor devices are more deployed in several applications for it sensing and obtaining data [4], [5]. WSN was originally designed to have features of low data rate escorted with low complexity and energy consumption. These features enable WSN to be deployed at a large-scale as well as within a low implementation cost [6]. Wireless Sensor Networks can be defined as a system of nodes that collaboratively sense, monitor, capture, process, and control other input-output (in form of data or signals) of other systems or support dealings between computational systems, people and the surrounding neighbor. WSNs present an influential integration of disseminated processes of sensing, communication, and computation [7]-[9]. There are many conceivable WSN applications almost to every sector. Basically, boundless from environmental monitoring and management, tracking and positioning, medical and health care services, and military, industrial and transportation applications [10]-[13]. In our day, modern technologies have worked efficiently and at an accelerated pace to decrease the weight and size of the sensor nodes, where they can be characterized by high sensing capabilities, processing, and wireless communication, likewise improving the accuracy of sense. Due to a high sensing node density spread across

the target environment, there is great redundancy between physical phenomena data measurements in this environment. Therefore, during various data packets forwarded to the sink, data aggregation is applied to multiple packets, so the volume of data after aggregation is definitely much less than the total size of the original data, so this technique aims to decrease redundant data transmissions to the upper layer, thus enhancing the overall performance of WSNs. Several studies have been proposed that dealt with the use of the three algorithms in the wireless network environment, which are as: Ullah and Youn [14] have suggested a model of data aggregation for reducing network degradation and energy depletion. After the data is eliminated the redundant and outliers, the self-organized map algorithm is used to re-cluster these data before forwarding them to the base station. Mittal and Kumar [15] have clustered sensed data rather than the entire aggregated it at the cluster head node. The proposed method uses three algorithms, and the implementation of cluster node values using the self-organizing map was better than the other two algorithms. Lung and Zhou [16] proposed the distributed hierarchical agglomerative clustering (HAC) algorithm, which groups similar sensor nodes together and creates clusters before selecting the CH with the aiming of providing efficient clusters without the need for global network knowledge. Also, Raghunandan *et al.* [17] applied HAC algorithm to reduce some network issues and select (CH) efficiently to prolong the WSNs lifespan. Khorasani and Naji [18] used four algorithms including radial basis data aggregation (RBDA) with aiming of aggregating data, eliminate redundancy, as well as increase network accuracy and its energy. Also, Ullah and Youn [19] proposed a data aggregation system to reduce excessive and erroneous sensed data by relying on data clusters and the radial basis function has been applied in cluster heads to decrease the instability of the training process. The main contributions of this paper are stated as:

− Apply (self-organization map (SOM), HAC, and radial basis function (RBF)) algorithms in WSNs environment using intel berkeley research lab dataset to enhance the WSNs performance.
− Suggesting more than one method in the pre-processing stage led to obtaining high results in terms of clustering using the SOM and HAC algorithms.
− The modification in the output hidden layer structure of the RBF algorithm yielded very high classification results (extended RBF algorithm).
− Describe the impact of data aggregation strategy on WSNs with comprehensive and accurate literature that applied these algorithms.

2. RESEARCH METHOD

Our proposed model consists of three main phases: data collection, pre-processing and analysis, and implementation of algorithms. Figure 1 displays our proposed model block diagram.
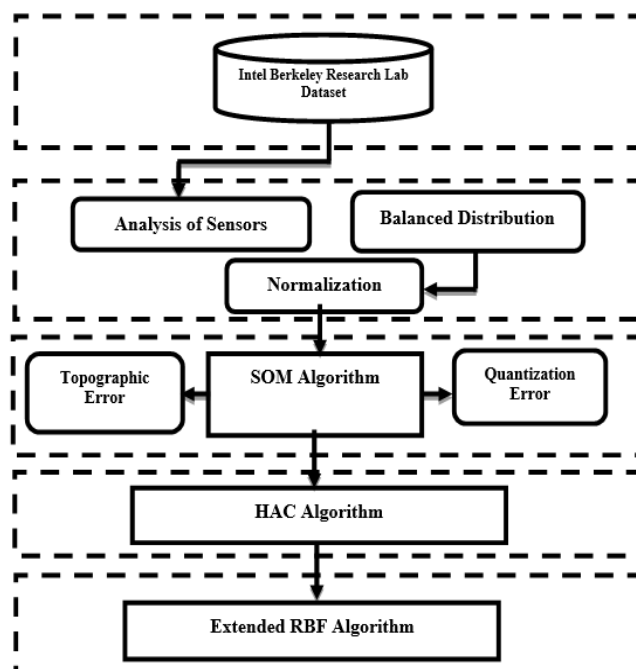

