Energy Efficient Data Aggregation in Wireless Sensor Networks Using Meta Heuristic Based Feed Forward Back Propagation Neural Network Approach

[1]Navjyot Kaur and [2]Vetrithangam D
[1,2]Chandigarh University, Mohali, India
[1]navjyot16.kaur@gmail.com, [2]vetrigold@gmail.com

Correspondence should be addressed to Navjyot Kaur : navjyot16.kaur@gmail.com

Article Info
Journal of Machine and Computing (http://anapub.co.ke/journals/jmc/jmc.html)
Doi: https://doi.org/10.53759/7669/jmc202404062
Received 28 November 2023; Revised from 26 January 2024; Accepted 08 June 2024
Available online 05 July 2024.

Abstract – Sensor nodes are low-cost, low-power, tiny devices that make up the majority of WSNs, or distributed, self-organizing systems. These sensor nodes are able to exchange, perceive, and interpret data. The sensor nodes are equipped with a wide variety of sensors, such as chemical, touch, motion, temperature, and weather sensors. Because of its adaptability, sensors are used in a variety of applications such as automation, tracking, monitoring, and surveillance. Despite the enormous number of sensor applications, WSNs continue to suffer from common challenges like as low memory, slow processing speed, and short network lifetime. The feed forward back propagation neural network mode (FFBPNN) based on meta heuristics aims to create many paths for effective data aggregation in wireless sensor networks. This model handled the process of identifying and selecting the optimum route path. The distributed sensor nodes are utilized to create the various route paths. In this research paper, data aggregation is done using meta-heuristic firefly algorithm that helped in identifying an optimal route from among the found routes. After selecting the operative ideal route choice, the data aggregation procedure practices a rank-based approach to accomplish lower latency and a better packet delivery ratio(PDR). In addition to throughput, simulation was done to improve and measure performance in terms of packet delivery ratio, energy consumption, and end-to-end latency.

Keywords – WSN, Routing, Data Aggregation, Clustering, Energy Efficient Techniques, Feed Forward Back Propagation Neural Network

I. INTRODUCTION

Wireless sensor networks (WSNs) plays an important role in tremendous applications like industries, smart cities, automotive manufacturing, healthcare, and environmental monitoring. In all these environment, self-governing sensor nodes are distributed throughout to monitor various environmental and physical conditions. Sensors collect data, and then send it to a central node for processing and further data analysis. Scalability, self-organization, and adaptability are among the characteristic of WSNs that make them best tools for catching data in real-time in potentially dangerous or inaccessible circumstances. Data aggregation and routing are two critical components of WSNs that support to enhance energy efficiency, increase the network lifetime, and reduce communication overhead [1]. Data aggregation is the method of aggregating raw sensor data at intermediate points before broadcast to the sink node in order to reduce redundant data transfer and conserve energy. Routing algorithms use various features such as energy levels, network topology, and data transfer patterns to determine the route that data takes as it travels from source nodes to sink nodes. Effective data aggregation and routing algorithms are required in WSNs for reliable information delivery, low latency, and reduced network congestion. However, various issues prohibit WSNs from becoming extensively utilized and effective[2][3]. These limitations include the limited energy resources of sensor nodes, altering network topologies, communication restrictions, and data processing constraints. Because battery-powered sensor nodes in wireless sensor networks (WSNs) are regularly used, energy consumption is a major consideration. Furthermore, the dynamic nature of network topology and environmental conditions introduce uncertainty, necessitating dependable and flexible data aggregation and routing solutions [4]. In recent years, the combination of machine learning and neural networks has been proposed as a potential solution to WSN difficulties. Machine learning algorithms, which include supervised, unsupervised, and reinforcement learning techniques, enable sensor nodes to learn from data and make informed judgments without the need for explicit programming [5]. In WSNs, neural networks have shown considerable potential for tasks such as data prediction, anomaly detection, and energy.

II. LIMITATIONS OF WSN

Table 1 shows the challenge in WSN. WSNs have many benefits for a range of applications, but they also have certain built-in drawbacks. These are major drawbacks[31]:

Table 1. Challenge in WSN

Sr. No.	Challenge	Description
1.	**Limited Energy Resources**	Sensor nodes typically rely on batteries or energy harvesting, which have finite energy capacities and require energy conservation.
2.	**Network Scalability**	Ensuring efficient network operation and management as the number of nodes increases is very challenging task for WSNs.
3.	**Communication Reliability**	Data reliability is impacted by wireless communication's susceptibility to interference, signal attenuation, and packet loss is a huge setback for WSNs.
4.	**Data Aggregation**	Aggregating and consolidating data effectively to reduce transmission overhead in WSNs and consuming less energy for this task is a major challenge in WSNs
5.	**Network Lifetime**	Extending the network's life through efficient energy usage and careful battery management is a challenging task for which a lot of research is ongoing.
6.	**Security and Privacy**	One of the main challenges with WSNs is protecting data privacy, integrity, and confidentiality from unauthorized access and modification.
7.	**Fault Tolerance and Node Failure**	One of the main challenges in WSNs is developing ways to identify and recover from node faults or failures to assure continued functioning.
8.	**Network Coverage and Connectivity**	It can be difficult for WSNs to provide sufficient coverage and connectivity in a variety of dynamic situations.
9.	**Time Synchronization**	In WSNs, synchronizing sensor node clocks to provide coordinated operations and data fusion is a difficult task.
10.	**Quality of Service (QoS)**	In WSNs, it might be difficult to meet application-specific criteria for latency, throughput, and reliability.

