# Accurate Data Aggregation Created by Neural Network and Data Classification Processed Through Machine Learning in Wireless Sensor Networks

**Sudha C** ( csudhahyd@gmail.com )
Annamalai University
**Suresh D**
Annamalai University
**Nagesh A**
Mahatma Gandhi Institute of Technology

**Research Article**

**Keywords:** Sensor network, Data Collection, Machine learning Noise, Errors Validation, Aggregate Data

**Posted Date:** September 17th, 2021

**DOI:** https://doi.org/10.21203/rs.3.rs-895195/v1

# Accurate Data Aggregation Created by Neural Network and Data Classification Processed Through Machine Learning in Wireless Sensor Networks

C. Sudha[1], D.Suresh[2], A. Nagesh[3]
[1]Research Scholar, Annamalai University, Chidambaram, Tamilnadu, India.
[2]Assistant Professor, Dept. of IT, Annamalai University, Chidambaram, Tamilnadu, India
[3]Professor, Dept. of CSE, Mahatma Gandhi Institute of Technology, Hyderabad, India.
[1]csudhahyd@gmail.com, [2]deiveekasuresh1@gmail.com, [3]akknagesh@rediffmail.com

## Abstract

Data aggregation is the process of efficiently collecting and transferring sensing data to a destination; it aids in the methodical and cautious utilization of sensor network resources. In the same way, when opening aggregated data that is free of noise and errors, the accuracy and efficiency of the data received improve. For this to happen, the data must flow to the destination in the most precise manner and in the shortest possible time by understanding the area's circumstances and addressing the immediate demands of the site where the sensors are positioned. This necessitates precise algorithms that eliminate errors and noise in the sense of data before aggregating it. The network used the ACNM technique – accurate data aggregation created by neural networks and data classification processed through machine learning in wireless sensor networks. We employ machine learning as part of artificial intelligence that learns from data. Following that, it can detect errors and noise in that Data in stages, reduce them, process them in a neural network model, and finally aggregate them to deliver the most accurate data to the destination. When data was provided using this ACNM protocol, the results showed that it arrived at its destination with the least delay.

Keywords: Sensor network; Data Collection; Machine learning Noise, Errors Validation; Aggregate Data

## 1. Introduction

The method of collecting and aggregating usable Data is data aggregation. Data aggregation is recognized as one of the simple processing procedures for saving resources. Data aggregation is an important way for WSN to conserve precious resources. In an energy-efficient strategy, the main goal of data aggregation algorithms is to collect and syndicate data to extend the network's lifetime. Wireless sensor networks comprising many sensor nodes are now extensively used for a few days in different applications. [1] Unlike wired networks, the contact varies between every sensor node, and the lifespan of the network system is typically restricted here. To address these limitations, the sensor nodes are gathered to accumulate the data before sending it to the BS (Base station). In resource-constricted wireless sensor networks (WSN), energy efficiency is an important metric. Multiple methods, such as task rotation, energy-optimized scheduling, energy-conscious routing, and data aggregation, can be used to minimize energy usage across the network. Data aggregation techniques have issues like delay, traffic load, redundancy, and accuracy [2]. The goal of the

current research is to revise the abilities of NN (Neural Networks).To demonstrate this capability, data aggregation of WSN and the provision of an algorithm were established on those sensor networks [3]. These procedures can be realistic in a way that helps one to reflect on their surroundings and shortcomings. Regarding data gathering, the objective is to provide an algorithm that diminishes the rate of transmission, energy consumption and increases the durability of the sensor network where possible. To accomplish this goal, sufficient Data is aggregated, current nodes in the setting are clustered, and proper CHs are picked using neural network nodes. Nodes are capable of obtaining and developing data like neurons in wireless sensor networks. Links between nodes function similarly to neuron synapses for the transmission of signals. Data-aggregation in wireless sensor networks and the algorithm of a neural network also have common characteristics. These, namely simple results, represent a wide variety of data through processing as well as computation. Neural networks are therefore used to understand data aggregation in wireless sensor networks. A critical problem in sensor networks is how to gather classified information in a more energy-saving operation—centered on neural network methods used to minimize energy consumption and data aggregation algorithms.

The rest of the paperwork is structured in the following way: Sect. 2 addresses the many data aggregation systems used for WSN. Then, it introduces the proposed Technique ACNM in Sect. 3. The simulation outcomes are discussed in section 4 and Section 5, the conclusion part.

## 2. Related work

In [4] BPNDA, a back-propagation network-based data aggregation system, was suggested. The neural network in the three-layer BP was used. The input layer neurons are grouped into cluster members even as the hidden layer neurons and the output layer are. Neurons are clustered at the cluster's head. Just those extracted Data reflecting the characteristics of the raw data will be transmitted to the drain, so the data collection efficiency has increased and reduced energy consumption. In [5] EEDC, it is conscious of the spatiotemporal connection between the Data for sensing. The spatial similarity is exploited by dynamically grouping Dissimilarity-based sensor nodes in clusters Measure data from sampling. Temporal correlation is used to represent the sampling information by the piecewise linear approximation method. Minimize the transmission of information under a given boundary and reduce the precision of the approximation. Back-propagation Network Scheme It was projected for the WSN data-aggregation in [6]. This uses multilayer information to minimize data dimensions while refining the data aggregation practice effectively. The spatiotemporal relation between knowledge is exploited by EAST [7] and is ahead of the necessary Data for the BS along a short distance path. Based on spatial similarity, it clusters the participant nodes.

In contrast, the illustrative nodes use the time suppression strategy, and the WSN Traffic movement is reduced by projecting future data from preceding data logs. In [8], the authors merged a reinforcement algorithm through a data fusion mechanism to construct an efficient system and suggested a data fusion learning method based on a kernel. Methods of

monitoring in an energy-efficient context. It was addressed in [9] and, depending on the situation, the surveillance procedure was modified. A framework application built on context-aware that energetically learns the connection between numerous context features has been proposed [10]. CARLOG [11] implements a novel rule-based approach to minimizing the cost of bandwidth (BW) and energy utilization, and it accepts a large number of concurrent context attribute queries. The ARIMA method for forecasting data from previous readings was proposed in [12]. Zhao et al. [13] are preoccupied with reducing overall data collection time. Different protocol regions have numerous mobile hoarders and multiple-access space divisions. The device increases the time between other regions or regions for data collection. Data collection is completed in clusters when the energy convertible modes of the sensor nodes are used. Low-Latency Data Collection Spread [14] Few reviews have addressed this issue, and most previous aggregation techniques rely on fixed structures that greatly restrict the misuse of neighborhood dynamic schedule vacancies. The aggregation and Circuits, Systems, and Signal Processing free schedules are generated at the equivalent period by using difficult periods for the entire neighborhood in two distributed aggregation calculations. Resilient data aggregation The Distributed data integration model [15], a spatial-transient relation-based method, is displayed for remote sensor systems. The calculation joins the center to separate, based on the required data convergence model, and the weighted computation will increase the data recovery combination accuracy. However, it can enhance the power of noise interference. The newly established metaheuristic optimization algorithm, the cuckoo search algorithm, is used to solve problems with optimization. This is the Enhanced cuckoo search algorithm suggested by Valian et al. [16] to develop junctions, speed, and accuracy. However, after the beginning of the Cuckoo, search parameters are also maintained. Therefore, there could be a chance of a decline. In [17], the ELM neural network can effectively accumulate the data at the cluster head before being transmitted to the sink to minimize network traffic and energy consumption. To clean the data at the respective sensor node, the KF is applied, and the cosine comparison is considered. The comparison of data with density is at the heart of cluster sensor nodes.

