

K-Means Based Cluster Formation and Head Selection Through Artificial Neural Network in MANET

Dhananjay Bisen (✉ bisen.it2007@gmail.com)

Rajkiya Engineering College

Sneha Mishra

University Institute of Technology

Praneet Saurabh

Mody University of Science and Technology

Research Article

Keywords: MANET, Cluster head, k-means, ANN, Stability time.

Posted Date: July 13th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-667651/v1>

License:  This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

K-means based Cluster Formation and Head Selection through Artificial Neural Network in MANET

Dhananjay Bisen¹, Sneha Mishra², Praneet Saurabh³

Rajkiya Engineering College, Banda, UP, India¹
University Institute of Technology, RGPV, Bhopal, MP, India²
Mody University of Science and Technology, Lakshmangarh, Rajasthan, India³

bisen.it2007@gmail.com¹, snehamishra2506@gmail.com², praneetsaurabh@gmail.com³

Abstract: Mobile ad-hoc network is a dynamic and self configuring network composed of mobile nodes that are cooperative and intelligent in nature, which forms a temporary network without any base station. Due to dynamic nature, route among source and destination is not fixed and it can change with time, this results in routing overhead, congestion and higher energy consumption in nodes. In this condition, clustering becomes very relevant in highly dense network in which the clusterhead initiates the routing instead of normal nodes. This paper proposes an efficient way for cluster formation and selection of stable cluster heads (E-CFSA) with high energy level. E-CFSA forms node cluster using k-means algorithm where distance with centroids acts as the key parameter. Afterwards, artificial neural network (ANN) is applied in each cluster to select efficient clusterhead. It also updates weight of input parameters such as mobility, packet drop, energy and number of neighbor nodes in order to minimize the errors at target neuron using back propagation algorithm. The overall process identifies and selects the nodes as clusterheads that possess longer stability and higher energy level. This procedure reduces the repetitive selection of cluster heads and re-affiliation of member nodes in a cluster. Thereafter, performance is evaluated in terms of overhead, packet delivery ratio, throughput and cluster head stability time with variation in speed of node or number of nodes in MANET. The experimental results obtained through proposed E-CFSA are compared to other existing mechanisms and a comparison is drawn. These comparative results very clearly revealed that proposed E-CFSA outperformed its peers.

Keywords: MANET, Cluster head, k-means, ANN, Stability time.

1. Introduction

Mobile ad-hoc network (MANET) is a collection of nodes that can communicate with each other and move freely. It is a self directed infrastructure less network where more than two mobile nodes can share data. The working of network has shown in figure 1 in which source mobile

node (S) can send data to destination mobile node (D), directly whenever they are in the coverage of transmission range. These can directly communicate to each other otherwise intermediate mobile nodes (like A, B and C) relay data from source to destination. In MANET, a mobile node can join and leave the network dynamically as the physical network changes frequently. Also, node movement differ for mobile nodes, operation topology also depend on the speed and the direction of the host [1]. Advantages like instant connectivity, router free, fault tolerance in case of connection failures and economical make MANET more enticing than traditional network. All these advantages with free mobility, without the limitation of localization and enabling communication without bothering about the details makes MANET realize pervasive and ubiquitous computing like ambient intelligence. The network routing in MANET is a challenging task due to dynamic nature of the topology. There exists many routing protocols for MANETs and it's classification are as follows: Proactive or Table driven routing protocols, Reactive or On demand routing protocols and Hybrid routing protocols [2].

Ad hoc On-demand distance vector (AODV) routing protocol is a single path routing protocol between the source and the destination. It tends to select single minimum-cost path since although different path with different cost metrics exist. Propagation of data to destination through a single path is not sufficient, therefore multipath routing comes into existence to overcome this limitation. In the multipath technique the best path is selected out of several available paths to transfer the data packets. If the path selected encounters complexity or error occurred or remains busy during the data transfer then alternate best available path is chosen for data transfer.