III. WORK IN THIS AREA

This section presents previous research done on data aggregation(DA) in WSNs. It is the process of collecting information from numerous nodes and passing it to nodes further up the network. DA lowers repetition of data in the network. Nguyen et al. (2016) used DA to recover the shortest path in a wireless sensor network. To do this, they developed a local tree reconstruction method and evaluated its performance in terms of aggregation and network lifetime. Despite achieving a longer network lifetime, the authors were unable to regenerate trees in the dynamic environment of the WSN [12]. Prathima et al. (2016) presented the concept of secure DA to handle many WSN queries that are resolved through DA followed by encryption [13]. Secure aggregation aided in distributing the query, minimizing the total energy usage for the process. A simulated study was made to determine the packet drop ratio, communication time, and energy usage using secure data aggregation. The scheme's key faults were its limitations in its capacity to measure performance against two states and its incapability for handling lots of queries. Wang et al. (2018) proposed using the perception of perpetual information aggregation for transmission in WSN to ensure energy-efficient privacy preservation. Two steps called data slicing and data aggregation, were developed with little energy and communication overhead. Overall, the results were expected to validate outstanding scalability and high accuracy. Even though the technique worked fine, it was impossible to reduce communication overhead due to poor correlation factor between sensory data [14]. Idrees et al. (2019) used the divide and conquer policy by strengthening K-means for congregation data in WSN, keeping energy conservation in mind. DA happens at two levels: the node and cluster head level [15]. When the suggested approach was applied at the sensor nodes level, redundant data was reduced and the optimal cluster collection was produced. Overall, the test demonstrated higher energy economy and data quality during data aggregation in WSN. According to Kaur and Manjul (2020), DA is one of the important technique that allows authors to drastically reduce the control energy consumption of individual nodes by deleting extraneous information from obtained WSN data. This drop contributes to the extending the WSN lifespan [11]. The authors of this study attempted to give an overview of the various DA strategies that were being used in the literature at the time. The authors then attempted to rank each of these DA techniques using a variety of performance metrics, such as energy utilization, latency, network longevity, energy expenditure, and so on. Zhang et al. (2020) proposed a multi-algorithm sensors deployment strategy to maximize WSN lifetime. The suggested Entropy-driven Data Aggregation with Gradient Distribution (EDAGD) deployment technique includes three methods: multi-hop tree-based data aggregation, entropy-driven aggregation tree-based routing, and gradient deployment. When all sensors near the sink lose power, the WSN connections are disconnected[16]. Because all of the connections to the sink have failed, data collected from the terminal sensors is unable to reach it. Empirical and numerical data show that the recommended DAGD technique outperforms the standard algorithm with the randomized setup. In the meantime, the cluster head (CH) was chosen, and Babu et al. (2021) proposed the Integration of Distributed Autonomous Fashion with Fuzzy If-then Rules (IDAF-FIT) technique for clustering. The packet was then routed from the source to the target node using the best available path [19].

In this case, adaptive source location privacy preservation (ASLPP-RR) was recommended as a routing method. In addition, the Secure Data Aggregation based on Principle Component Analysis (SDA-PCA) technique ensures end-to-end secrecy and integrity. Finally, a complete investigation of sensitive data security produces results that outperform current methodologies. To address the issue of low latency, Pham et al. (2021) designed a network architecture and used a collision avoidance data aggregation technique. The authors prioritized data transport and created the relative collision graph using the scheduling approach. The performance evaluation computation for the [20] needed time periods has been completed.to reduce latency, and simulation results revealed that the recommended strategy performs better. John et al. (2022) proposed an energy-efficient data aggregation method to enhance the surveillance system's overall prediction analysis. This technique was constructed on the foundation of machine learning and swarm intelligence [21]. To complete the work with the very less amount of energy , the load was optimized and balanced using a technique called as Particle Swarm Intelligence (PSO). Furthermore, the authors said that enhanced scheduling reduced the system's computational complexity. When swarm-based sensor optimization was applied, the investigation system became tremendously cooperative, and SVM was used to further improve it. Jatothu et al. (2022) used honey bee swarm intelligence to decrease node power usage while interactive [22]. Furthermore, the simulated lead to a reduction in response rates. The project used clustering based on a swarm of bees to choose the best colony leaders for relaying the aggregated data. Waiting time and overall power consumption were reduced, according to an assessment of the work's effectiveness and performance in terms of packet loss ratio. Sharman et al. (2023) proposed an energy-efficient data aggregation clustering technique based on Particle Swarm Intelligence (PSO) [7]. Using a few assumptions, such as selecting the node from the WSN with the highest energy and the greatest number of nearby nodes as the CH, it was able to identify the cluster head using these strategies. The PSO approach determines the optimal CH by evaluating whether the conditions for the best particles are met. If not, a new exploration for the ideal particle will start. If not, it will determine which particles are at the top. The CH collects data from cluster members, aggregates it, and sends it to the base.

IV. RESEARCH GAPS,PROBLEM STATEMENT AND OBJECTIVE

The purpose of this research is to design a machine learning based Computational Approach for energy efficient data aggregation model in WSNs. Based on the studied literature, the proposed work is divided into two phases. The first phase aggregates the data that has been discovered by the route discovery process and the second phase performs an analysis of the aggregated data. The second phase aims to create a rule mining engine, that generates a score for each node based on the QoS parameters[6][8] that have been attained in segment 1. Overall process flow diagram is as discussed in **Fig 1**.

Proposed Methodology-Phase 1

The sensor nodes are deployed over a specified area size, "A," to begin Phase 1. Multiple regions are created from the nodes. Using the subsequent ordinal measurements, the proposed algorithm will determine an ideal number of 'k' values to divide the regions into segments. Following points are assumed while implementing the scenario:

- Total number of nodes to be deployed
- The distance of the node from the base station is calculated using equation 1.

$$\boldsymbol{d} = \sqrt{(\boldsymbol{nx} - \boldsymbol{bs_x})\text{^}\mathbf{2} + (\boldsymbol{ny} - \boldsymbol{bs_y})\text{^}\mathbf{2}} \quad (1)$$

- Area of deployment

The highest amount of residual energy associated to the nodes will be utilized in selecting the CHs at the beginning of the process, following the determination of the regions [9][10]. The sources will put forward the demands, and the neighboring CHs will be notified of their relevance via their respective clusters. It is evident that each CH, including the CH with the source's request, will have a certain sensing interval. Equation 2 will be used to determine the overall cost of transfer through the respondent CHs based on the responses received at the transmitter end.

Based on Wt(waiting time), the computation cost is calculated as defined by equation as follows.

$$Cc = Wt \times E_{idle_{ucn}} + Tc \quad (2)$$

Where Tc is the transfer cost and is dependent upon the total number of data packets to be transferred. $E_{idle_{uc_n}}$ is the unit cost of the consumed energy in idle time spent in the buffer under n^{th} node. The next CH who offers the lowest cost for the transfer will be selected by the present CH. Until the destination's route containing the CH is included, the process continues. Each CH will additionally need to aggregate the data packets that are given to it from other CHs throughout each sensing interval. A two-level computation is required to assess the QoS after the transfer [23, 24]. Both the node level and the route level computations are included in two-level calculations. While each node in the path listing undergoes a node-level computation, the receiver end does the route-level computation.