Figure 1. The block diagram of our proposed model

2.1. Data collection

Intel berkeley research lab dataset is an open-source dataset collected at the intel berkeley research lab. Capturing period extends from February 28th to April 5th, 2004. As shown in Figure 2, intel lab dataset composed of eight features as observational variables, whereas the captured sensed data items) observational instances reaches up to 2,313,682 instances. the dataset was collected using 54 Mica 2 Dot sensors along with weather boards deployed in the field of intel berkeley research lab. these boards measures temperature, humidity, and light weather attributes every 31 seconds [20].
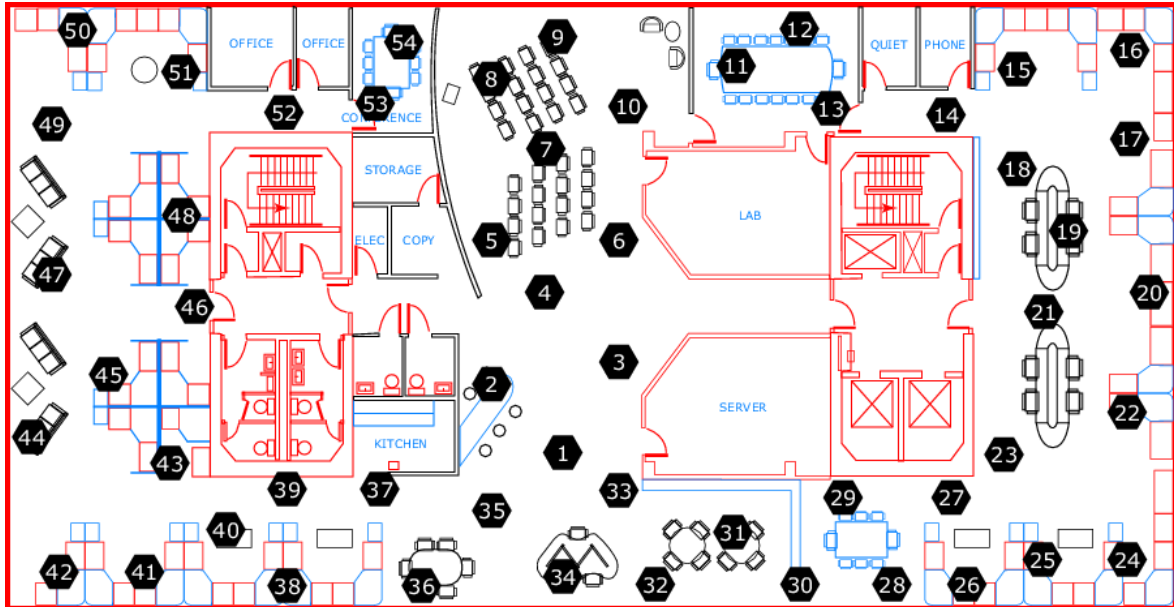

Figure 2. Intel berkeley research lab dataset

2.2. Pre-processing and analysis

To know the exact details and the total repeated measurements of each sensor, the use of each sensor to measure temperature, humidity, and light were measured to pave the way for the next steps and select the appropriate algorithms and functions that ultimately lead to the success of the data aggregation strategy and thus reduce the excess data as mentioned above. Figure 3 shows the frequency of measurement of the three phenomena for 54 sensors (Motid).
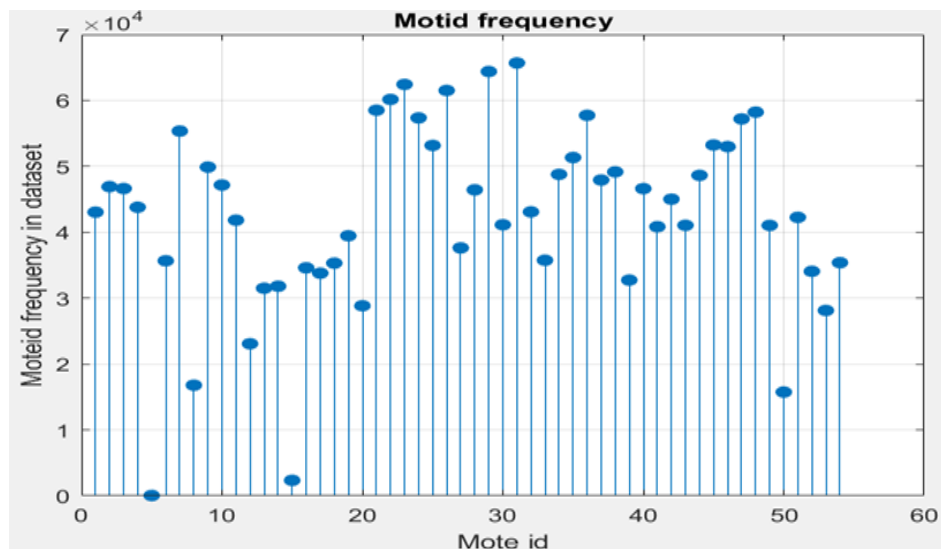

Figure 3. Frequency measurement of 54 sensors

2.2.1. Balanced distribution

Figure 3 shows the unevenness of the sensor readings so that the measurement effort is very high on one sensor and the other is less than that, which will negatively affect the performance of the algorithms used in this paper. Because it will tend to the sensor that uses the most data and thus obtain inaccurately and few results. Therefore, a function was proposed to randomly redistribute the sensor measurements, while keeping the measurement values without modification. Figure 4(a) shows the original dataset, while Figure 4(b) shows the balanced distribution method.