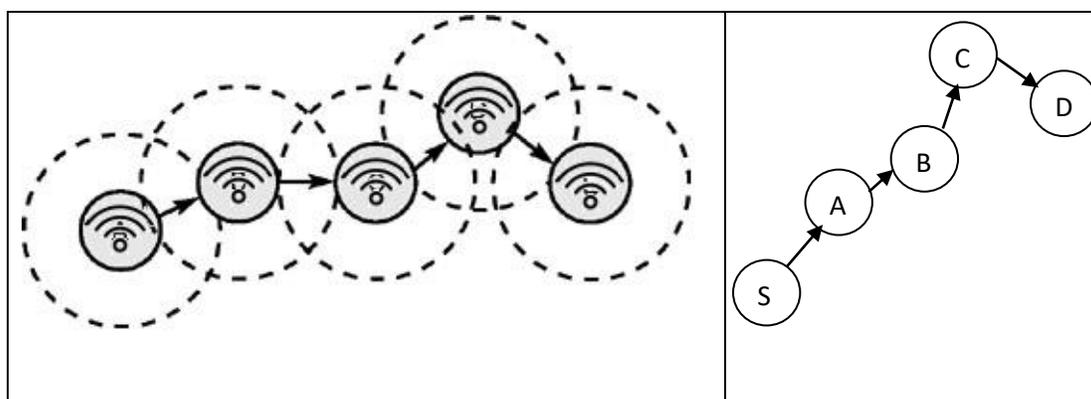


Figure: 1 Mobile ad-hoc network

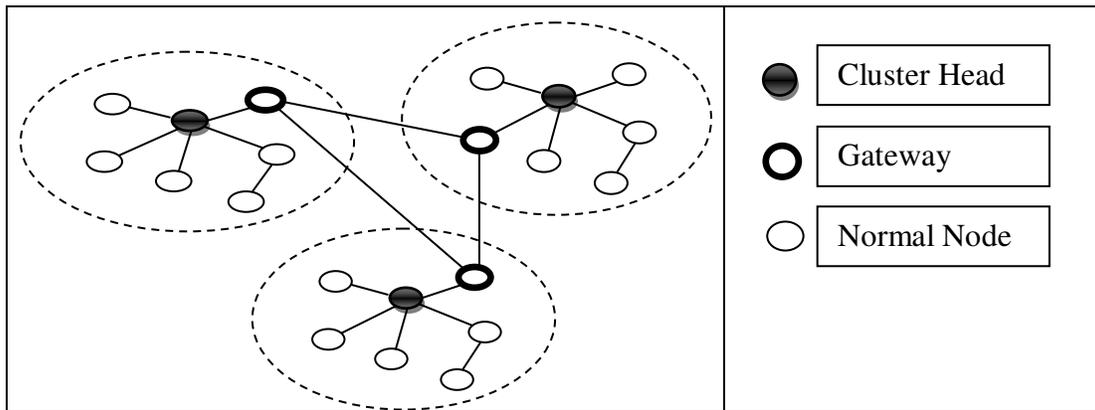


Figure: 2 Clustering in MANETs

In MANET, when source nodes want to communicate with destination node, it broadcasts or receives control messages like route request, route reply to connected neighbor nodes. This activity is continued till it finds a route to the destination node. In the scenarios, when network has excess number of nodes or network is highly dense, then existing routing approach generates high control overhead. Small group of nodes are created in the form of clusters to overcome the increased overhead, where routing is initiated through clusterhead, in place of normal nodes. Finally, routes are formed and data transfer will take place through cluster node instead of communication to individual nodes, represented as figure 2. The mobile nodes are aggregated into clusters and one of them is elected as a cluster head on the basis of its performance. Nodes that are in the communication range of the clusterhead belong to its cluster. In a dynamic network, clusterhead scheme can cause performance degradation due to frequent clusterhead selections. In recent times there are various explores involving cluster based routings like least cluster change, in which clusterhead get changed in the condition that mapped two clusterheads to come into one cluster [3]. Several techniques have been expressed to select a trusted clusterhead like link cluster architecture based lower-ID, highest degree techniques, node performance based single metric and multiple metric techniques [4]. Some algorithm are based on neural networks in which feed forward and multi layer perceptron is used to optimize the route and select cluster head which can overcome the issues and improve the usability and survivability of network [5].

The key objective of this work is seamless clusterhead selection for the challenging scenarios where nodes are constantly changing its position along with the other nodes in MANET. The clusterhead node must have best credentials in terms of their throughput, energy etc. Also, it does coordination among the members of each cluster. On the basis of these characteristics the nodes of cluster are analyzed and the clusterhead is selected. Another issue

like due to dynamic nature of MANET, number of members in the cluster can increase suddenly, which increases the size of the cluster that causes overloading. Now, handling of all these nodes becomes challenging and as a result performance suffers. This can lead to failure in the transmission of data among the clusters. Another challenge exist in the scenarios of highly dynamic cluster where mobile nodes acting as a clusterhead changes its position so much that it actually leaves the cluster then new clusterheads has to be identified and put in place so that it can manage and look after the cluster. The proposed coupling of K-means algorithm with ANN successfully achieves this aim and also works fine in the context of scalability in MANET. Taking all the challenges into account, this paper strives to find an optimal cluster head, reduce the scenario of overloading within in cluster and making new cluster head seamlessly.

The rest of the paper is organized as follows. Section 2 covers the related works in the area of cluster head selection algorithms in MANET. Section 3 represents the proposed approach of cluster head selection and weight updation (E-CFSA). Section 4 presents the experimental result and analysis with comparison from current state of art while section 5 concludes this paper.

2. Related work

In the last decade, numerous algorithms have been introduced to design efficient concept for clustering of nodes, mitigating the problem of network overhead and effective selection of cluster head. Torkestani and Meybodi [6] proposed fully distributed MCFA cluster algorithm based on learning automata and mobility parameters of host. Moreover, performance of MCFA analyzed in terms of the number of clusters, lifetime of cluster, reaffiliation rate and control overhead. Furthermore, Cheng et al. [7] proposed dynamic genetic algorithm for solving problem of load balanced clustering. Genetic algorithms show clustering structure based on load balancing metric with its fitness value as nodes change their position from one place to another frequently. During this time, Agarwal et al. [4], reviewed weight based clustering algorithms introduced where cluster head was identified on the basis of nodes performance. Relatively, adaptive cluster based energy aware routing [8] was proposed that considered node position and power saving parameters to balance the energy of node as well as improve network life. In addition, this protocol worked on such parameters like loading of network, energy conservation and lifetime of nodes and networks. Subsequently, battery power and connectivity level based cluster head technique was introduced for selecting cluster head in MANET [9]. Afterwards, one hop clustering algorithm was proposed by Basurra [10] in that introduced zone based routing

with parallel and distributed broadcasting technique to minimize redundant broadcasting and improve path discovery process. Review of different kind of routing has already done in article [13].