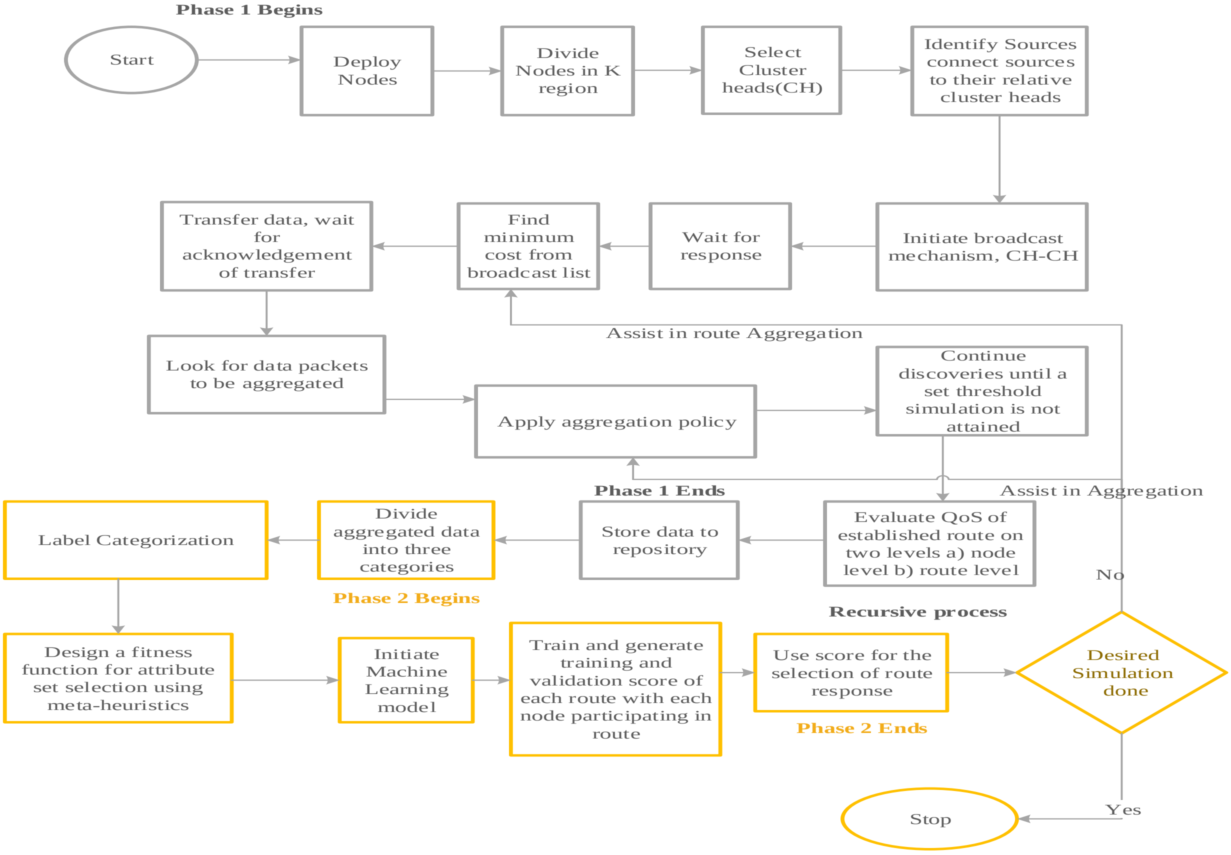


Fig 1. Proposed Methodology.

Proposed Methodology-Phase 2

The second phase aims to create a rule mining engine, that generates a score for each node based on the QoS parameters that have been obtained in phase 1 while doing route discovery and selection. Once the route is finalized, the QoS parameters are evaluated as shown in **Fig 1**. These aggregated data values will be passed for processing into phase 2. Phase 2 part aims to develop a ranking model for the nodes that participate in the route discovery process to utilize the rank score to choose the node for the route in the later stage. In addition to this, as has been illustrated earlier data aggregation is one of the major tasks to be performed, and hence the aggregated data will contain the delays that have been produced from data aggregation.

Training Of System

The QoS parameters like throughput, packet delivery ratio (PDR), packet drop ratio, latency, and energy usage collected from Phase 1 are used in phase 2 for doing the aggregation analysis and categorization of data in three categories labelled as Good, Bad and Moderate Transfer. These labels are generated on the basis of similarity index between the calculated parameters. The labelling is done on the basis of comparison of values with the help of centroid which is selected using k-means clustering mechanism. Due to the initial iteration's random centroid selection and subsequent adjustment of the centroids based on the Euclidean distance, which is created by extending the Cartesian distance system, the k-means clustering algorithm encounters problems [18] [25]. The first iteration of the k-means clustering algorithm has problems since the centroids are chosen at random, and later on, the centroids are changed using the Euclidean distance, which is created by extending the Cartesian distance system. The result of improved k-means will form 3 clusters but as it is a clustering algorithm and cannot name the clusters, statistical machine learning has to be utilized for the nomenclature. The hybrid similarity will be used as a standard parameter and labelling will be done using a rule-based fuzzy model[17]. Algorithmically it can be written as follows:

Algorithm 1: Data Aggregation Analysis and Clustering

1. Require: fg where **fg** represents formed groups
2. **similarities =[ ]** create a blank array of similarities
   For i in fg
3. **As=fg[i].Attributes** where As is the attribute set, In case of the proposed work it would be QoS parameters.
4. **Ce=As.cent()** ; //find centroid of current group.
5. **sim1=CalculateSimilarity1(ce,As)**; //Calculate Similarity 1
6. **sim2=CalculateSimilarity2(ce,As);** //Calculate Similarity 2
7. **asim=(sim1+sim2)/2**
8. **Similarities[i]= asim**, where asim is the similarity list

End For_i

9. **fmin=findmin(similarities);** //where fmin is the minimum similarity
10. **fmax= findmax(similarities);** //where fmax is the maximum similarity.
11. If **similarities.index.value** is **fmin,**

then co-relation is least and labeled as **bad transfer.**
Else If **similarities.index.value** is **fmax,**
then co-relation is most and labeled as **good transfer**.
else
co-relation is moderate and labeled as **moderate transfer.**
end if

14. return labeled

The labelled data along with their cluster identities will be passed to the data selector as the data selection step in the proposed work shown in **Fig 1** . Once the data is labelled, it is necessary to validate the formed group to draw an efficiency rank for each node that has acted as a data aggregator in the route. To validate the formed group, FFBPNN will be used due to the advantages of Neural Networks over other validators.

Optimized Next Node Selection using Machine Learning Based Approach

For each node that served as a data aggregator along the route, an efficiency rank must be determined by validating the formed group once the data has been grouped, or labeled. Due to neural networks' benefits over other validators, FFBPNN (Feed Forward Back Propagation Neural Network) will be utilized to validate the newly created group. This part of phase 2 requires the data saved in repository while performing route discovery and route selection. Following approach is used for doing same:

Algorithm 2: Optimized Next Node Selection using Machine Learning Based Approach

Input :

1. Retrieve the cluster info data from the **'clusterinfo.mat'** file.
2. Open **'networkoutput_firefly.mat'** to load the firefly algorithm output data.
3. Open the **'networkpattern'** file and load the network pattern data.