2.2.2. Normalization

This step is necessary for the features of data that have a wide range of values, which forces a bias of the network weight towards higher values. The numerical content of the measured dataset was scaled to a new scale in this step, allowing it to be easily fed into the network as input. In Figure 4(c), the minimum and maximum normalization with values of 0.0,+0.5 was used in intel berkeley research lab dataset as given by (1).

$$\mathfrak{D}' = \frac{\mathfrak{D} - Min\,\mathfrak{D}}{Max\,\mathfrak{D} - Min\,\mathfrak{D}}(NewMax\,\mathfrak{D} - NewMin\,\mathfrak{D}) + NewMin\,\mathfrak{D} \tag{1}$$

	1	2	3
1	122.1530	22.1719	1.2586e+03
2	19.3220	44.5482	64.4000
3	122.1530	-3.9190	104.8800
4	21.8798	42.1830	9.2000
5	122.1530	-3.9190	13.8000

(a)

	1	2	3
1	122.1530	-3.9190	11.0400
2	19.9884	37.0933	45.0800
3	19.3024	38.4629	45.0800
4	19.1652	38.8039	45.0800
5	19.1750	38.8379	45.0800

(b)

	1	2	3
1	0.1893	0.4937	0.3406
2	0.0681	0.4949	0.0174
3	0.1893	0.4922	0.0284
4	0.0711	0.4948	0.0025
5	0.1893	0.4922	0.0037

(c)

Figure 4. Measurement of, (a) original dataset, (b) balanced distribution, and (c) normalization

2.3. Implementation of algorithms

In this paper, the dataset that has been collected in the data collection stage is unlabeled inherently. In the context of machine learning and artificial intelligence, one approach is utilizing the unsupervised learning schemes to label a given dataset. Supervised training of RBF network requires labeled dataset, however, due to the lack of labeled training dataset two powerful unsupervised learning algorithms are used: SOM network and HAC algorithm are used to this task.

2.3.1. Self-organization map algorithm

SOM is one of the neurobiologically-inspired neural networks and one of most important unsupervised learning algorithms. The most basic form of it is that proposed by the Finish researcher, Teuvo Kohonen in 1982 [21], [22]. Two prevalent use-cases for the unsupervised learning are the dimensionality reduction and the exploratory analysis. The most common and essential tasks of the explanatory analysis is the data clustering and analysis, where the system learns the actual trend (or structure) of the data without the need to explicit providing of the labels, which is the main task of utilizing SOM network in our work.

In our proposed model, the main goal of utilizing SOM network is two-fold: first is to infer the natural structure exists in the intel lab dataset and secondly is to provide μ-parameters to RBF network through applying another unsupervised clustering. As a starting point, in our case, we use SOM network for providing an initial explanatory analysis of the sensed dataset that collected in the data-collection stage.

2.3.1.1. Applying self-organization map in WSNs

SOM network comprises of two layers, at the bottom layer lies the input features of data instance, where each item is connected in parallel to all neurons that comprises the second layer which arranged in two-dimensional lattice as illustrated in Figure 5.

Each neuron in the grid (as so-called codebook) in this lattice is a vector of weights (code vector) as same gradients length as that of the input training data items. The core idea of SOM is that each presentation of input vector adjusts the weight vectors of the winning node as well as the topological neighborhood till reaching a close resembling for the input vector.
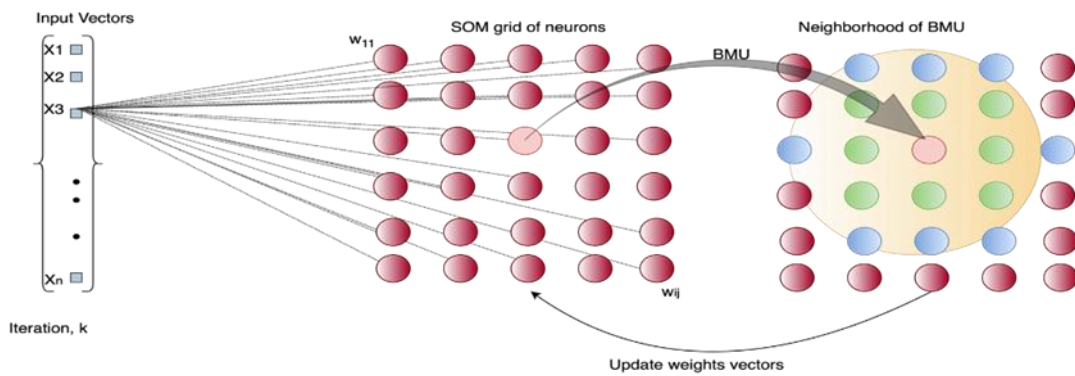
Figure 5. Self-organizing map learning process

Let's denote input training data subset taken from the pre-processed intel lab dataset, as $x \epsilon \mathbb{R}^3$ (temperature, humidity, and lights features) associating a weight vector $w_k \epsilon \mathbb{R}^3$, $k = 1,2,\cdots,K$. Each data presentation referred as $x(t)$ where t represents the current presentation and range from $t = 1$ to $t = P$ where P represents the total number of data instances in the training subset. Among many variants of SOM, in this work, we implemented a variant of SOM called " batch learning-SOM (BL-SOM) "[23]. In the on-line SOM, weights vectors of SOM codebook w_k are updated after each input presentation $x(t)$, whereas in BL-SOM, this updating occurs at the end of each epoch (one epoch is a single pass over the entire dataset). For each input vector $x(t)$ is presented, the Euclidean distance between $x(t)$ and each weight vector on the grid w_k is computed as in (2).