In other works, Periyasamy et al [11] employed selection of clusterhead with modified K-means while Prince and Kannan [12] used bat-inspired strategy, and introduced optimal clustering algorithms. Subsequently, other authors focused on work related to optimal cluster head selection in current years and discussed the issues related to reduce network overhead and improve security within the network after effective selection of cluster head. Agarwal et al. [14] presented trusted weight based clustering approach and used parameters like trust, load balancing, energy consumption, mobility and battery consumption. Bisen and Sharma [15] proposed agent based secure enhanced performance approach for MANET (AB-SEP). The key advantage highlighted included increase in the performance of nodes, lesser energy consumption and also identifies malicious activities of network [16]. Furthermore, neural network based adaptive neuro-fuzzy inference system [17] and multilayer perceptron neural network [5] was introduced for selecting cluster head on the basis of mobility and residual energy that improved the performance of network. In another explore Malar et al. [18] have introduced bio inspired methods to improve routing in MANET. In their work they have used Ant colony optimization technique for developing an energy efficient routing in MANET and utilized the energy in a smarter way. On the basis of above available state of the art, this paper considers problem of cluster formation, cluster head selection within network as key challenges and employs k-means algorithm with artificial neural network to find a solution.

3. Efficient Cluster formation and Head selection algorithm (E-CFSA)

This section presents an Efficient Cluster formation and Head selection algorithm (E-CFSA) that uses k-means algorithm and artificial neural network. In this work, mobile nodes are grouped together into clusters on the basis of distance between nodes, it is calculated by finding out the X-coordinate and Y-coordinate of the nodes. Distance is compared with all the neighbors or other nodes. Nodes having least difference of distance are grouped into one cluster and likewise the clusters are formed. E-CFSA use k-means algorithm [10] for clustering of nodes, it is a method of vector quantization, and it partitions 'n' observations into 'k' clusters in which each observation belongs to the cluster with the nearest mean serving as a prototype of cluster, hence 'k' here represents number of clusters and mean is the calculated average value of

parameters. [11]. After clustering, the most efficient node within the cluster is selected as cluster head. The cluster head is selected on the basis of parameters such as energy, mobility, number of hops and neighbors. There are various constraints in selection of cluster head and these worked out using several routing protocols such as clusterhead gateway switch routing protocol (CGSR) [19]. Cluster formation procedure is given below in detail.

3.1 Cluster formation procedure:

- (i) Initially assume the number of centroids according to the value of k (i.e. number of clusters to be formed).

For example if the value of “k” is 2 then two nodes are selected randomly as centroid. For the calculation of ‘n’ dimensional centroid point among k n-dimensional point following equation (1) is used:

$$CP(X_1, X_2, X_3 \dots \dots X_n) = \left(\frac{\sum_{K=1}^n X_{1st}}{k}, \frac{\sum_{K=1}^n X_{2nd}}{k} \dots \dots \dots, \frac{\sum_{K=1}^n X_{nth}}{k} \right) \quad (1)$$

From these centroids, the distance of node is calculated and compared with that of the neighbor or other nodes. This distance is called as Euclidean distance and is calculated using following formula illustrated in equation (2) below:

$$d(i, j) = \sqrt{|X_{i_1} - X_{j_1}|^2 + |X_{i_2} - X_{j_2}|^2 + |X_{i_3} - X_{j_3}|^2 + \dots + |X_{i_n} - X_{j_n}|^2} \quad (2)$$

- (ii) After comparing the distance among nodes, the nodes with least difference of distance are clustered together.
- (iii) Now, calculate mean of the distances and again takes difference of the distance and mean value. Accordingly, iteratively new cluster is formed and the same step is repeated whenever get the common clusters. Finally K numbers of groups are formed.
- (iv) Suppose the distances (D) between nodes are D= {60, 56, 40, 45, 10, 5, 80, 90, 0, 95} in network simulation and number of clusters in network to be created is 2, therefore, K=2. M₁ and M₂ are the mean values these are taken randomly from the set D, shown in table 1.

Table 1: Cluster formation table

| M1 | M2 | Cluster (K1) | Cluster (K2) | Mean of K1(M1) | Mean of K2(M2) |
|----|----|----------------------------|----------------------|----------------|----------------|
| 56 | 80 | {60, 56, 40, 45, 10, 5, 0} | {80, 90, 95} | 31 | 88 |
| 31 | 88 | {56, 40, 45, 10, 5, 0} | {60, 80, 90, 95} | 26 | 81 |
| 26 | 81 | {40, 45, 10, 5, 0} | {80, 90, 95, 60, 56} | 20 | 76 |
| 20 | 76 | {40, 45, 10, 5, 0} | {80, 90, 95, 60, 56} | 20 (Repeat) | 76 (Repeat) |

(v) Finally two clusters are formed as:

(a) Cluster1= {40, 45, 10, 5, 0}

(b) Cluster2= {60, 50, 80, 90, 95}

(vi) After clustering, ANN (artificial neural network) is applied within cluster for selecting efficient cluster head.

3.2 Cluster head selection Algorithm and weight updating algorithm

Artificial neural networks (ANNs) are computing systems inspired by biological neural networks that constitute human brains [20]. An ANN is collection of connected units or nodes called as artificial neurons (simplified version of biological neurons). Once, neurons are connected to each other they transmit signals between each other. Each neuron receives signals, processes it and transmits it to other next nearest neuron. This transmission of signals takes place only if the calculated value is greater than or equal to some threshold value. In ANN the signals are in the form of real numbers and output of each artificial neuron is calculated by a non-linear function as sum of its inputs. The basic model represented in figure 3:

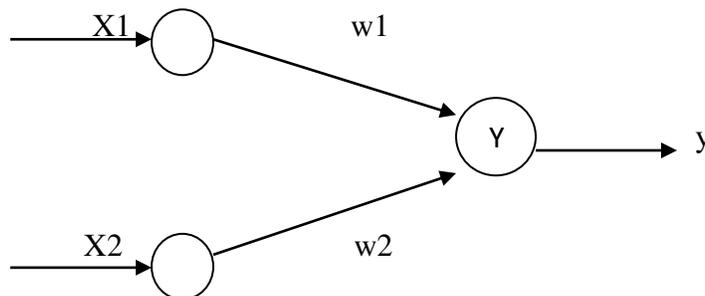


Figure 3: Simple artificial neuron net

Step1: After network simulation, four input parameters are taken into consideration such as:

- (i) Energy (E_N)
- (ii) Neighbor of node (node degree) (ND_N)
- (iii) Mobility (M_N)
- (iv) Packet drop (PD_N)

Step 2: Now weights are assigned to each parameters according to each artificial neurons. The change in value of these weights results into change in the strength of signals. The summation of these weights should be equal to 1. For selecting the cluster head, the input parameters should be weighted such that the final target value is maximum. According to these input parameters the

energy and neighbor nodes should be maximum while mobility and packet drop should be minimum.