   Neural Network Training:

4. Assign the data of the **networkoutcomegraph** from **networkoutput_firefly.mat'** to the **datatocluster** variable.
5. Using the **newff** function, build a feedforward neural network net with 10 neurons in the hidden layer.
6. Set the neural network's training epoch count to 100 or higher if required.
7. With the loaded data **(datatocluster)** and the cluster indices **(k_index)** as targets, train the neural network **(net).**
8. Continue training the model till the convergence happen.

   Node Ranking and Simulation:

9. To produce **resultp**, simulate the trained neural network **(net)**
10. Store the average score for every node in the network, initialize an array called **node_rank.**
11. Store the actual score for every node, initialize an additional array called **actual_score.**
12. Get the current route from **networkoutcome** and calculate the route score. Evaluate number of nodes on the current path.
13. Calculate score per node by dividing the route score by the total number of nodes.
14. By averaging the score per node for every node in the current path, update the **node_rank** array.
15. Save the node ranks to a file named **'rankofnodes'.**
16. Pass the **rankofnodes to** firefly algorithm.

17. Find top two nodes with highest rank.

18. Select the next node with highest rank while considering cost factors also.
19. Repeat same process till destination node is not met.

End

V. IMPLEMENTATION-PHASE 1

While simulating the desired phenomenon certain parameters as mentioned in **Table 2** are considered. Along with that a Core i5 CPU running at 2.1 GHz, Windows 7 64-bit, 4 GB of RAM, and 4 GB of RAM is required for doing same [17][26]. The various steps involved for obtaining the desired results are as following:

Table 2. Parameters for Implementation

Sr. No	Parameters	Value
1.	Nodes	60 and 100
2.	Network Area	1000x1000
3	Software Used	MATLAB
4.	Power	0-100 mJ
5.	Placement of Node	Random

1. Firstly, the WSN nodes are distributed randomly in the area as in **Fig 2**. The entire area is divided into 1000x1000. towards x-y axis. The nodes are labeled as N1, N2, N3, ….., N60.

2. After creation of WSN, the nodes are divided into different areas using k-means clustering rule. By applying the 10% rule of LEACH[32]protocol, the respective number of Cluster Heads (CH) are selected on the basis of maximum Residual Energy (RE) in the respective nodes. As 60 number of nodes are considered initially so 6 CHs are chosen for forwarding the data to sink node. The clusters are formed on the basis of distance of sensing nodes from the CHs. The nodes nearest to the first cluster head (CH-C1) are aligned with CH. **Fig 3** represents the clustered data [33].

3. After the clustering process, with the help of routing process, the communication initiates between source and destination nodes. The source node broadcasts to destination. **Fig 4** illustrates the route discovery process followed by the WSN. Multiple routes are formed during packet delivery. Additionally, while data is sent between multiple nodes various QoS parameters are also calculated and saved during this transmission. **Fig 5** represent the QoS parameters collected in this case i.e. Packet Delivery Ratio (PDR), Throughput, delay (in ms), drop ratio and Total Energy consumed (in mJ) etc.

4.

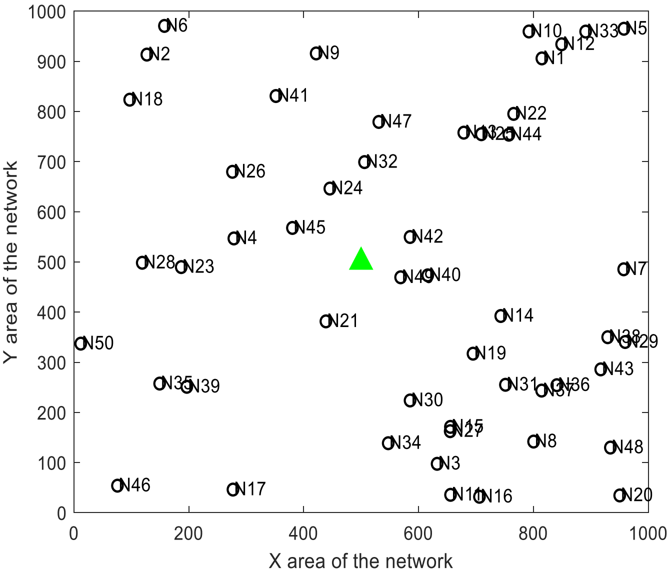


Fig 2. Network Structure.

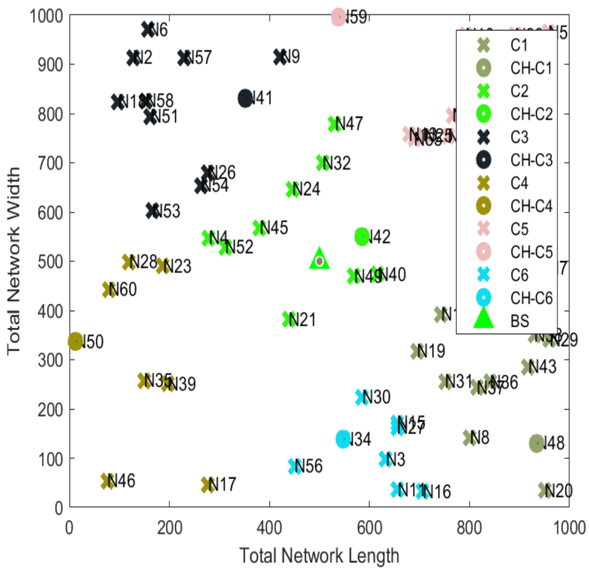


Fig 3. Clustering of nodes.

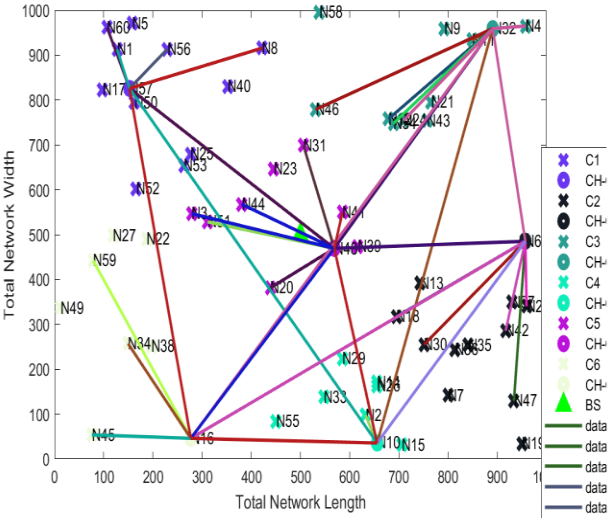


Fig 4. Route Discovery Process.