## 3. ACNM Design and Implementation

### 3.1 Network Formation

In the ACNM proposal, the sensor network is developed, and the path selection is built according to a clustering system, before data transmission, then sensors sensed Data is learned through a machine learning technique$M_L$ By the nodes, as shown in Figure.1, after that from the sensed data noise, errors are removed, and then it is processed into a neural network to get precise aggregation data outputs.

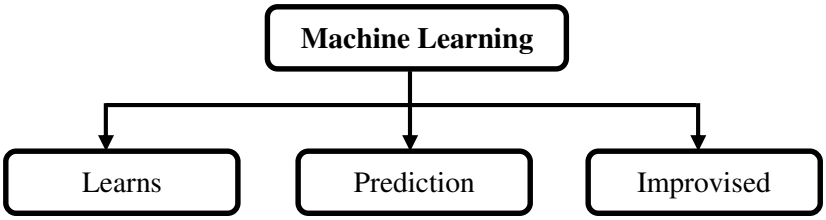


Figure. 1. Process of Machine Learning

The topography contains a set of parameters and sensors to form the wireless network. The network nodes are built with location-aware global positioning system GPS awareness to get the location knowledge. The sensors can be aware of the present location through GPS and share their locations during the communication period with other nodes. Here network topology has the bound for sensor network as $G = (S, N)$, at this point, $S$ defines the set of sensors ( $s_1 \ldots\ldots s_n$) and $N$ defines the set of neighbor's sensors ( $n_1 \ldots. n_n$) in a network. Similarly, there are few other orderings defined $G$ as per the ACNM design: known as source, cluster head, $C_H$ and a destination named base station $B_s$ Is placed on the network and shared their data. From the set of $S$, we signified the sensors $n_i$ as a sensor, and $n_j$ as the neighbor of the sensor, these $n_i$ and $n_j$ both form the network connectivity around the network.

### 3.2 Initiate Data Collection

During the data transmission, the sensors and cluster heads $C_H$ have identified the path to send the data packets to the destination $B_s$. The sensed data are collected and filtrate $FC_H$by cluster head so that it applying the filtering techniques to reduce the noise rank of sensed data packets, the noise level can rise or decline on data packets depends on the sensing environmental changes, this noise can degrade the signal strength between the sensor and the $C_H$.

Also, we need to calculate the number of data packets that each sensor senses as a quantity of the sensing data $S_D$. Similarly, calculate the previous sensed Data $S_D$ and store it in a sensing table $S_T$ along with the random numbers from 0 to 1 for each packet. Later, discover the differences between the random numbers $R_1R_2$ order and then find the differences between the minimum value $M_N$ and the maximum value $M_A$ of the $S_D$ and store it in each sensor $S_T$ as a trained description of collected data packets $D$ as shown in Figure.2


Figure.2 Boundary of Data.

Subsequently, these values are computed by a linear active $L_A$ state method to classify the structure of the data packets from the beginning, and it will recalculate in all phases of the network process to update the $L_A$, before that update a few necessary parameters to get more precision values of $L_A$ as follows: here $D_S$ defines the probability of the sensing value,

$$D_S = \frac{(R_1 - R_2)}{(M_A - M_N)}, \text{ Then}$$

$$D = D_S(S_D - M_N) + M_A$$

### 3.3 Sensing Packet Predictions

At this juncture, future data arrival activity predictions $P_S$ are made based on the data that the sensors sensed from the environment. This $P_S$ allows us to understand the future data arrival quantity as per the current situation of the channel state and network environment. Based on this prediction, $C_H$ Can regulate the receiving packets from its member nodes to minimize the loss; according to the predictions, sensors minimize and maximize the data transmissions to the cluster head. Furthermore, they made accurate interactions between them and adjusted the transmissions based on the prediction estimations.

$$P_S = S_D D + N_G P_S$$

### 3.4 Channel Gain Predictions

After the data arrival predictions, depends on the packet entrance total receiving signal strength $T_{RSS}$ was computed as $T_{XP} + \eta - P_L$, to know the power used during data transfer through the channel as a radio signal, here $\eta$ is the random executions of power standard deviation, also $P_L$ is known as path loss factor, then the signal-noise ratio $S_{INR}$ is computed as follows $S_{INR} = (T_{RSS} - N_{0i} - N_{0E})$. Here $N_{0i}$ is the noise generated internally by the sensor and $N_{0E}$ is the noise generated according to the environmental situation. Subsequently, noise-based channel gain $N_G$ is calculated, this calculation allows the sensor to detect the amount of noise $N_O$ during sensing, and it needs to regulate the transmission depends on the $N_O$ and $T_{RSS}$. The noise-

based channel gain $N_G$ between sensors ( $n_i$ , $n_j$) is defined by $Gain = N_G\, n_i + N_O$ where $n_i$ is the packet's sender, $Gain$ are the packets received by $n_j$.

$$N_G = (Gain + (M_V\text{-}S_D))$$

Here noise generates unwanted errors in packets. This error deviation decreases the performance of the sensor sensing capability and reduces the packet transmission between source to destination. Here, $M_V$ is the mean value of data packets computed from the data sensed by sensors. It is also known as average data sensed packet counts. It is calculated by the division number of data packets with currently sensed packet counts $S_D$. First, we need to calculate the total number $T_N$ of sensed data packets up to a specific period. Then divide the current number of sensed data packets $S_D$ by $T_N$ and get the mean value $M_V$ of the sensed Data over time from a different context.