Step 3: Consider the numbers of nodes are 10 as a sample data and weights for the input parameters are represented in equation (3):

$$W_1 (E_N) = 0.3, W_2 (ND_N) = 0.3, W_3 (M_N) = 0.2, W_4 (PD_N) = 0.2$$

$$\text{Here } W_1 + W_2 + W_3 + W_4 = 1 \quad (3)$$

Step 4: Value of input parameters for 10 nodes are illustrated in table 2, with simulation time of 100 sec. of network.

Table 2: Value of input parameters

| Parameters | N ₁ | N ₂ | N ₃ | N ₄ | N ₅ | N ₆ | N ₇ | N ₈ | N ₉ | N ₁₀ | MAX |
|-----------------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|-----------------|-------------|
| Energy(J) | 60 | 56 | 40 | 45 | 10 | 05 | 80 | 90 | 00 | 95 | 100 J |
| Node degree (nodes) | 6 | 4 | 5 | 8 | 6 | 2 | 3 | 9 | 9 | 8 | 10 nodes |
| Mobility(m/s) | 20 | 06 | 12 | 07 | 18 | 20 | 09 | 13 | 01 | 05 | 20 m/s |
| Packet drop (packets) | 500 | 550 | 400 | 300 | 100 | 300 | 120 | 150 | 10 | 245 | 550 packets |

Step 5: After normalizing the inputs (dividing parameters by maximum value) the new values are given in table 3:

Table 3: Normalized input values

| Parameters | N ₁ | N ₂ | N ₃ | N ₄ | N ₅ | N ₆ | N ₇ | N ₈ | N ₉ | N ₁₀ |
|-----------------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|-----------------|
| Energy(J) | 0.63 | 0.58 | 0.42 | 0.47 | 0.10 | 0.005 | 0.84 | 0.94 | 0.00 | 1 |
| Node degree (nodes) | 0.66 | 0.44 | 0.55 | 0.88 | 0.66 | 0.22 | 0.33 | 0.99 | 0.99 | 0.88 |
| Mobility(m/s) | 1 | 0.3 | 0.6 | 0.35 | 0.9 | 1 | 0.45 | 0.65 | 0.05 | 0.25 |
| Packet drop (packets) | 0.90 | 1 | 0.72 | 0.54 | 0.18 | 0.54 | 0.21 | 0.27 | 0.018 | 0.44 |
| Target value | 0.407 | 0.446 | 0.427 | 0.626 | 0.522 | 0.172 | 0.617 | 0.798 | 0.686 | 0.526 |

Step 6: Now calculate the target value [21] for given inputs using equation (4):

$$\text{Target value} = \text{Energy} * W_1 + \text{Node degree} * W_2 + (1 - \text{mobility}) * W_3 + (1 - \text{packet drop}) * W_4 \quad (4)$$

Step 7: So, target values for 10 nodes like $N_1=0.407$, $N_2=0.446$, $N_3=0.427$, $N_4=0.626$, $N_5=0.522$, $N_6=0.172$, $N_7=0.617$, $N_8=0.798$, $N_9=0.686$, $N_{10}=0.526$, included in table 2.

Step 8: Neural network works on the basis of parameters given to it and accordingly it trains the network and do optimization automatically. These parameters are processed into the nodes which are the artificial neurons and layer wise the output is forwarded for further processing of the input. The neural network comprises of three layers like input layer, hidden layer and output layer. Number of Hidden layers can be varied according to the optimization level needed. Each layer has nodes that are the artificial neurons. The nodes in input layer are taken according to the number of inputs given. Suppose there are four inputs then in the input layer four nodes will be taken. Basic training model has represented in figure 4.

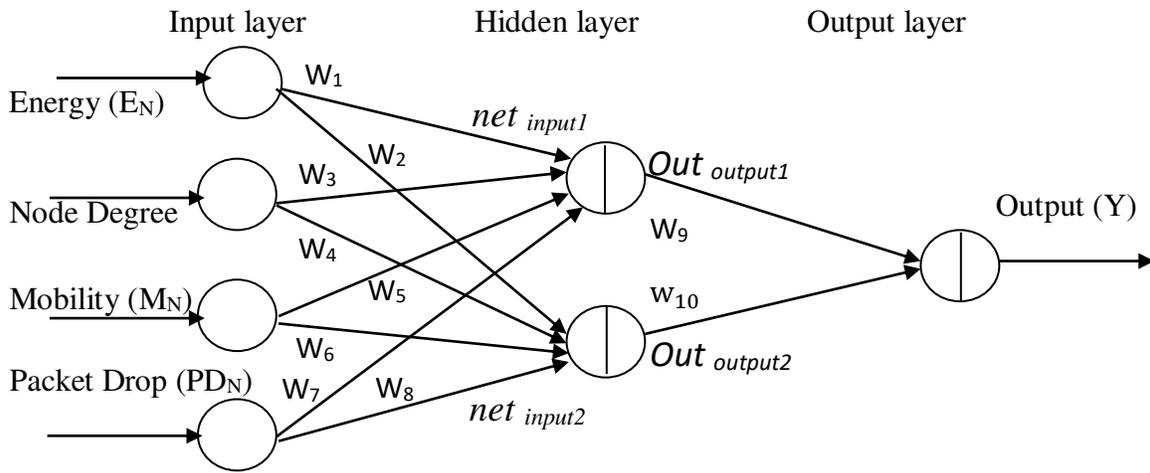


Figure 4: ANN Training Model

Step 9: The forward pass: After giving the inputs (x_i) and weights into input layer, neural network predicts some output. The total net input is calculated according to following formula given in equation (5) (6) and (7).