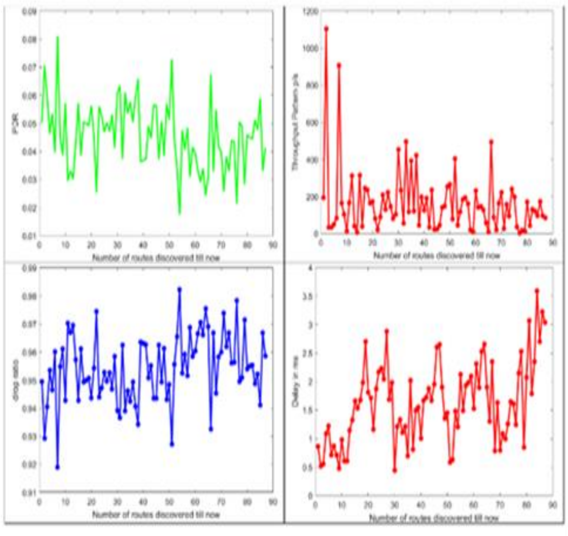
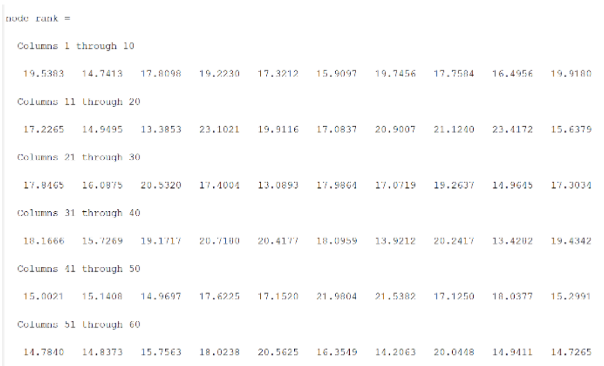


Fig 5. QoS Parameters Collected during data transmission phase.

IV. IMPLEMENTATION-PHASE 2

Phase 2 consists of optimized route discovery using Firefly Algorithm and afterwards clustering of the data is done using k-means algorithm. Phase 2 of this research can be understood using following steps:

1. The optimized route discovery is done using the Firefly algorithm. This optimization method is modelled after the social interactions of fireflies in their natural habitat. To lure in other fireflies, fireflies flash their beacons. On this planet as a whole, there are roughly 2000 recognized different species. The fundamental idea behind the firefly algorithm is that the brightest firefly will attract the attention of others, and if the firefly that is closest to you is not as bright as it is, it will travel randomly in quest of a brighter one. **Fig 6** represents the route discovered, throughput, PDR and delay incurred during optimized route discovery using metaheuristic algorithm.


```
node rank =

  Columns 1 through 10

   19.5383   14.7413   17.8098   19.2230   17.3212   15.9097   19.7456   17.7584   16.4956   19.9180

  Columns 11 through 20

   17.2265   14.9495   13.3853   23.1021   19.9116   17.0837   20.9007   21.1240   23.4172   15.6379

  Columns 21 through 30

   17.8465   16.0875   20.5320   17.4004   13.0893   17.9864   17.0719   19.2637   14.9645   17.3034

  Columns 31 through 40

   18.1666   15.7269   19.1717   20.7180   20.4177   18.0959   13.9212   20.2417   13.4202   19.4342

  Columns 41 through 50

   15.0021   15.1408   14.9697   17.6225   17.1520   21.9804   21.5382   17.1250   18.0377   15.2991

  Columns 51 through 60

   14.7840   14.8373   15.7563   18.0238   20.5625   16.3549   14.2063   20.0448   14.9411   14.7265
```

Fig 6. Parameters collected during optimized route discovery.

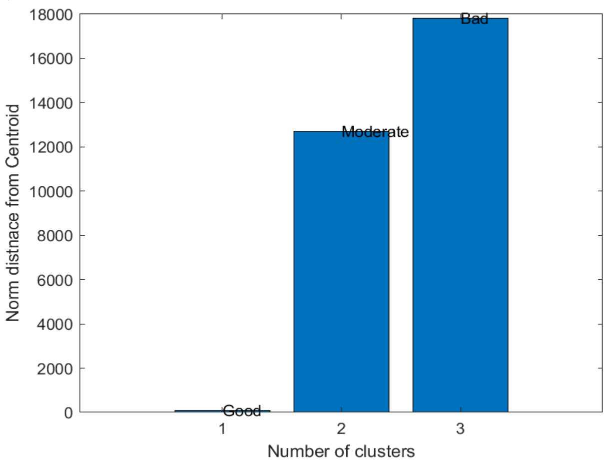
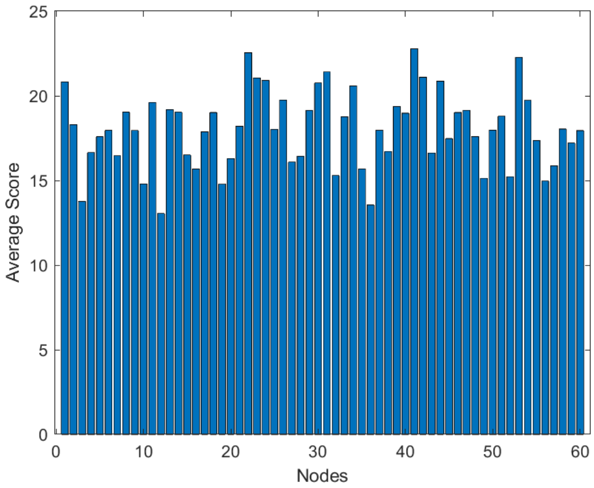


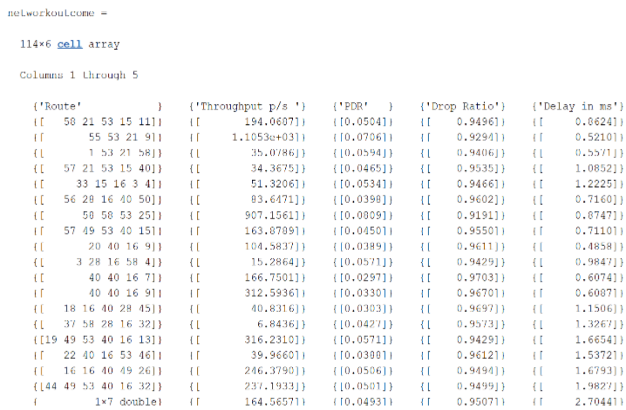
Fig 7. Good, Bad and Moderate Clusters.

2. An iterative technique called K-means clustering divides the given dataset into 3 unique, non-overlapping clusters. First, centroids are assigned to each cluster at random. The closest centroid is then iteratively allocated to each data point, and the centroids are then recalculated using the average of the points assigned to each cluster. Cartesian distance system is then extended to include Euclidean distance, which is used to modify these centroids. In this step the data is divided into three categories name Good, Bad and Moderate as shown in **Fig 7**, depending upon the similarity index between the transfer on the basis of Algorithm 1 as discussed previously. It shows the normalized distance of data from centroid.