$$T_N = \sum S_D$$

$$M_V = \frac{T_N}{S_D}$$

Later, calculate the difference between those mean values as $D_{MV} = (M_V - S_D)$, as per these differences are further distinguished as the amount of squared distance of each sensed data from the $M_V$, from this, we subtract the sensed data packets. Afterward, update the total number of those difference values $TD_{MV}$ as $\sum D_{MV}$. Then calculate an error variance $V$ between $TD_{MV}$ and the number of data packets currently perceived, such as now $V$ is computed as below:

$$V = \frac{TD_{MV}}{S_D}$$

Then it's essential to calculate and update the standard deviations $\eta$ between those differences to know the various deviations of the consecutive sensing packets as $\sqrt{V}$, it is the square root of the variance, and it defines the measurements stated of just how many sensors are different from the frequent $M_V$ in a cluster, then compute the $L_A$ as follows:

$$L_A = F\,(S_D - I_P M_M) + N_G, \text{ Here, } S_D = (N_O + I_P)\, M_M$$

### 3.5 Packets Boundary Validation

During transmission, dynamic packet classifications are updated frequently, also$B_V$ They are considered as the border value of the lower and upper bound of the average data packets. $I_P$ $d$defines the input of the matrix. Below we have calculated the boundary limits of the packets along with the channel gain and forecasts the receipt of the packets as $BP_S$.

$$BP_S = DBP_S + I_P B_V + N_C$$

Noise errors might contain quantity errors then packet sampling-based errors; the $V$ between the $S_D$ We have observed their actual beginning values. Initial error $V$ is estimated by computing the average boundary values of the $S_D$. And the current error variance is updated using the summation value of the nearest point with the matrix multiplication of A and error variance with the product of transpose A matrix. Once the system state matrices are formed, the predicted estimation for the current measurement is computed using the product and summation of state matrices. And the received $S_D$ is updated as a summation of measurement noise with the product of observation and border predicted estimation values $BP_S$.

Sensor network errors are unavoidable, generating according to the sensing environment, transmission state, channel condition, and hardware errors. So, we calculated the error differences $E_D$ about the present state packet arrival based on the packets continuously coming from the beginning of the network and the packets receiving in the present state. When calculating this, we need to calculate the error differences along with $S_D$ noise level based updated channel gain $N_C$ , $L_A$ and D as given below.

$$E_D = DL_A + S_D N_C$$

**3.6 Training Data and Filtering Noise**

Here, we need to recommend a node with a minor noise channel to act as the head among all neighbors for data transmission; the channel noise is detected and removed that channel-connected node from head election based on the past interactions between nodes. Therefore, the input data $I_p$ for the linear filtering $FC_H$ $the$ system will contain all historical Data of the interactions between the nodes as training data. This Data is saved with node ID in one column, and its historical transaction details are held at each row of the array. This $FC_H$ $the$ system helps to classify the noise existing in the $S_D$ and then we evaluate and update the matrix with renewed data set. The matrix is formed by taking the sensed Data as input $I_p$ and the following formula is computed to get the noise removal linear filtering system as below, and the data covariance is calculated and represented as a new error

difference $NE_D$ Matrix. Once the updated new error difference is computed, then the relationship is formed to generalize the update cycle $FC_H$ . In the update cycle, filter value is computed using the product and transpose of $T_G$ with an inverse product of the summation of $NE_D$:

$$T_G = (S_D - NE_D)^{-1}$$

Also, the error variance of the current sensed Data is computed multiple times as several iterations to get perfect accuracy. Table.1 shows the testing, training accuracy, and error variations based on data changes as per the input. Here D1, D2, D3, D4, D5 indicates the Data runs at the various interval and that data packets passed into the network for processing and obtained the following results.

Table.1 Processed Values

| Inputs | Hidden neurons | Accuracy in Training | | Accuracy in Testing | | Testing Errors | |
|---|---|---|---|---|---|---|---|
| | | ELM | ACNM | ELM | ACNM | ELM | ACNM |
| D1 | 4 | 0.458797 | 0.471726 | 0.392594 | 0.418452 | 3 | 1 |
| D2 | 9 | 0.61823 | 0.670426 | 0.422506 | 0.526898 | 7 | 4 |
| D3 | 16 | 0.706657 | 0.755127 | 0.44034 | 0.537281 | 13 | 9 |
| D4 | 25 | 0.66775 | 0.737539 | 0.391056 | 0.530634 | 21 | 16 |
| D5 | 36 | 0.683304 | 0.771348 | 0.366608 | 0.542696 | 31 | 25 |

In the first level, the unit difference with the product of $I_p$ , the matrix is computed. In the second level, the transpose value of the first level is calculated. In the third level, the matrix product of filter and $BP_S$ with the inverse, the filter is estimated. In the last level, the product of the first level with the current error covariance is evaluated with the product of second-level values Here $FC_H$ The is computed as below.

$$FC_H = BP_S D(NE_D I_p BP_S D)^{-1}$$

After the computation of $FC_H$, the current noise $N_O$ level from the $FC_H$ is calculated by using the summation of estimation with the product of $FC_H$ and the difference of sensor sensing dimension and the product of the observation matrix and previously predicted estimation.

$$N_O = BP_S + FC_H(S_D - I_{Pi} BP_S)$$

As per the three levels of matrix computations, the output value is combined using the summation operator with the third level value. Once the $FC_H$ Values are computed, then the radial basis function-based neural network formation is initiated. Later, the matrix is updated with the error value differences of the currently perceived $S_D$along with the new error differences$N_{ED}$.

$$N_{ED} = V - S_D$$

Also, the reorder gain value of the minimum and maximum values of the sensed data updated into the state matrix as $D$, The mean value deviations $M_D$ Computed between two different periods of sensed data, they can vary according to the respective context was calculated as below.

$$M_D = DN_{ED}D_s + T_{ED}$$

The minimum and maximum values of the differences are updated to see two error value differences$T_{ED}$. We continuously need to update the data packets sensed by sensors and then separate and store the minimum and maximum values to find these error differences. Then, find the lower boundary nearest value $N_{VL}$ and the upper bound values $N_{PU}$ of those minimum and maximum values$N_V$.

$$T_{ED} = \max(N_{VL}, N_{VU})$$

We updated the lower boundary and upper boundary values as follows,

$$N_{VL} = M_N - S_D$$

$$N_{VU} = M_A - S_D$$

### 3.7 Prediction of Noise and Nearest Values

Then the prediction of the state estimate is computed without the process gain in the predicted estimate for the current measurement. The bound values in both lower and upper are used to determine the minimum nearest value and the corresponding noise point of the sensed data. We assign the minimum values between these as the closest values $N_V$ and the maximum values as the noise values $V_{NO}$ as below.

$$N_V = \min(N_{PL}, N_{PU})$$

$$V_{NO} = \max(N_{PL}, N_{PU})$$

Perhaps, in a case $N_V$ or $V_N$ is not available, we need to calculate the grip point $G_P$ as shown below.