$$net_{input} = \sum w_i * x_i \quad (5)$$

$$net_{input1} = w_1 * E_{N_i} + w_3 * ND_{N_i} + w_5 * M_{N_i} + w_7 * PD_{N_i} \quad (6)$$

$$net_{input2} = w_2 * E_{N_i} + w_4 * ND_{N_i} + w_6 * M_{N_i} + w_8 * PD_{N_i} \quad (7)$$

Step 10: Then the input is squashed using the logistic function to get the output represented in equation (8) and (9).

$$out_{output1} = \frac{1}{1 + e^{-net_{input1}}} \quad (8)$$

$$out_{output2} = \frac{1}{1+e^{-net_{input2}}} \quad (9)$$

Step 11: This process is repeated for the output layer neurons, using the output from the hidden layer neurons as inputs.

Step 12: Calculating the total error: Now the error for each output neuron is calculated using squared error function and sum them to get the total error. Squared error function is given in equation (10).

$$E_{total} = \sum \frac{1}{2} (target - output)^2 \quad (10)$$

Step 13: Total error for this neural network is the sum of the calculated errors illustrated in equation (11).

$$E_{total} = E_{output1} + E_{output2} \quad (11)$$

$$E_{output1} = \frac{1}{2} (target_{o1} - out_{output1})^2$$

$$E_{output2} = \frac{1}{2} (target_{o2} - out_{output2})^2$$

Step 14: Substituting the values of $E_{output1}$ and $E_{output2}$ in equation (11), yield equation (12).

$$E_{total} = \frac{1}{2} (target_{o1} - out_{output1})^2 + \frac{1}{2} (target_{o2} - out_{output2})^2 \quad (12)$$

Step 15: The backward pass: After calculating the total error, back propagation is performed to update the weights in the network so that they cause the actual output to be closer to that of the target value, thereby minimizing error for each output neuron and the network as a whole. In back propagation, the partial derivative of E_{total} is taken with respect to the weights given. Here, considering weights w_1, w_2, w_3 and w_4 .

Step 16: By applying chain rule at output layer given in equation (13):

$$\frac{\partial E_{total}}{\partial w_9} = \frac{\partial E_{total}}{\partial E_{total}} * \frac{\partial out_{output1}}{\partial net_{input1}} * \frac{\partial net_{input1}}{\partial w_9} \quad (13)$$

Repeating same for weight w_{10} .

Step 17: Similarly, applying chain rule at hidden layer stated below:

$$\frac{\partial E_{total}}{\partial w_n} = \frac{\partial E_{total}}{\partial E_{total}} * \frac{\partial out_{output1}}{\partial net_{input1}} * \frac{\partial net_{input1}}{\partial w_n}$$

Repeating same for other weights at hidden layer where $n = (1, 2, \dots, 8)$

Step 18: Total error change with respect to the output is:

$$E_{total} = \frac{1}{2}(target_{o1} - out_{output1})^2 + \frac{1}{2}(target_{o2} - out_{output2})^2$$

Next, how much does the output (o1) change with respect to its total net input is calculated below.

$$out_{output1} = \frac{1}{1+e^{-net_{input1}}}$$

$$out_{output2} = \frac{1}{1+e^{-net_{input2}}}$$

After partially differentiating equation (13) with respect to $out_{output1}$, leads to equation (14).

$$\frac{\partial E_{total}}{\partial out_{output1}} = -(target_{o1} - out_{output1}) \quad (14)$$

Partially differentiating equation (8) with respect to net_{input1} , yields equation (15).

$$\frac{\partial out_{output1}}{\partial net_{input1}} = out_{output1}(1 - out_{output1}) \quad (15)$$

Total net input of output1 change with respect to weights-

Step 19: Partially differentiating equation (6) and (7) with respect to their respective weights, leads to equation (16)

$$\frac{\partial net_{input1}}{\partial w_n} = out_{output1} \quad (16)$$

Putting the values of equation (14), (15) and (16) into equation (10), the total error can be calculated.

For each weight, $\frac{\partial E_{total}}{\partial w_n}$ is calculated (n=1, 2,, 10).

Step 20: Next, to decrease the error, the calculated $\frac{\partial E_{total}}{\partial w_n}$ partial total is subtracted from the actual weight results in equation (17). (μ is learning rate)

$$\text{New weights are: } w_n^+ = w_n + \mu * \frac{\partial E_{total}}{\partial w_n} \quad (17)$$

In above algorithm, sigmoid function is used as activation function in hidden and output layer. On the basis of given input samples, neural networks trains itself and update weight according to calculate errors in output neuron using back propagation algorithm. Here, Adam optimizer is used instead of the classical stochastic gradient descent procedure, it calculates adaptive learning rate helping in faster convergence thereby resulting in reduced overall training cost and time. Whenever final output (Y) is approximately equal to the target value for the node, it is considered as error is minimum and model train correctly.