3. On the basis of clustered data, the rank of entire route is calculated [29]. However, we need to know the rank for each node that is participating in the route discovery and data transmission process. In this method the route ranking score will be equally divided in all nodes involved in the route. eg. If nodes {1,5,10,20,15} are creating a route to deliver data between node 1 as source node to destination node 15 and route rank score is calculated as 100 then node rank will be 20 each as it

will be equally divided into 5 nodes. **Fig 8a** and **8b** shows the rank of each node that will act as the input to FFBPNN. **Fig 8b** shows the rank assigned to each node from 1 to 60 that was involved in forwarding the data to destined destination.





```
networkoutcome =

  114×6 cell array

  Columns 1 through 5

    {'Route'              }    {'Throughput p/s '}    {'PDR'    }    {'Drop Ratio'}    {'Delay in ms'}
    {[    58 21 53 15 11]}    {[        194.0607]}    {[0.0504]}    {[    0.9496]}    {[     0.8624]}
    {[       55 53 21 9]}    {[      1.1053e+03]}    {[0.0706]}    {[    0.9294]}    {[     0.5210]}
    {[       1 53 21 58]}    {[         35.0786]}    {[0.0594]}    {[    0.9406]}    {[     0.5571]}
    {[    57 21 53 15 40]}    {[         34.3675]}    {[0.0465]}    {[    0.9535]}    {[     1.0852]}
    {[      33 15 16 3 4]}    {[         51.3206]}    {[0.0534]}    {[    0.9466]}    {[     1.2225]}
    {[    56 28 16 40 50]}    {[         83.6471]}    {[0.0398]}    {[    0.9602]}    {[     0.7160]}
    {[       58 58 53 25]}    {[        907.1561]}    {[0.0809]}    {[    0.9191]}    {[     0.8747]}
    {[    57 49 53 40 15]}    {[        163.8789]}    {[0.0450]}    {[    0.9550]}    {[     0.7110]}
    {[       20 40 16 9]}    {[        104.5837]}    {[0.0389]}    {[    0.9611]}    {[     0.4858]}
    {[      3 28 16 58 4]}    {[         15.2864]}    {[0.0571]}    {[    0.9429]}    {[     0.9847]}
    {[       40 40 16 7]}    {[        166.7501]}    {[0.0297]}    {[    0.9703]}    {[     0.6074]}
    {[       40 40 16 9]}    {[        312.5936]}    {[0.0330]}    {[    0.9670]}    {[     0.6087]}
    {[    18 16 40 28 45]}    {[         40.8316]}    {[0.0303]}    {[    0.9697]}    {[     1.1506]}
    {[    37 58 28 16 32]}    {[          6.8436]}    {[0.0427]}    {[    0.9573]}    {[     1.3267]}
    {[19 49 53 40 16 13]}    {[        316.2310]}    {[0.0571]}    {[    0.9429]}    {[     1.6654]}
    {[    22 40 16 53 46]}    {[         39.9660]}    {[0.0388]}    {[    0.9612]}    {[     1.5372]}
    {[    16 16 40 49 26]}    {[        246.3790]}    {[0.0506]}    {[    0.9494]}    {[     1.6793]}
    {[44 49 53 40 16 32]}    {[        237.1933]}    {[0.0501]}    {[    0.9499]}    {[     1.9827]}
    {          1×7 double}    {[        164.5657]}    {[0.0493]}    {[    0.9507]}    {[     2.7044]}
```

Fig 8. (a) Ranking Score for Each Node in Route. **Fig 8. (b)** Ranking Score for Each Node in Route.

4. During the past 10-15 years, routing in WSN has been a significant area of study. In WSN, energy is a crucial resource for extending network lifetime Numerous studies have been conducted in this area, but energy-efficient data collection has not improved.

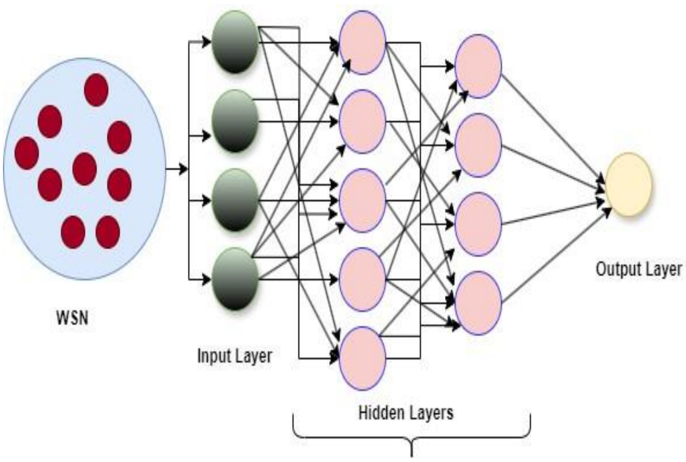


Fig 9. (a) Construction of Deep Feed Forward Neural Learning [30].

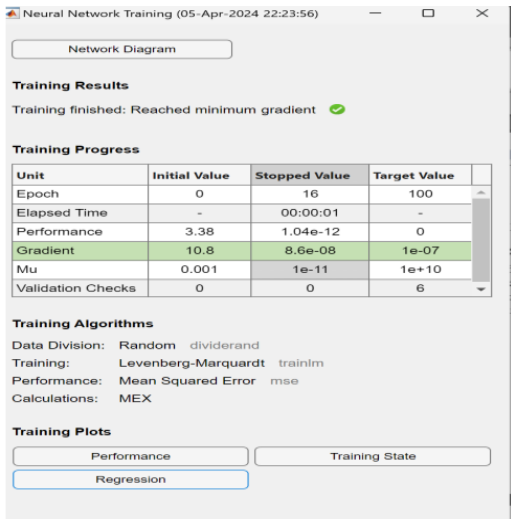


Fig 9. (b) Construction of Deep Feed Forward Neural Learning from Matlab Simulation and Parameters Considered for Same.

In WSNs, multipath routing is a novel technique for data packet routing. Through the exploitation of available network resources, the multipath routing strategy improved network performance. In **Fig 9a** and **9b**, this step Feed Forward Backpropagation Model based neural network (FFBPNN) technique [27] is used to do multipath routing. In this training model total four layers are used which is made up of an input layer, two hidden layers, and an output layer for data aggregation in WSN [28]. In our proposed methodology the input layer consists of 5 features which are collected from the previous steps. There are two hidden layers involved in our proposed model. The features will randomly hit the neurons involved in hidden layer and activates it. Neurons can be hit two ways like linear and polynomial. We will be using polynomial activation function in our model. The model will be trained using multiple epochs. We will use epoch =100. The simulation process will continue till the time convergence is achieved. The point where convergence happens is called gradient.