$$\mathfrak{D}_k(t) = \|x(t) - w_k(t)\|^2 \tag{2}$$

Afterwards, as shown Figure 5, the best match unit (BMU or winner neuron), is determined as in (3)

$$\mathfrak{D}_{k,BMU}(t) \triangleq min(\mathfrak{D}_k(t)) \tag{3}$$

as mentioned before, in batch SOM, weight updating takes place at the end of each epoch, thus, let's define, t_o and t_f define the start and the finish of each epoch, then, weight updating is given as in (4).

$$w_k(t_f) = \frac{\sum_{tf}^{to} h_{bmu,k}(t)x(t)}{\sum_{tf}^{to} h_{bmu,k}(t)} \tag{4}$$

where; $w_k(t_f)$: weight vector of kth SOM neuron computed at the end of Epoch t_f .

$x(t)$: t^{th} input training vector.

$h_{bmu,k}(t)$: neighborhood function, controls the zone to which w_k is able to adjust responding to an input vector $x(t)$ that is most closely resembling w_{bmu}. In our work, we used the standard Gaussian function as illustrated in (5).

$$h_{k,bmu}(t) = exp(-\|r_k - r_{bmu}\|/\sigma(t)^2) \tag{5}$$

where; r_k : coordinates of kth SOM neuron

r_{bmu} : coordinates BMU neuron in response to t^{th} input vector

$\sigma(t)$: width of neighborhood function, decreases with respect to t from a pre-specified value to a final value equal the width of one neuron

2.3.2. Hierarchical agglomerative clustering algorithm

HAC is a mathematically clustering algorithm and widely employed in modern machine learning and data science and demand to partition the database into a set of clusters and produce a hierarchical relationship within the data representations themselves [24]. Generally, hierarchical algorithms are categorized as agglomerative or divisive. The first category starts with one object, and then the closest pairs

of clusters are merged (agglomerates) successively with every iteration until all clusters are combined into one large cluster containing all pairs (bottom-up approach) while the second category called divisive that reverse approach to agglomerative clustering starting with one cluster of data and then splits the appropriate cluster [25]. HAC algorithm produces a tree-like structure called a dendrogram, designed to provide multiple, high-level parts of a database. This type of clustering help create a small cluster that provides an informative data presentation that begins from a non-redundant data set to the complete data set belonging to a single set.

2.3.2.1. Applying HAC algorithm in WSNs
Though powerful, major problems with SOM network that it is often intractable, is to attain a perfect or close-to-perfect mapping in such a way that SOM neurons are unique representations to the input data. Moreover, in this proposed model, to achieve high accurate data aggregation, label precision is an issue to be considered.

In order to address both issues, and for better and more stable clustering performance in the WSN operation stage, we propose a novel solution by first using preprocessed intel lab dataset to generate the trained SOM grid map, then, we apply the hierarchical agglomerative clustering [26] on SOM's neurons per se. Figure 6 illustrates the general core idea behind HAC algorithm.
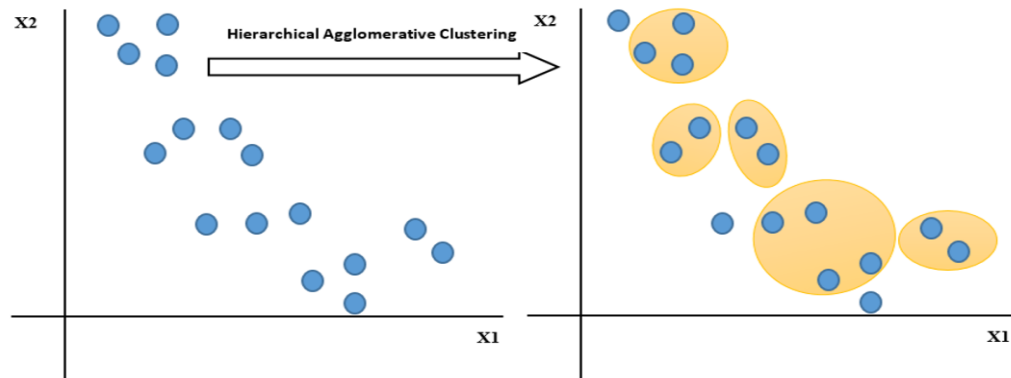

Figure 6. Concept of hierarchical agglomerative clustering algorithm

In the first step, each weight vector on the SOM grid defines its own cluster. in our case, SOM grid has K neurons, therefore, we have K initial clusters. In a series of the subsequent steps, the algorithm of a hierarchical agglomeration always recognizes a pair of clusters that have the smallest value of mutual distance using the distance (linkage) metric $\mathfrak{D}_{average}$ described in (6).