$$G_P = \frac{\sqrt{N_{VL} + N_{VU}}}{2}$$

Minimum and maximum values are identified from the sensed filtered values to normalize the values, and it is represented as a primary K matrix. The difference between the average

and the sensed values are determined as deviations and represented as M matrices. For this update, the points of sensed maximum and minimum values are known as normalized values $N_A$. Each sensor maintains a packet table$P_T$. Sensors update the packet counts and the delay in this table periodically.

$$N_A = (1 - FC_H M_D(1 - T_G D + FC_H N_{ED}))$$

Minimum and maximum values are identified from the sensed values to normalize the values, and it is represented as a primary K matrix. The difference between the average and the sensed values is identified as deviations and updated its lower $L_B$ and upper bound $U_B$ values in M matrices, for this update, the points of sensed maximum and minimum values are defined as $N_A$.

$$L_B = (M_V - S_{DV}), U_B = (M_V + S_{DV})$$

The corresponding boundary values of sensed data are represented as $B_V$ in-state matrices and the initial predicted value is marked as $P_I$ is specified as 1 in a matrix. As per the activities, the state observation matrix $I_P$ is uniformly distributed with 0 and 1. The matrix is filled with minimum and maximum values.

For the sensed data matrix, the initial weight $w$and bias$B$ are randomly distributed between 0 and 1. The positive and negative values variance is computed between the average sensed value and the currently sensed values. If the $V$is > 0 update $\eta +$, else update $\eta -$.

At this stage, the function extended with neural networks to improve the efficiency of the data aggregation. The neural network is handled with three layers in this work, that is an input layer to receive the inputs, the second layer is a mid hidden layer, here nonlinear initiation role is employed to do the processing, and the final layer of output is made in a linear model as shown in Figure.3.

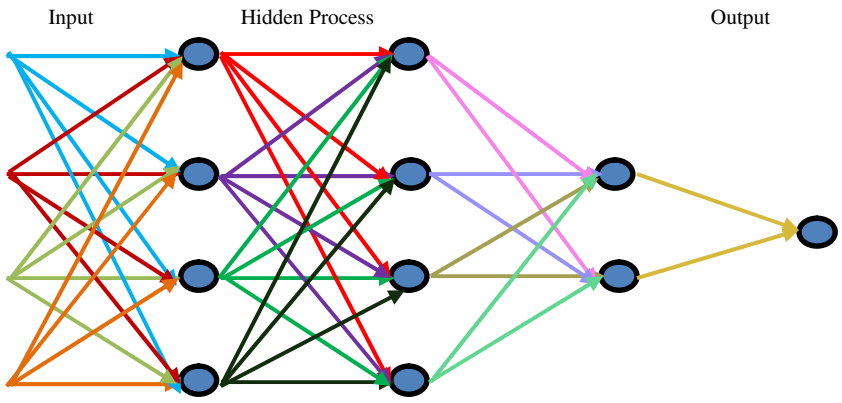

Figure.3 Neural networks behavior

We have the activation value from the processed value of the hidden layer, and this activation value is collected from the distance amid the center of the neuron and input. Based on the Gaussian function, $\varphi$ the value of the $S_D$ is computed using the negative exponential distribution of the squared value of sensed value and its $M_V$ with the normalization of the standard deviation. Here $\varphi$ is computed as below:

$$\varphi = e - \frac{(S_{DV} - M_V)^2}{2\eta^2}$$

$$\varphi(S_{DV} - M_V) = B + w(\sqrt{(x-y)^T V^{-1}(x-y)})$$

The transpose $T$ radial bias kernel is computed as follows

$$R_{BK} = B + w(\sqrt{(x-y)^T V^{-1}(x-y)})$$

$C_F$ Defines the capability of the bias function, it will vary according to the deviations is computed as below:

$$C_F \pm \varphi w + B$$

The neural network consists of three layers, and they have input is hidden and output layers. This is considered as input to the neural network, and the hidden neurons are formed using the $M_V$ Value, variance $V$value, and the distance between the input and center value, these values are used to maintain the ease of the Gaussian function. If $S_D <$ $M_V$value than the negative variance $V_N$ value is used to compute the hidden neuron as below.

$$V_N = e - \frac{(S_D - M_V)^2}{2\eta^{-2}}, \quad V_N = e - \frac{(S_D - M_V)^2}{2\eta^{+2}}$$

If the $S_D$ > $M_V$ Value then the positive variance value is used to compute the hidden neuron. Once the hidden neuron space is formed, the distance of the populous-variable track $\varphi$is estimated instead of the regular distance values. In case the sensed Data is > $V$,

$$\varphi i = e - \frac{\sqrt{S_D - M_V}}{2\eta +} \text{ else, } \varphi i = e - \frac{\sqrt{S_D - M_V}}{2\eta -}$$

In the distance of the populous-variable track, the level is estimated as the product of transpose of difference matrix of $S_D$and $M_V$ with the inverse function of covariance of the difference matrix of $S_D$ and $M_V$.

$$x = \varphi i(1 - w) + (1 - B)$$

$$y \pm \varphi i(1 - w) + (1 - B)$$

And the estimated level value is given as an input to the square root function to compute the distance$S_D$.

$$D_{ist} = \sqrt{(x - y)^T V^{-1}(x - y)}$$

This distance value is used in the hidden neuron $H_N$by merging with the weighted matrix and bias $F_W$ Defines the function weight.

$$H_N \pm \alpha(1 - w) + (1 - B)$$

$\alpha = BF_W$, here, $F_W \pm xe^{-x}$

***Hidden layer process:***

*Initialize*$\alpha\alpha xyx$, $\alpha\sum$

*Update the* hidden *layer processing counts*

$\sum \pm \sum \pm xF_W$

$x = \varphi i(1 - w) + (1 - B)$

$M_L$*with Input hidden process is computed as below*

$\alpha x = \sum + B$

$yx \pm \alpha(1 - w) + (1 - B)$

$\alpha\sum \pm \alpha x$

$$\alpha = \frac{\alpha\sum}{N_D}$$

$Hidden\ Output H_O = w\alpha x + \alpha x$

Here, $F_W$ defines the function weight, later the corresponding radial basis function kernel output is computed as a weighted summation of the distance-based $\varphi i$ value with the activated bias vector, update the hidden layer process as $H_P = \varphi i F_{Wj}$ and the flow of process discussed below:

Also, the kernel bias $K_B$ $a$ function is computed $as\ (1-B)+H_P$ and it is initialized randomly, $M_L$ is used to enhance the data accuracy during the training period. Afterward, the aggregation accuracy is improved by using this kernel process as $A_{Agg}$, here, $N_D$ defines the number of data.