V. RESULTS AND ANALYSIS

In this section, the result analysis of Improved route discovery and aggregation is discussed with traditional route discovery and data aggregation process. The parameters considered for same are energy consumption, PDR, delay and throughput.

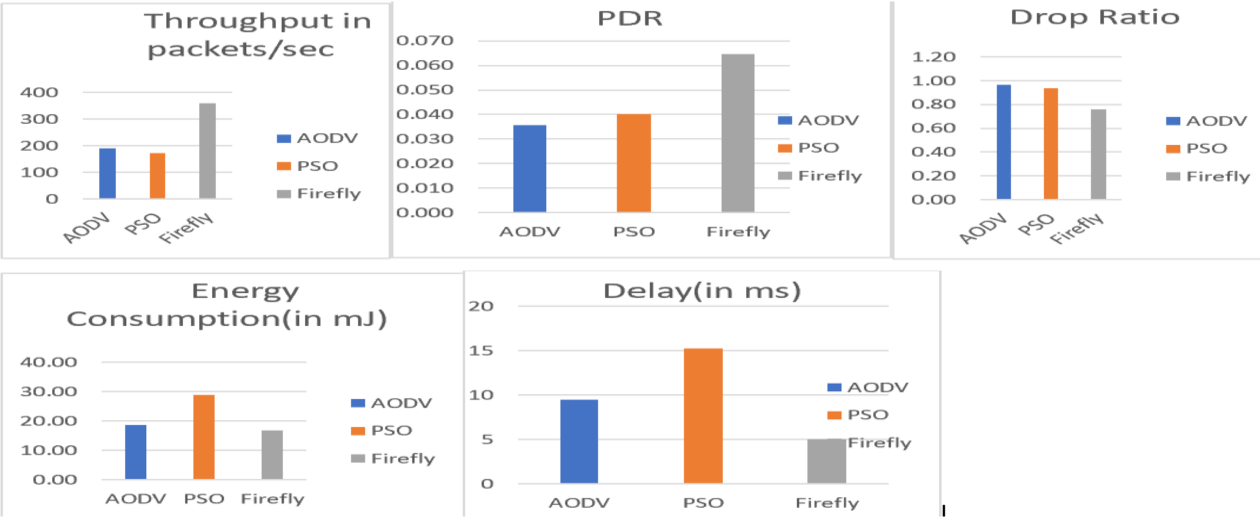


Fig 10. QoS parameter comparison between AODV, PSO and Optimized firefly.

The mentioned graphs in **Fig 10** are depicting the comparative analysis of QoS parameters between AODV, PSO and Optimized Firefly algorithm **Fig 10** showcases that the overall performance of metaheuristic approach is better than that of the other approaches like PSO and traditional method of route discovery and data aggregation.

VI. CONCLUSION AND FUTURE SCOPE

This research implemented the route discovery and data aggregation using the traditional method and collected various QoS parameters. The same environment was simulated using the metaheuristic approach and QoS parameters were again collected from the given setup. Those collected parameters were then passed on to the FFBPNN using the Levenberg-Marquardt training model. With the help of machine learning based model, the route and route nodes were selected on the basis of rank assigned to them by training model. The experimental result illustrates that the metaheuristic based FFBPNN based Model provided better performance with an improvement of packet delivery ratio and minimization of delay along with reducing the energy consumption when compared to state-of-the-art work. In future work, this strategy can be applied to heterogenous networks as well. The security feature can also be incorporated in the given architecture.

Data Availability
No data was used to support this study.

Conflicts of Interests
The author(s) declare(s) that they have no conflicts of interest.

Funding
No funding agency is associated with this research.

Competing Interests
There are no competing interests.

References
[1]. N. Kaur and D. Vetrithangam, “Routing and Data Aggregation Techniques in Wireless Sensor Networks: Previous Research and Future Scope,” Data Science and Communication, pp. 705–718, 2024, doi: 10.1007/978-981-99-5435-3_51.
[2]. Abd El-Kader, S. M., "Performance evaluation for flat and hierarchical WSN routing protocols", Mediterr J Comput Networks United Kingdom, (2011), 237-43.
[3]. Md. A. Razzaque, C. S. Hong, and S. Lee, “Data-Centric Multiobjective QoS-Aware Routing Protocol for Body Sensor Networks,” Sensors, vol. 11, no. 1, pp. 917–937, Jan. 2011, doi: 10.3390/s110100917.
[4]. Fu, G. Fortino, P. Pace, G. Aloi, and W. Li, “Environment-fusion multipath routing protocol for wireless sensor networks,” Information Fusion, vol. 53, pp. 4–19, Jan. 2020, doi: 10.1016/j.inffus.2019.06.001.
[5]. T. M. Behera et al., “Energy-Efficient Routing Protocols for Wireless Sensor Networks: Architectures, Strategies, and Performance,” Electronics, vol. 11, no. 15, p. 2282, Jul. 2022, doi: 10.3390/electronics11152282.
[6]. S. S. Sharifi and H. Barati, “A method for routing and data aggregating in cluster-based wireless sensor networks,” Int. J. Commun. Syst., vol. 34, no. 7, p. e4754, May 2021, doi: 10.1002/DAC.4754.
[7]. S. Kumar and S. Kumar, “Data aggregation using spatial and temporal data correlation,” 2015 International Conference on Futuristic Trends on Computational Analysis and Knowledge Management (ABLAZE), Feb. 2015, doi: 10.1109/ablaze.2015.7155043.
[8]. L. Dash et al., “A Data Aggregation Approach Exploiting Spatial and Temporal Correlation among Sensor Data in Wireless Sensor Networks,” Electronics, vol. 11, no. 7, p. 989, Mar. 2022, doi: 10.3390/electronics11070989.
[9]. Olshevsky, Alex., "Efficient information aggregation strategies for distributed control and signal processing.", arXiv preprint arXiv, (2010) :1009.6036.