$$\mathfrak{D}_{average}(cluster^A, cluster^B) = \frac{1}{n_A n_B}\left(\sum_{i=1}^{n_A}\sum_{j=1}^{n_B} dist(x_{Ai}, x_{Bj})\right) \qquad (6)$$

where; n_A : number of objects in $cluster^A$
$\quad n_B$: number of objects in cluster B
$\quad dist$: Euclidean distance between $cluster^A$ and cluster B
$\quad x_{Ai}$: is the ith object in $cluster^A$
$\quad x_{Bj}$: is the jth object in $cluster^B$

2.3.3. Radial basis function algorithm
At the royal signals and radar establishment, radial basis function networks were first born and formulated in a paper published by Broomhead and Lowe [27], however, this type of neural networks was popularized by the researchers Moody and Darken [28]. In a concise definition, radial-basis function neural network is a special type of feedforward network architecture, composed from three layers only: input, hidden, and output layer. This type of neural networks can provide a local representation of a N-dimensional input space by using localized overlapping zones of distance-based functions called: network receptive fields. The core idea of radial basis function neural network resides in the theory of radial basis functions which, in turn, a pure mathematical discipline known as the approximation theory. Figure 7 shows the regular radial basis function algorithm.
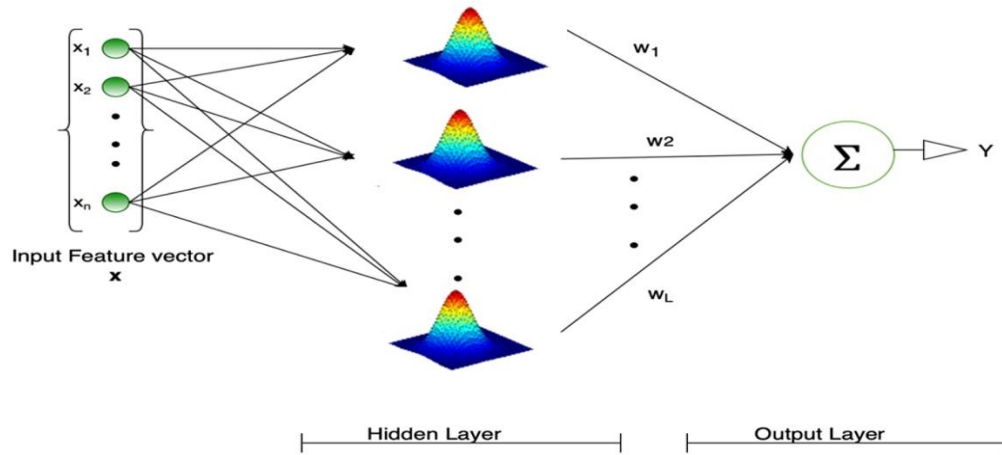
Figure 7. Regular radial basis function algorithm structure with Gaussian activation functions

2.3.3.1. Applying RBF algorithm in WSNs

Typically, in terms of data aggregation, RBF is used to classify (interpolate) the most appropriate representative value for a set of sensed data items. Mathematically, consider a multivariate function $(f : \mathbb{R}^d \rightarrow \mathbb{R}^m)$, and without loss of generality, in our case, feature vectors composed of three features (temperature, humidity and light), i.e., $\mathbb{R}^d, d = 3$, and we expect to get a scalar value as the output of the RBF, i.e., $\mathbb{R}^m, m = 1$. However, in our proposed data aggregator function, RBF network classification model has to carry out a mapping from the continuous input space of sensed data $X \in \mathbb{R}^3$ into a finite set of classes $Y = \{1,2,3,\cdots,\mathcal{L}\}$, where, $\mathcal{L}$ is the quantization levels, which, in turn, represents the classes (labels) of training data. Thus, we can formulate training data as in (7).

$$\mathcal{D}_{tr}^t = \{(x^t, y^t)| \ x^t \in \mathbb{R}^3, y^t \in Y, \qquad t = 1,\cdots,T\} \tag{7}$$

In the recall phase (testing phase) further unlabeled observational sensed data $\mathcal{D}_{tst}^s = \{(x^s)| \ x^s \in \mathbb{R}^3, s = 1,\cdots,\acute{S}\}$ is presented to trained RBF classification model, where RBF network estimate their class membership $y_{tst}^s \in Y$ accordingly. In our RBF model, we used the extended version of RBF network as elaborated in Figure 8.
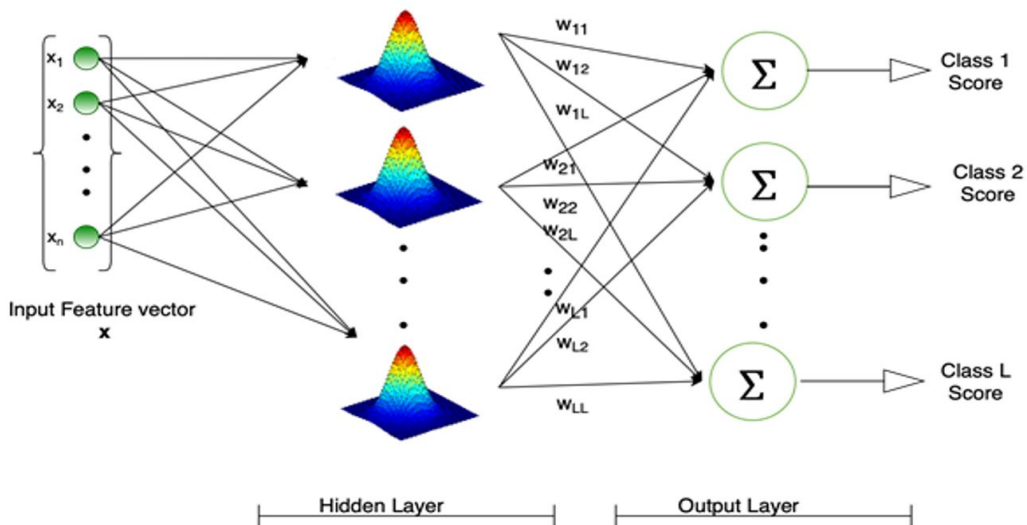