$$A_{gg} \pm (1-w)K_B + (1-B),\quad AA_{gg} = e - \frac{A_{gg}}{N_D}$$

Total bias kernel computed from $TB_S \sum K_B$And the radial bias is computed as $B_R = \frac{TB_S}{N_D}$. Once the radial bias function is formed, it is projected with machine learning, which converts the random and non-trainable weight and bias into trainable weight and bias. In $M_L$The hidden nodes are formed by giving the weighted data as input to the inverse function and the biased value. Here, the negative exponential function is used as the inverse function in the hidden nodes. The formed hidden nodes are given as an input to the weighted summation value of the hidden node with the inverse bias value. It produces the output $H_O$ Value of the machine learning network and the same is given as an input to the least-square fitting to determine the hidden output.

$$H_O = wB_S\varphi t + TB_S$$

After identifying the hidden output, the learning value is computed by taking the minimized hidden output$MH_O$ With the inverse weighted value along with the difference of the neural output value. The complete neural process is discussed below:

- In the neural network, the following process is evoked to train and test the data's generated by the network,
- Data generation imitated by the network
- Trained the data set from the generation
- Computed the start function $\varphi i$ for the number of hidden processes
- Enclosed the parameters of $\alpha$ and $B$
- Allocated each neuron with $\eta$ and $M_V$
- Computed the Hidden layer output $H_O$ and weighted-based hidden layer output

For the minimized value, the final kernel value is determined by considering the weighted sum of $\varphi i$ value with the inverse activated bias value.

$$MH_O = H_O(1 - w) - H_N$$

$M_L$The following process will be considered. Machine learning-based neural processes, during the prediction time, the *w* of the connection, are coordinated with the midst of the neuron transform at the hidden layer. The utmost intention of the training system with the internal neurons is to put the Gaussian functions. During the data clustering time, the cost of neurons is governing with *w*, with the training and *w* allocation accomplishing the ultimate productivity $U_P$ is obtained.

- In the $M_L the$ following process is evoked to train the data's generated by the network,
- Trained the data from the generation
- Compute the start function $\alpha$ for the number of hidden processes
- Allocate w and $B$
- Computed the weight-based Hidden layer output $H_O$

$$U_p = H_O(1 - w) - R_{BK}$$

$$U_{PS} = \pm U_P$$

$$U_{PAC} = \frac{U_{PS}}{N_D}$$

The current measurement data values of the $S_D$ are given as an input to the forming machine learning-based distance-based redial basis function. The $M_L$ based distance-based redial basis function neural network produces the aggregated Data $U_{PAC}$ to be forwarded to the destination $F_D$.

$$F_D = \frac{U_{PAC} + R_{BK}}{WN_D - 1}$$

## 4. Results and Discussion

In this portion, the proposed scheme is tested using NS-3- 3.30 simulator to determine the effectiveness of the ACNM technique in terms of cluster precision, energy consumption, and network survival.

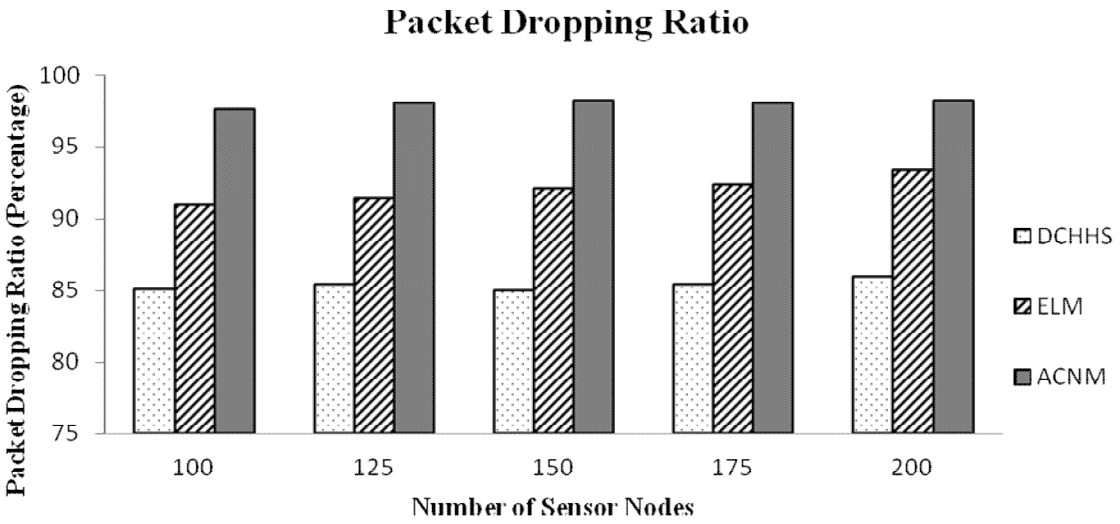


Figure.4 The comparison of PDR of Different Technique with ACNM

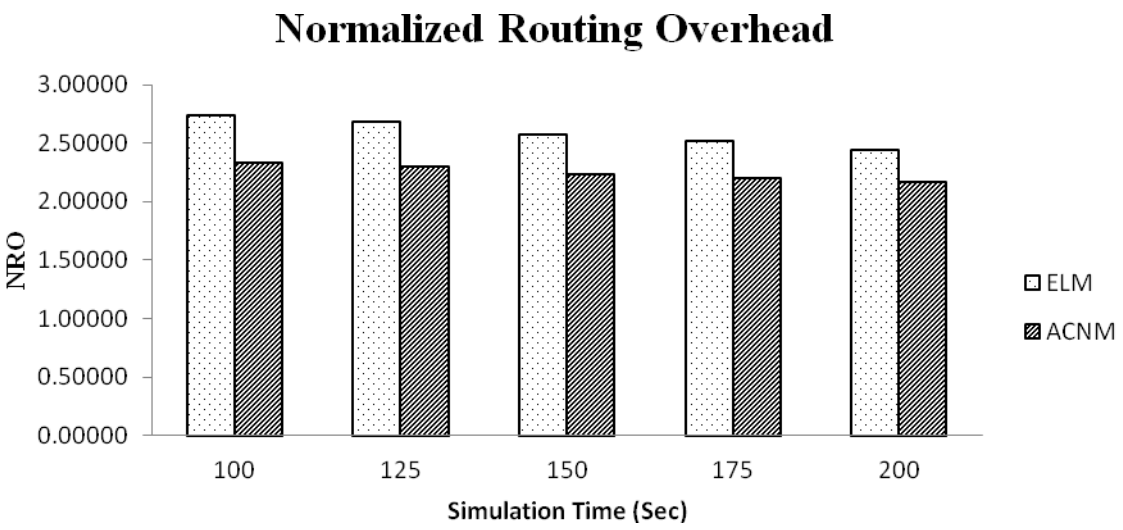


Figure.5 The comparison of Normalized Routing overhead of EDAM with ACNM

The packet delivery ratio is the percentage of packets dropped to the destination if the max. Number of packets delivered in a network, the delay in that network will be reduced. Similarly, network overheads will be reduced. Here in the above Figure 4,5, ACNM output has a higher packet delivery ratio. This allows us to know which computations in the ACNM protocol work best for the sensors. In this, the sensed Data is processed and then learned, later the current data in the neural network is taken as testing data, and all the other Data is considered as training data and processed in the hidden layer. Then the data goes to the destination in aggregated mode. So, the packets delivered more in the proposed method, as shown above.