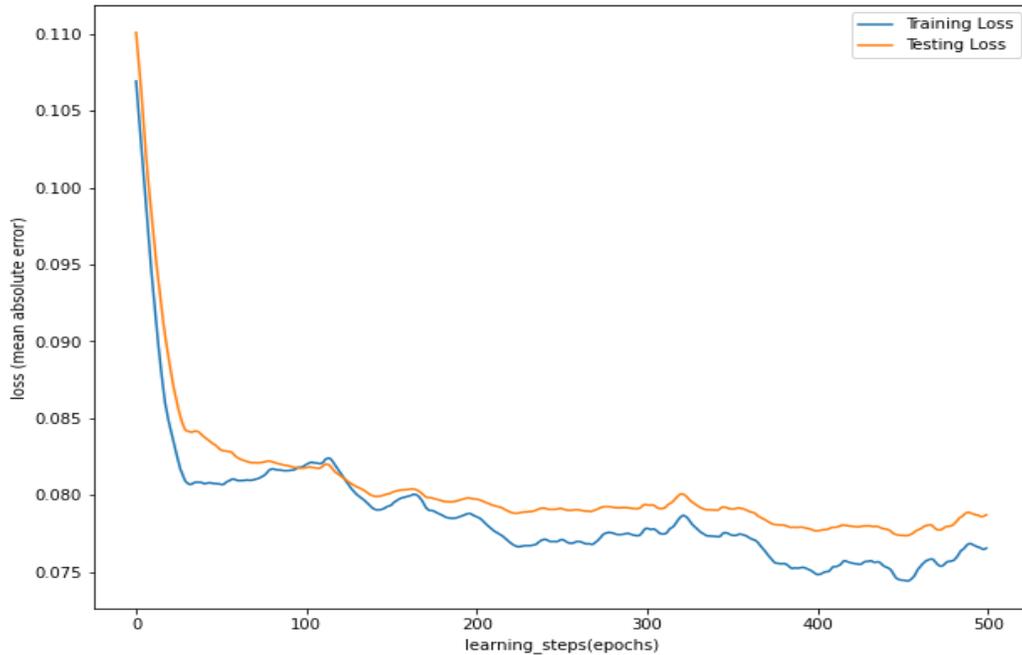


Figure 5: Learning curve

Figure 5 represents the learning curve or training vs testing error, this curve depicts the model's error for training and testing dataset respectively at each epoch (1 epoch = 1 pass over complete dataset). The testing error shows the model's robustness on unseen data. In the initial 50 epochs the model's loss reduced drastically which is a good sign. At the end the training and testing curve fluctuates a little due to small dataset. But, as the testing loss is very close to the training loss it is clear that the model is very good at predicting on unknown datasets. E-CFSA is trained for 500 epochs, during the initial 100 epochs the model was very fast at learning, while at the final epochs the curve of learning reduces drastically. The testing loss is slightly higher than the training loss because the model has never seen that data before and real life datasets are full of random noise which reduces the model accuracy. The considerable difference between the training loss and testing loss shows the model is not overfitting to the training dataset. After training this model is able to predict scores with accuracy of upto (+/-) 0.075. The model returns a score of the data points in the range of 0-1. Afterward, the node (sample) having maximum output value is taken as clusterhead node. Here, for this consideration, N8 node is selected as the clusterhead.

4. Simulation and Results Analysis

This section presents the experiments with proposed E-CFSA to observe and validate its findings, experimental results are then compared to conventional technique of weighted

clustering algorithm (WCA) [21] and Agent-based secure enhanced performance approach (AB-SEP) [15] keeping all the test conditions same. Experiments are performed on Python [22] and NS-2.35 [23]. Experiments are performed with 20-100 nodes, placed within 1000*1000 m² area for a simulation time is 200sec. Each node is configured using random way point mobility model with speed 0-25m/s, 250m transmission range and 200 joules initial energy. Data traffic generated through constant bit rate (CBR) UDP traffic sources and the size of data packet is 512 bytes. Table 4 shows vital simulation parameters and their values.

Table 4: Simulation parameters

| Parameters | Values |
|---------------------------|---|
| Number of nodes | 20,40,60,80,100 |
| Simulation time | 200s |
| Network dimensions | 1000*1000 m ² |
| Mobility type | Random way point |
| Node speed(Min-Max) | 0 m/s – 25m/s (randomly assign each node) |
| Initial NE | 0 – 200 joule (randomly assign each node) |
| Traffic type | CBR-UDP |
| CBR connections with load | 10 pairs with packet size 512 bytes |
| Channel type | Wireless interface |
| Radio propagation model | Two ray ground |

4.1 Performance parameters

This subsection details the parameters that measure the working of both conventional and proposed E-CFSA. Experimental are carried to find packet delivery ratio, throughput, routing or overhead and stability time period of cluster head [14] [15] to draw the comparison of E-CFSA with conventional method.

- **Packet delivery ratio (PDR)** – Packet delivery ratio is defined as the ratio of data packets received by the destinations to those generated by the sources. Mathematically, it can be defined as: $PDR = S_1 / S_2$; where, S_1 is the sum of data packets received by the each destination and S_2 is the sum of data packets generated by the each source given in equation (18)

$$PDR (\%) = \frac{\text{Number of packets received}}{\text{Number of packets sent}} * 100 \quad (18)$$

- **Throughput-** It is ratio of total number of packets delivered over the total simulation time as stated in equation (19).

$$\text{Throughput (kbps)} = \frac{\text{received packets(bytes)} * 8}{1024 * (\text{stop time} - \text{start time})} \quad (19)$$

- **Routing Overhead-** Network overhead is the number of control (hello packets) and routing packets required for an overall network communication illustrated in equation (20).

$$\text{Overhead (in ratio)} = \frac{\text{Total control and routing packets}}{\text{number of data packets received}} \quad (20)$$

- **Stability time period of cluster head-** Stability of a cluster head node is defined as the time period for which the node worked as a cluster head of the cluster. Average of that time period is known as average stability time.