Figure 8. Extended radial basis function network structure with Gaussian activation functions

Where the output layer composed of a set of weighted summation nodes, one per each class. for each input feature vector, there is a corresponding weighted sum of the value of activation function from every RBF neuron.

Each output node computes the "*score*" that corresponding to the category that represented. the classification decision is made by taking the maximum of output scores, and the output neuron with maximum value is the one that represent the category corresponding to the input feature vector. Based on the 1-of-L encoding version of RBF network as shown in Figure 7, where we have K basis functions that performing a mapping (approximation) $F: \mathbb{R}^3 \to \mathbb{R}^L$ as in (8).

$$F_l = \sum_j^K w_{jl}\,\varphi(\|x - \mu_j\|) + w_l^o \qquad l = 1, \cdots, L \quad \mu_j = \mu_1, \cdots, \mu_K \qquad (8)$$

where, $\|\cdot\|$ represents the Euclidean distance norm, w_{jl} are real numbers used as weights associated with the activation functions $\varphi(\cdot)$. w_l^o refers to the biases which can absorb in the summation by implicate an extra basis function has a constant activation output set always to 1, i.e., $\varphi_o = 1$. Now, the class of output is determined as in (9).

$$class(x) = argmax(F_l(x)) \quad l = 1, \cdots, L \qquad (9)$$

The mapping problem represented by (8) and (9) can be formulated in a matrix form as in (10).

$$\phi W = Y \qquad (10)$$

where, $W = (w_1, w_2, \cdots w_L)$ each component is a vector of weights, $Y = (y^1, y^2, \cdots, y^L)$ and ϕ is an $L \times L$ matrix defined as in (11).

$$\phi = (\varphi^1, \varphi^2, \cdots, \varphi^L) \qquad (11)$$

where, $\varphi^j = \varphi(\|x - \mu_j\|)$, $\mu_j = \mu_1, \cdots, \mu_K$ is a real valued function depends on the " radial " norm distance between the input vector x and the origin or a certain point, in our case, this point is the corresponding peak center $\mu_j \epsilon \mathbb{R}^3$ of the Gaussian function. Conventionally, the activation functions of the radial basis function network are chosen as Gaussian functions, which is well-known, and we adapted in our work, and it is given in (12).

$$\varphi(\|x - \mu_j\|) = exp(-\|x - \mu_j\|^2/2\sigma^2) \qquad (12)$$

where, σ is the standard deviation peak center and it specifies the width of the Gaussian function. The solution to (10) (which represents the optimal weights), is given by (13).

$$W = \phi^{-1}Y \qquad (13)$$

However, including the bias function in ϕ matrix, make it a non-square matrix, and therefore, we can't obtain the inverse ϕ^{-1}. Thus, to remedy this problem, a solution from linear algebra is used, which called the pseudo-inverse instead of inverse as in (14).

$$W = (\phi^T \phi^{-1})^{-1} \phi^T Y \qquad (14)$$

where, ϕ^T represents the transpose of ϕ matrix.

3.   RESULTS AND DISCUSSION

In this paper, the quantization error and the topographic error were measured respectively to prove the quality of SOM algorithm.

3.1. Quantization error

The average distance between a data point and its assigned node is measured by quantization error; the smaller the value, the better the fit [29]. The final quantization error after applying the SOM algorithm equals 0.005 as illustrated in Figure 9.
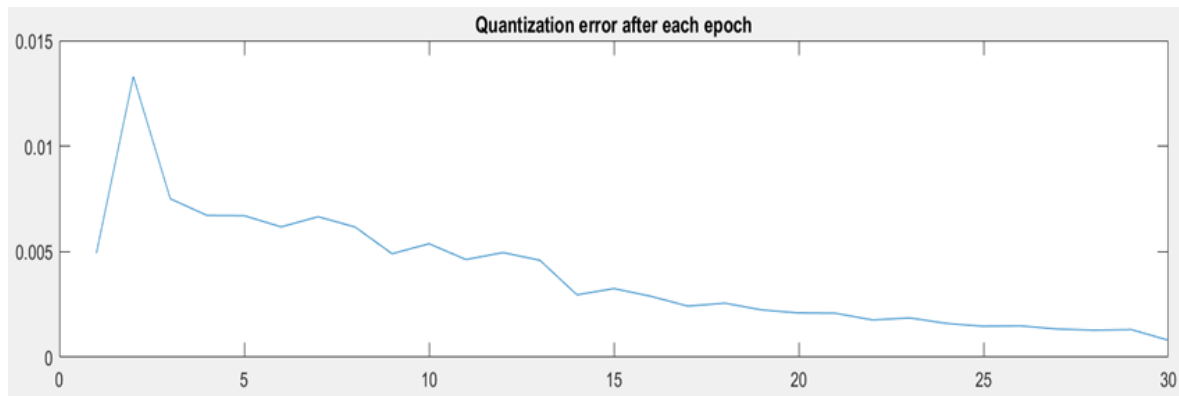
Figure 9. Quantization error