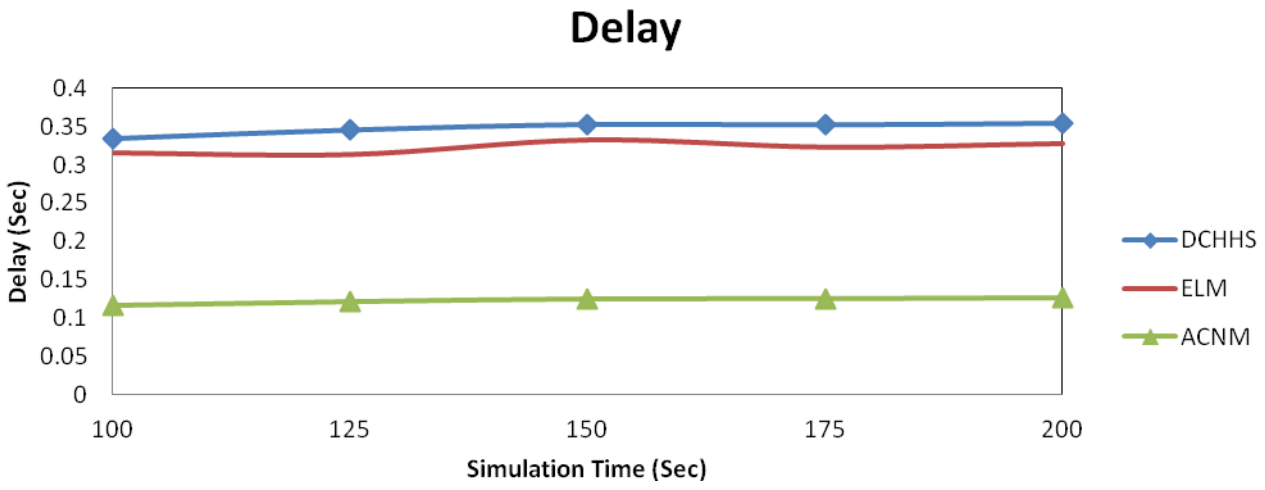

Figure.6 The evaluation of Network Delay.

Network delay is the delay in ACNM that is minimized due to various technical views and computations, such as the computation periods, redundant data transmissions reduction, and controlling the data speed of the previous node depending on the current node receiving capability. As well as reducing noise and errors in the data can send the correct information to the destination. These calculations also minimize delivery delays and avoid unnecessary losses. If the delay in the network is short, then the network is performing better. In the figure given here, ACNM is seen with less delay. This shows that the highest packets are delivered in the following graphs, as shown in Figure.6.

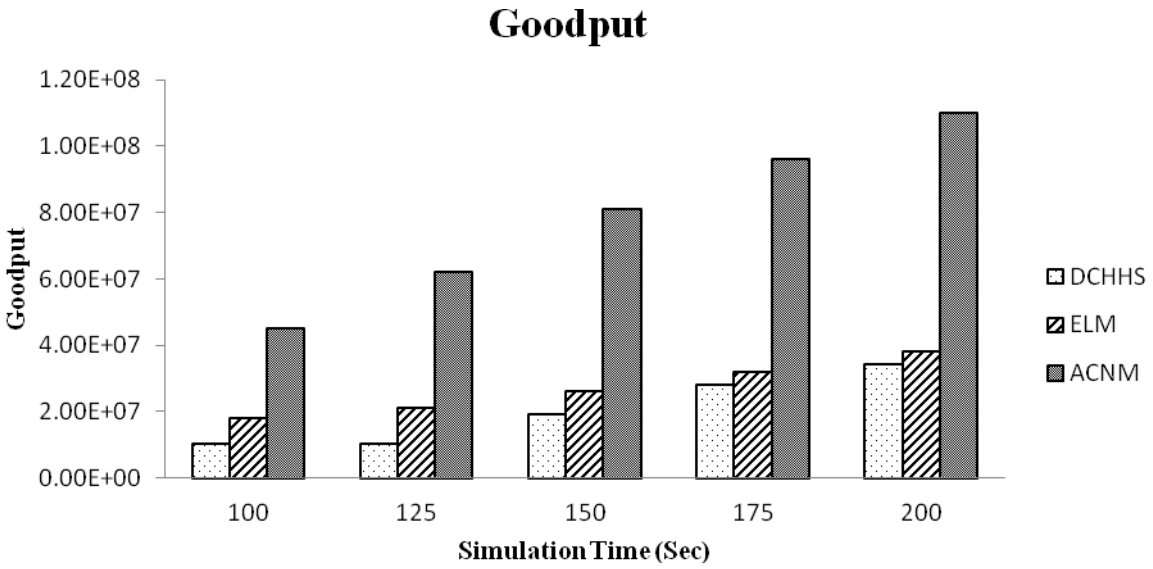


Figure.7 The comparison of Good put with Different Technique

Goodput refers to the receiving data bits per second delivered from source to destination. It has many features to improve the transmissions, such as the network model, the number of nodes, and protocol behaviors. In this ACNM design, the network is designed as a protocol with very precise calculations. It analyses the critical areas of data filtering and machine learning-based data aggregation in depth. As a result, we can see in this figure that the number of pocket bits received is high. Aggregation mode controls the redundant transmission of a network, also saving network resources bandwidth, energy, etc. This allows the network to work for a long time, as shown in Figure.7.