4.2 Experimentation and result analysis

Experiments are performed to measure the performance of the proposed E-CFSA and then a comparison is done with current state of the arts. Experimental numerical statistics of E-CFSA, AB-SEP and WCA are presented in table 5 and 6.

Table 5: Effect of Network Size

| Varying Number of nodes | Packet Delivery Ratio (%) | | | Throughput (kbps) | | | Routing Overhead | | | Average Stability time of CH's (sec) | | |
|-------------------------|---------------------------|--------|-------|-------------------|--------|-----|------------------|--------|------|--------------------------------------|--------|-----|
| | E-CFSA | AB-SEP | WCA | E-CFSA | AB-SEP | WCA | E-CFSA | AB-SEP | WCA | E-CFSA | AB-SEP | WCA |
| 20 nodes | 84.32 | 82.5 | 75.68 | 126 | 110 | 98 | 0.4 | 0.4 | 0.6 | 22 | 15 | 10 |
| 40 nodes | 86.43 | 84.6 | 77.74 | 157 | 124 | 110 | 0.5 | 0.6 | 0.8 | 50 | 40 | 30 |
| 60 nodes | 89.01 | 85.0 | 69.02 | 176 | 135 | 138 | 0.62 | 0.7 | 0.9 | 75 | 60 | 40 |
| 80 nodes | 92.15 | 87.8 | 69.87 | 220 | 180 | 149 | 0.82 | 0.9 | 0.98 | 88 | 75 | 55 |
| 100 nodes | 88.21 | 86.0 | 74.97 | 255 | 210 | 172 | 0.96 | 1.0 | 1.23 | 110 | 95 | 65 |

Table 6: Effect of Node speed

| Varying node speed (m/s) | Packet Delivery Ratio (%) | | | Throughput (kbps) | | | Packet loss rate (%) | | | Average Stability time of CH's (sec) | | |
|--------------------------|---------------------------|--------|-------|-------------------|--------|-------|----------------------|--------|-----|--------------------------------------|--------|-----|
| | E-CFSA | AB-SEP | WCA | E-CFSA | AB-SEP | WCA | E-CFSA | AB-SEP | WCA | E-CFSA | AB-SEP | WCA |
| 5 m/s | 94.23 | 89.2 | 80.21 | 115.3 | 100.5 | 95.31 | 12 | 16 | 20 | 40 | 35 | 25 |
| 10 m/s | 93.92 | 90.6 | 80.91 | 113.2 | 98.4 | 93.2 | 17 | 19 | 25 | 35 | 31 | 24 |

| | | | | | | | | | | | | |
|--------|-------|------|-------|-------|------|------|----|----|----|----|----|----|
| 15 m/s | 90.31 | 89.0 | 77.32 | 102.8 | 99.0 | 92.8 | 16 | 19 | 29 | 28 | 27 | 20 |
| 20 m/s | 89.06 | 87.0 | 70.01 | 100.8 | 98.0 | 85.8 | 22 | 24 | 29 | 25 | 24 | 21 |
| 25 m/s | 88.00 | 86.0 | 71.20 | 98.4 | 96.0 | 78.4 | 25 | 26 | 34 | 22 | 22 | 15 |

Figure 6 and 7 shows the experimental results of packet delivery ratio of the proposed E-CFSA compared with the standard WCA and AB-SEP, with network size (varying number of nodes) and node speed. It can be observed that the packet delivery ratio (PDR) of E-CFSA is progressively increase under almost all the variations of network size (20 to 100 nodes) and node speed (5m/s to 25m/s). These results stress the fact that new integrations are working fine and enabling E-CFSA to achieve high packet delivery ratio.

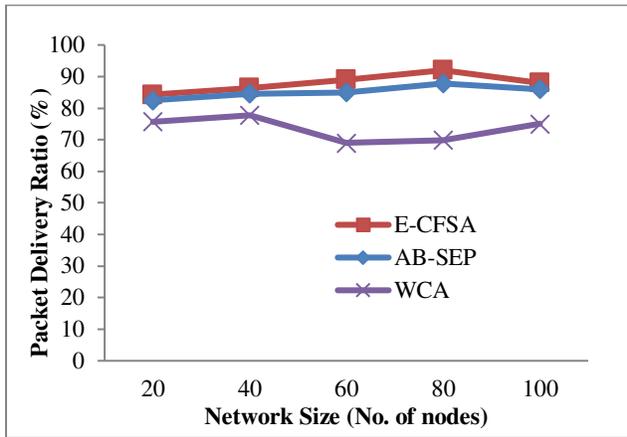


Figure 6: Packet delivery ratio (%) versus network size (No. of nodes)

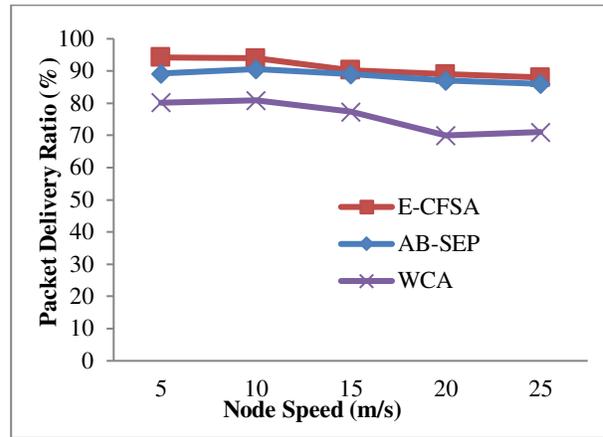


Figure 7: Packet delivery ratio (%) versus node speed (m/s)