[10]. Y. P. Chen, A. L. Liestman, and J. Liu, “A Hierarchical Energy-Efficient Framework for Data Aggregation in Wireless Sensor Networks,” IEEE Transactions on Vehicular Technology, vol. 55, no. 3, pp. 789–796, May 2006, doi: 10.1109/tvt.2006.873841
[11]. M. Kaur and A. Munjal, “Data aggregation algorithms for wireless sensor network: A review,” Ad Hoc Networks, vol. 100, p. 102083, Apr. 2020, doi: 10.1016/j.adhoc.2020.102083.
[12]. N.-T. Nguyen, B.-H. Liu, V.-T. Pham, and Y.-S. Luo, “On maximizing the lifetime for data aggregation in wireless sensor networks using virtual data aggregation trees,” Computer Networks, vol. 105, pp. 99–110, Aug. 2016, doi: 10.1016/j.comnet.2016.05.022.
[13]. E. G. Prathima, T. S. Prakash, K. R. Venugopal, S. S. Iyengar, and L. M. Patnaik, “SDAMQ: Secure Data Aggregation for Multiple Queries in Wireless Sensor Networks,” Procedia Computer Science, vol. 89, pp. 283–292, 2016, doi: 10.1016/j.procs.2016.06.060.
[14]. T. Wang, X. Qin, Y. Ding, L. Liu, and Y. Luo, “Privacy-Preserving and Energy-Efficient Continuous Data Aggregation Algorithm in Wireless Sensor Networks,” Wireless Personal Communications, vol. 98, no. 1, pp. 665–684, Aug. 2017, doi: 10.1007/s11277-017-4889-5.
[15]. K. Idrees, A. K. M. Al-Qurabat, C. A. Jaoude, and W. Laftah Al-Yaseen, “Integrated Divide and Conquer with Enhanced k-means technique for Energy-saving Data Aggregation in Wireless Sensor Networks,” 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC), Jun. 2019, doi: 10.1109/iwcmc.2019.8766784.
[16]. J. Zhang, Z. Lin, P.-W. Tsai, and L. Xu, “Entropy-driven data aggregation method for energy-efficient wireless sensor networks,” Information Fusion, vol. 56, pp. 103–113, Apr. 2020, doi: 10.1016/j.inffus.2019.10.008.
[17]. R. Sharma, V. Vashisht, and U. Singh, “Modelling and simulation frameworks for wireless sensor networks: a comparative study,” IET Wireless Sensor Systems, vol. 10, no. 5, pp. 181–197, Oct. 2020, doi: 10.1049/iet-wss.2020.0046.
[18]. Paul, Subhra Prosun, and D. Vetrithangam., "Design and Analysis of an Efficient and Load-Balanced Multipath Routing Algorithm for Energy-Effective Wireless Sensor Networks",." International Journal of Intelligent Systems and Applications in Engineering 11.10s, (2023): 601-617.
[19]. M. V. Babu, J. A. Alzubi, R. Sekaran, R. Patan, M. Ramachandran, and D. Gupta, “An Improved IDAF-FIT Clustering Based ASLPP-RR Routing with Secure Data Aggregation in Wireless Sensor Network,” Mobile Networks and Applications, vol. 26, no. 3, pp. 1059–1067, Oct. 2020, doi: 10.1007/s11036-020-01664-7.
[20]. T. Pham, Eun Jik Kim, and M. Moh, “On data aggregation quality and energy efficiency of wireless sensor network protocols - extended summary,” First International Conference on Broadband Networks, doi: 10.1109/broadnets.2004.51.
[21]. N. John, N. Joseph, N. Manuel, S. Emmanuel, and S. Kurian, “Energy efficient data aggregation and improved prediction in cooperative surveillance system through Machine Learning and Particle Swarm based Optimization,” EAI Endorsed Transactions on Energy Web, p. 170014, Jul. 2018, doi: 10.4108/eai.3-6-2021.170014.
[22]. R. Jatothu, S. Sira Jacob, S. Sulaiman Hamid, A. Kumar Saini, D. Singh, and D. Kapila, “Data Aggregation of Wireless Sensor Network Using BEE Swarm Optimisation Technique,” 2022 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES), Jul. 2022, doi: 10.1109/icses55317.2022.9914242.
[23]. S. Sharmin, I. Ahmedy, and R. Md Noor, “An Energy-Efficient Data Aggregation Clustering Algorithm for Wireless Sensor Networks Using Hybrid PSO,” Energies, vol. 16, no. 5, p. 2487, Mar. 2023, doi: 10.3390/en16052487.
[24]. N. Mazloomi, M. Gholipour, and A. Zaretalab, “Efficient configuration for multi-objective QoS optimization in wireless sensor network,” Ad Hoc Networks, vol. 125, p. 102730, Feb. 2022, doi: 10.1016/j.adhoc.2021.102730.
[25]. S. El Khediri, W. Fakhet, T. Moulahi, R. Khan, A. Thaljaoui, and A. Kachouri, “Improved node localization using K-means clustering for Wireless Sensor Networks,” Computer Science Review, vol. 37, p. 100284, Aug. 2020, doi: 10.1016/j.cosrev.2020.100284.
[26]. S. Radouche and C. Leghris, “New Network Selection Algorithm Based on Cosine Similarity Distance and PSO in Heterogeneous Wireless Networks,” Journal of Computer Networks and Communications, vol. 2021, pp. 1–11, Oct. 2021, doi: 10.1155/2021/6613460.
[27]. T. Kim, L. F. Vecchietti, K. Choi, S. Lee, and D. Har, “Machine Learning for Advanced Wireless Sensor Networks: A Review,” IEEE Sensors Journal, vol. 21, no. 11, pp. 12379–12397, Jun. 2021, doi: 10.1109/jsen.2020.3035846.
[28]. Saravanaselvan and B. Paramasivan, “FFBP Neural Network Optimized with Woodpecker Mating Algorithm for Dynamic Cluster-based Secure Routing in WSN,” IETE Journal of Research, pp. 1–10, Feb. 2024, doi: 10.1080/03772063.2023.2300349.
[29]. S. Mostafavi and V. Hakami, “A new rank-order clustering algorithm for prolonging the lifetime of wireless sensor networks,” International Journal of Communication Systems, vol. 33, no. 7, Jan. 2020, doi: 10.1002/dac.4313.
[30]. M. Revanesh, S. S. Gundal, J. R. Arunkumar, P. J. Josephson, S. Suhasini, and T. K. Devi, “Artificial neural networks-based improved Levenberg–Marquardt neural network for energy efficiency and anomaly detection in WSN,” Wireless Networks, Mar. 2023, doi: 10.1007/s11276-023-03297-6.
[31]. V. L. Narla, R. Kachhoria, M. Arun, A. Haldorai, D. Vijendra Babu, and B. M. Jos, “IoT based energy efficient multipath power control for underwater sensor network,” International Journal of System Assurance Engineering and Management, Apr. 2022, doi: 10.1007/s13198-021-01560-7.
[32]. Paul, Subhra Prosun, and D. Vetrithangam., "Design and Analysis of an Efficient and Load-Balanced Multipath Routing Algorithm for Energy-Effective Wireless Sensor Networks", International Journal of Intelligent Systems and Applications in Engineering 11.10s,601-617, (2023).
[33]. S. P. Paul and D.Vetrithangam, “A Comprehensive Analysis on Issues and Challenges of Wireless Sensor Network Communication in Commercial Applications,” 2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), Nov. 2022, doi: 10.1109/icccis56430.2022.10037743.