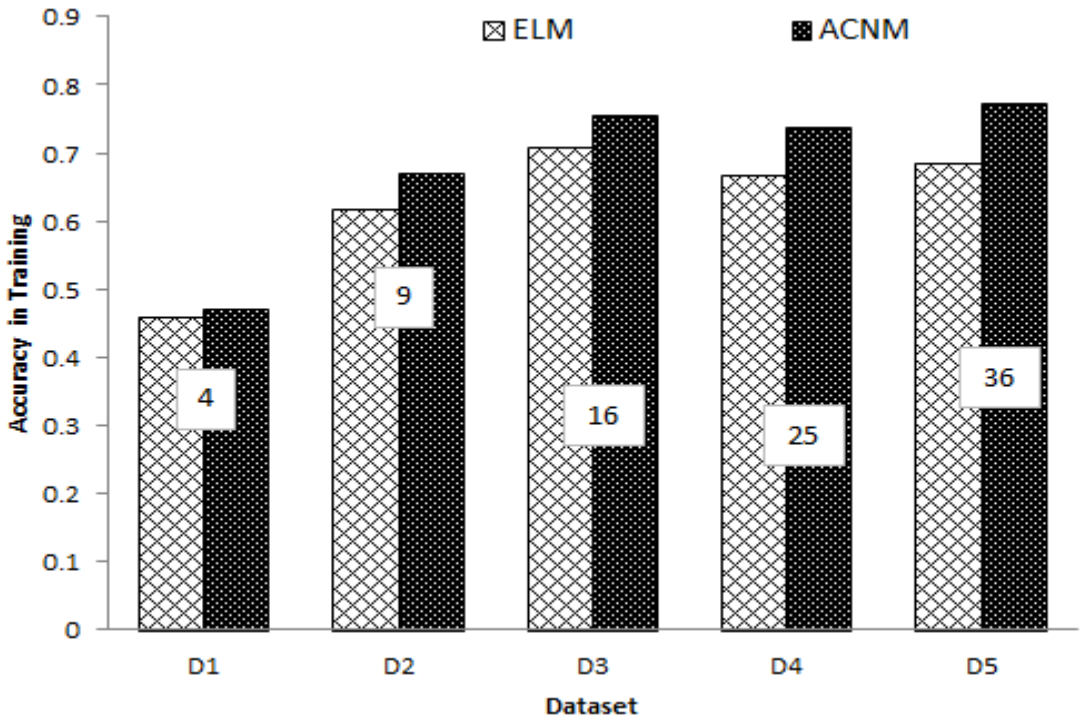


Figure.8 Accuracy in Training

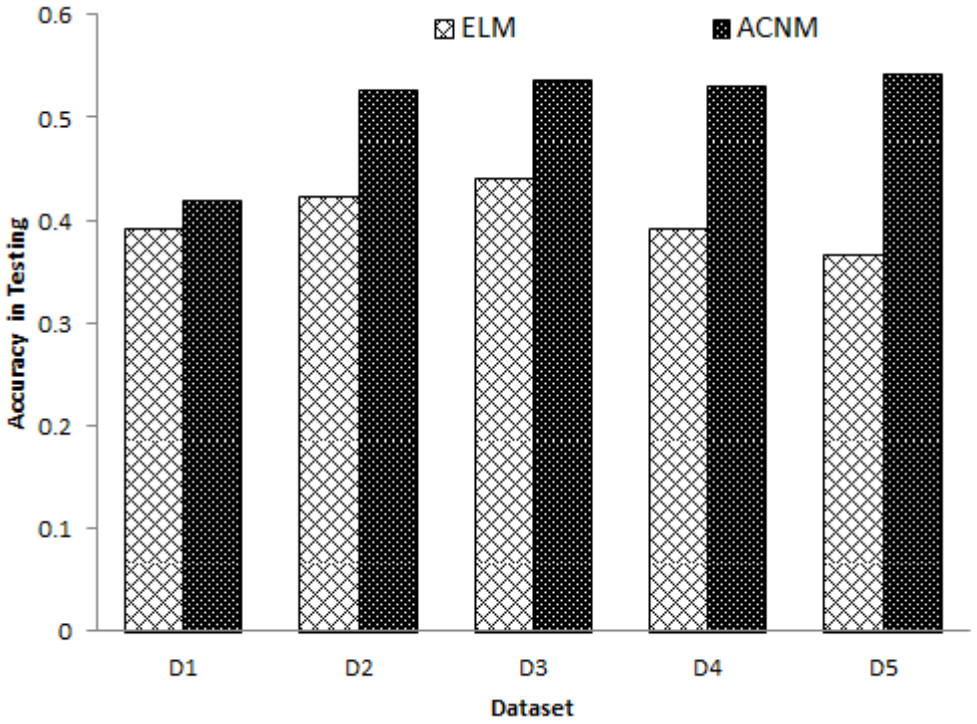


Figure.9 Accuracy in Testing

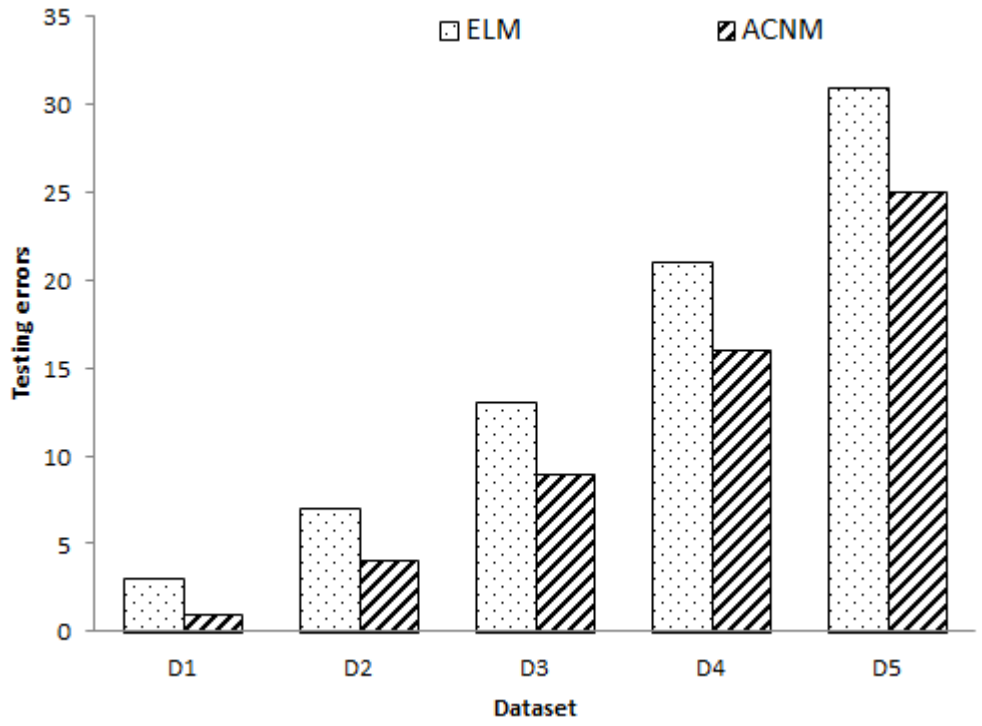


Figure.10 Testing errors

Fig. 8,9,10 The accuracy of training and testing, ELM error testing, and the suggested ACNM are shown in the middle layer of the model with specific neuron numbers from 4 to 50. The number of neurons in the middle layer is altered to train the model and produce a better outcome with fewer errors.