3.2. Topographic error

It's a metric for the quality of the spatial structure used in the map's design. For each input, the calculation is finished by determining the positions of the best matching neuron and the second best match in the map. The structure of this entry has been preserved if the nodes are next to each other. If it isn't, this is an error. As a result, the errors total number divided by the total number of data points equals the topographic error of a map [29]. The final topographic error after applying the dataset was 0.105.

In this paper, the RBF algorithm was used as the classification engine of our proposed model. Before the training stage, we used (84000 samples) as X-train for measuring temperature, humidity, and light and 84000 samples as Y-train for labeling to measure the accuracy training. Also, 6000 samples as X-test and 6000 samples as Y-train for labeling were used for measuring testing accuracy. The optimal weights of the extended RBF algorithm have been obtained after the training stage performed and then fed into the next stage. Our proposed model has achieved overall classification accuracy reached 97.54% and 97.70% for training and testing accuracy respectively. The high results obtained from the training and testing step prove the efficiency of modification in the output of the hidden layer of the RBF algorithm. This high result gave us the green light to propose a new protocol that uses the remaining samples from the intel berkeley research lab dataset to aggregate the data with the aiming of improving WSN performance in terms of energy, accuracy, and latency.

4. CONCLUSION

Data aggregation is the most powerful concern in WSNs that must be considered in all aspects of these networks. A comprehensive description of three algorithms, namely (SOM, HAC, RBF) for WSNs was presented in this paper, starting from their structure, fundamental components, and arithmetic, ending with applying these algorithms to data aggregation for WSNs. The results obtained prove that the proposed work can give effective performance to WSNs after modifying the output of the hidden layer of the RBF algorithm. This paper can be a guide for finding appropriate solutions to overcome or reduce data aggregation issues. In future work, we can propose a new protocol that implements the three algorithms with the help of a data aggregation strategy for obtaining the best performance of WSNs in terms of energy, accuracy, and latency.

REFERENCES

[1] A. Li, W. Liu, L. Zeng, C. Fa and Y. Tan, "An Efficient Data Aggregation Scheme Based on Differentiated Threshold Configuring Joint Optimal Relay Selection in WSNs," *IEEE Access*, vol. 9, pp. 19254-19269, 2021, doi: 10.1109/ACCESS.2021.3054630.

[2] R. A. Rahman, U. R. Hashim, and S. Ahmad, "IoT based temperature and humidity monitoring framework," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 1, pp. 229-237, 2020, doi: 10.11591/eei.v9i1.1557.

[3] V. H. Céspedes, G. Y. Florez, and Y. A. G. Gómez, "The internet of things in high andean wetland monitoring, historical review approach," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 3, pp. 1572-1579, 2021, doi: 10.11591/eei.v10i3.2653.

[4] A. S. Abdalkafor and S. Aliesawi, "The Impact of Data Aggregation Strategy on a Performance of Wireless Sensor Networks (WSNs)," *the 14th International Conference on Developments in eSystems Engineering (DeSE)*, 2021.

[5] A. Abdalkafor and S. Aliesawi, "Data Aggregation Techniques in Wireless Sensors Networks (WSNs): Taxonomy and An Accurate literature Survey,", in The First Virtual International Conference on Sciences (VICS2021), 2021.

[6] C. H. Lo, N. Ansari, M. S. Obaidat, A. Anpalagan, and I. Woungang, "IEEE 802.15. 4 based wireless sensor network design for smart grid communications," in *Handbook on Green Information and Communications Systems*, Academic, 2013, pp. 91–114.

[7] A. M. E. Tamizharazi, J. J. M. E. Selvathai, A. K. Priya, R. Maarlin, and M. Harinetha, "Energy aware heuristic approach for

cluster head selection in wireless sensor networks," *Bulletin of Electrical Engineering and Informatics*, vol. 6, no. 1, pp. 70-75, 2017, doi: 10.11591/eei.v6i1.598.

[8]   A. Sharmin, F. Anwar, and S. M. A. Motakabber, "A novel bio-inspired routing algorithm based on ACO for WSNs," *Bulletin of Electrical Engineering and Informatics*, vol. 8, no. 2, pp. 718-726, 2019, doi: 10.11591/eei.v8i2.1492.

[9]   G. I. Hapsari, G. A. Mutiara, L. Rohendi, and A. Mulia, "Wireless sensor network for monitoring irrigation using XBee Pro S2C," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 4, pp. 1345-1356, 2020, doi: 10.11591/eei.v9i4.1994.

[10]   D. Kandris, C. Nakas, D. Vomvas, and G. Koulouras, "Applications of wireless sensor networks: an up-to-date survey," *ASI*, vol. 3, no. 1, pp. 1-73, 2020, doi: 10.3390/asi3010014.

[11]   A. S. Abdalkafor and S. Aliesawi, "Efficient Data Aggregation Strategy in Wireless Sensor Networks: Challenges and Significant Applications," *International Conference on Computing and Communication Networks (ICCCN-2021)*, 2021.

[12]   K. W. A. Ani, A. S. Abdalkafor, and A. M. Nassar, "Smart City Applications: A Survey," *Proceedings of the 9th International Conference on Information Systems and Technologies*, 2019, no. 36, pp. 1-4, doi: 10.1145/3361570.3361616.

[13]   K. W. A. Ani, A. S. Abdalkafor, and A. Nassar, "An overview of wireless sensor network and its applications," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 17, no. 3, pp. 1480-1486, 2020, doi: 10.11591/ijeecs.v17.i3.pp1480-1486.

[14]   I. Ullah and H. Y. Youn, "A novel data aggregation scheme based on self-organized map for WSN," *The Journal Supercomput*, vol. 75, no. 7, pp. 3975-3996, 2019.

[15]   M. Mittal and K. Kumar, "Data clustering in wireless sensor network implemented on self organization feature map (SOFM) neural network," *International Conference on Computing, Communication and Automation (ICCCA)*, 2016, pp. 202-207, doi: 10.1109/CCAA.2016.7813718.

[16]   C.-H. Lung and C. Zhou, "Using hierarchical agglomerative clustering in wireless sensor networks: An energy-efficient and flexible approach," *Ad Hoc Networks*, vol. 8, no. 3, pp. 328-344, 2010, doi: 10.1016/j.adhoc.2009.09.004.

[17]   G. H. Raghunandan, A. S. Rani, S. Y. Nanditha and G. Swathi, "Hierarchical agglomerative clustering based routing algorithm for overall efficiency of wireless sensor network," *International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, 2017, pp. 1290-1293, doi: 10.1109/ICICICT1.2017.8342755.

[18]   F. Khorasani and H. R. Naji, "Energy efficient data aggregation in wireless sensor networks using neural networks," *International Journal of Sensor Networks*, vol. 24, no. 1, pp. 26–42, 2017.

[19]   I. Ullah and H. Y. Youn, "Efficient data aggregation with node clustering and extreme learning machine for WSN," *Journal of Supercomputing*, vol. 76, no. 12, pp. 10009-10035, 2020.

[20]   Intel Lab Data, [Online] Available: "http://db.csail.mit.edu/labdata/labdata.html."

[21]   T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biol. Cybern.*, vol. 43, no. 1, pp. 59–69, 1982.

[22]   T. Kohonen, "Self-organizing maps: ophmization approaches," in *Artificial neural networks*, Elsevier, 1991, pp. 981–990.

[23]   R. D. Lawrence, G. S. Almasi, and H. E. Rushmeier, "A scalable parallel algorithm for self-organizing maps with applications to sparse data mining problems," *Data Min. Knowl. Discov.*, vol. 3, no. 2, pp. 171–195, 1999.

[24]   Smarika, N. Mattas, P. Kalra and D. Mehrotra, "Agglomerative hierarchical Clustering technique for partitioning patent dataset," *4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions)*, 2015, pp. 1-4, doi: 10.1109/ICRITO.2015.7359281.

[25]   A. Chhabra, V. Vashishth and P. Mohapatra, "Fair algorithms for hierarchical agglomerative clustering," *arXiv Prepr*, arXiv2005.03197, 2020.

[26]   M. Kubat, "An introduction to machine learning". *Springer*, 2017.

[27]   D. S. Broomhead and D. Lowe, "Multivariable functional interpolation and adaptive networks, complex systems, vol. 2," 1988.

[28]   J. Moody and C. Darken, *Learning with localized receptive fields*. Yale Univ., Department of Computer Science, 1988.

[29]   E. A. Uriarte and F. D. Martín, "Topology preservation in SOM," *International Journal of Applied Mathematics and computer science*, vol. 1, no. 1, pp. 19-22, 2005.

BIOGRAPHIES OF AUTHORS

Ahmed Subhi Abdalkafor ⓘ 🇬 SC Ⓟ was born in Anbar-Iraq in 1988. He received his MSc. degree from Computer Science Department in Middle East University, Jordan in 2016. The B.Sc. from Anbar University, Iraq. Currently, he is a Ph.D. student. His research interests are: wireless sensor networks, IoT, image processing, pattern classification, neural network, and cloud computing. He can be contacted at email: ahmed_abdalkafor@yahoo.com.

Salah A. Aliesawi ⓘ 🇬 SC Ⓟ received the Ph.D. degree from the University of Newcastle, UK in 2012. His research interests include wireless networks, IoT, digital signal processing and multiuser detection. Currently, he is working at University of Anbar. He can be contacted at email: salah_eng1996@uoanbar.edu.iq.