## 5. Conclusion

This ACNM design presented the machine learning-based data aggregation for sensor networks. Here the network is constructed based on the clustering method. The topology contains sensors classified as cluster heads and members. Each member senses the data and transfers it to the cluster heads. Then the cluster head runs the ACNM methodology to filter the noise and errors initially from each sensor. Later, it runs the machine learning-based neural network to train and test the data with learning techniques. Later, it aggregates the sensed Data before passing it on to the destination. This way of packet delivery in sensor networks reduces packet redundancy and network load. The network nodes accurately handle the entire network resources, thus minimizing the delay and network overheads. These reductions enhance the network life to work on the region than the normal life period. So that the network substantially reduces data transmission, which controls the delay as shown in the result. The entire network manages the network load even in a critical environment. In the

future, the network nodes can build with the trust model to get more accurate results and assure the security system built in the network so that the network nodes can communicate with the genuine nodes to minimize the errors and losses.

**Declaration:**

Ethics Approval and Consent to Participate:

No participation of humans takes place in this implementation process

Human and Animal Rights:

No violation of Human and Animal Rights is involved.

Funding: No funding is involved in this work.

Conflict of Interest: Conflict of Interest is not applicable in this work.

Authorship contributions:

There is no authorship contribution

Acknowledgment :

No Acknoweldgments

Reference

[1] Ullah I, Youn HY (2018) Statistical multipath queue-wise preemption routing for ZigBee-based.Wirel Pers Commun 100(4):1537–1551

[2] Gravina R, Alinia P, Ghasemzadeh H, Fortino G (2017) Multisensor fusion in body sensor networks: state-of-the-art and research challenges. Inf Fus 35:68–80

[3] Zhang Q, Yang LT, Chen Z, Li P (2018) A survey on deep learning for big data. Inf Fus 42:146–157

[4] Sun L-Y, Cai W, Huang X-X (2010) Data aggregation scheme using neural networks in wireless sensor networks. IEEE, Piscataway, pp V1–725.

[5] Liu C, Wu K, Pei J (2007) An energy-efficient data collection framework for wireless sensor networks exploit spatiotemporal correlation. IEEE Trans Parallel Distrib Syst 18(7):1010–1023.

[6] Sung W-T (2009) Employed BPN to multi-sensors data fusion for environment monitoring services. Auton Trust Comput 5586:149–163

[7] Villas LA, Boukerche A, Guidoni DL, De Oliveira HA, De Araujo RB, Loureiro AA (2013) An energy-aware Spatio-temporal correlation mechanism to perform efficient data collection wireless sensor networks. Comput Commun 36(9):1054–1066.

[8] Hossain MA, Atrey PK, El Saddik A (2009) Learning multisensor confidence using a reward-and-punishment mechanism. IEEE Trans Instrum Meas 58(5):1525–1534.

[9] Kang S, Lee J, Jang H, Lee Y, Park S, Song J (2010) A scalable and energy-efficient context monitoring framework for mobile personal sensor networks. IEEE Trans Mob Comput 9(5):686–702.

[10] Nath S (2012) ACE: exploiting correlation for energy-efficient and continuous context sensing. ACM, New York, pp 29–42.

[11] Jiang Y, Qiu H, McCartney M, Halfond WG, Bai F, Grimm D, et al (2014) Carlog: a platform for flexible and efficient automotive sensing. ACM, New York, pp 221–235

[12] Li G, Wang Y (2013) Automatic ARIMA modeling-based data aggregation scheme in wireless sensor networks. EURASIP J Wirel Commun Netw 2013(1):85

[13] Zhao M, Ma M, Yang Y (2011) Efficient data gathering with mobile collectors and space-division multiple access techniques in wireless sensor networks. IEEE Trans Comput 60(3):400–417

[14] Y. Ma, Y. Guo, X. Tian, M. Ghanem, Distributed clustering-based aggregation algorithm for spatially correlated sensor networks. IEEE Sens. J. 11(3), 641–648 (2011)

[15] Y. Lu, N. Sun, A resilient data aggregation method based on Spatio-temporal correlation for wireless sensor networks. EURASIP J. Wirel. Commun. Netw. 1, 157 (2018)

[16] Valian, E., Mohanna, S., & Tavakoli, S. (2011). The improved cuckoo search algorithm for feedforward neural network training. International Journal of Artificial Intelligence & Applications, 2(3), 36-43.`

[17] Ihsan Ullah., Hee Yong Youn(2020), Efficient data aggregation with node clustering and extreme learning machine for WSN, The Journal of Supercomputing, https://doi.org/10.1007/s11227-020-03236-8

[18] S. Pirbhulal, H. Zhang, W. Wu, S.C. Mukhopadhyay, Y.T. Zhang, Heart-beats based biometric random binary sequences generation to secure wireless body sensor networks, in IEEE Transactions on Biomedical Engineering, SCI (2018), pp. 1–1

[19] Tabibi, S., & Ghaffari, A. (2019). Energy-efficient routing mechanism for the mobile sink in wireless sensor networks using particle swarm optimization algorithm. Wireless Personal Communicate ions, 104(1), 199-216.

[20] Vimal Kumar Stephen, K, & Mathivanan, V. (2019). Neural network-based virtual backbone tree construction and dynamic sink implementation enhance the lifetime of the network and minimize energy consumption. International Journal of Advanced Intelligence Paradigms, 13(3-4), 304-323.

## AUTHORS BIOGRAPHY

***C.Sudha*** *is currently a Part Time Research Scholar in CSE Department at Annamalai University, Chidambaram, and Tamilnadu. She completed her M. E in CSE from Anna University Regional center, Coimbatore in 2013.She Received. her B.E degree in CSE fromSona collegeof Technology, Salem.Currently she is working as an Assistant Professor at Mahatma Gandhi Institute of Technology, Hyderabad. Her Research interest includes Computer Networks, Wireless Sensor Networks, IoT based Architectures.*

**DrD. Suresh**received the B.E(IT)., degree in Mohamed Sathak Engineering College in 2004. He received M.E degree in Computer Science and Engineering from the Annamalai Universityin 2008. He has been with Annamalai University, since 2005. He completed his Ph.D degree in Computer Science and Engineering at Annamalai University, in the year 2015. Heis currently an Assistant Professor in Information Technology at Annamalai University. He published 20 papers in international conferences and journals. His research interest includes Mobile Networks, Network Security, Wireless Sensor Networks and Network Simulator.

**Dr A.Nagesh** is currently working as a professor in CSE at MGIT, Hyderabad. He completed B.E and M. Tech from Osmania University, Hyderabad in 1996 and 2002 respectively. He completed hisPh. D in CSE from JNTUH, Hyderabad in the year 2012. He is having total 22 years of teaching experience. Presently he is supervising five Ph. D students. He published 40 papers in national & international journals. His research areas include pattern recognition, speech processingand data mining.