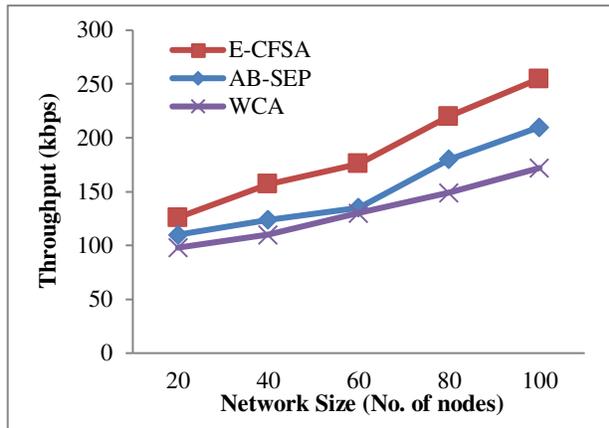


Figure 8: Throughput (kbps) versus network size (No. of nodes)

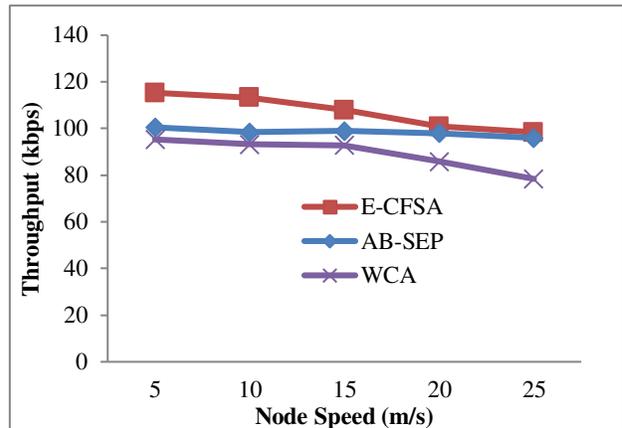


Figure 9: Throughput (kbps) versus node speed (m/s)

Figure 8 and figure 9 illustrates the throughput for proposed E-CFSA, WCA and AB-SEP with variation in number of nodes and node speed in the network. Experimental results in both the experiments and variations very clearly illustrates that E-CFSA clocks higher throughput under all the variations of number of nodes (20 to 100 nodes) and node speed (5m/s

to 25m/s) in the network. These results also highlight the fact that WCA does not calculate node reliability factor of the nodes and preferred shortest routes by default. Therefore, the network sizes are regularly varied in the routing process, which leads to an overall lower throughput.

Figure 10 and figure 11 presents the experimental findings of routing overhead with change in network size (number of nodes) and packet loss ratio with respect to variation in node speed respectively. These both experiments are very significant in measuring and comparing the performance of E-CFSA with conventional WCA and AB-SEP. Lower routing overhead and packet drop ratio is desired from any technique. In both the experiments proposed E-CFSA accounted lower routing overhead and packet drop ratio under all the variations in the test conditions.

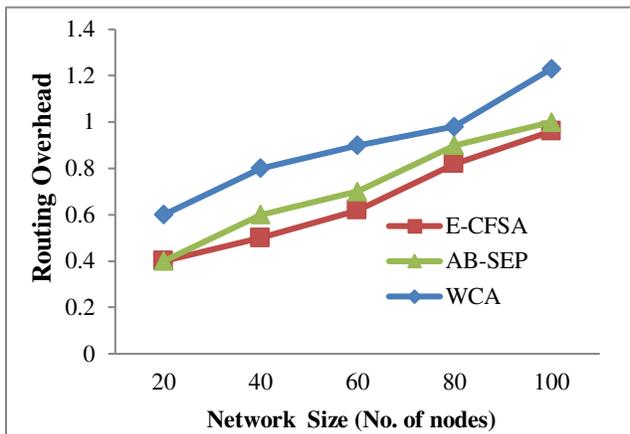


Figure 10: Routing or overhead versus network size (number of nodes)

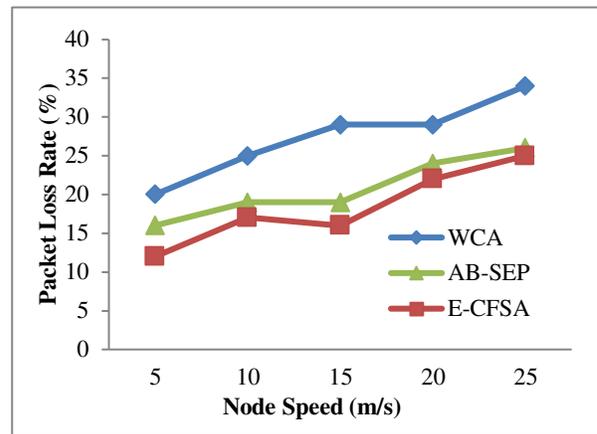


Figure 11: Packet loss ratio (%) versus node speed (m/s)

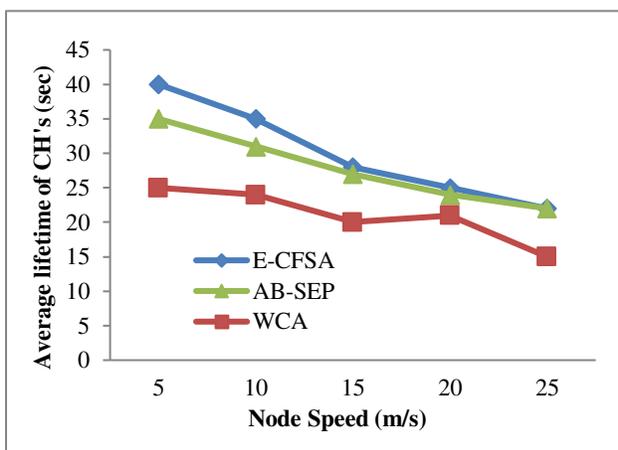


Figure 12: Average Stability Time of CH's (sec) versus node speed (m/s)

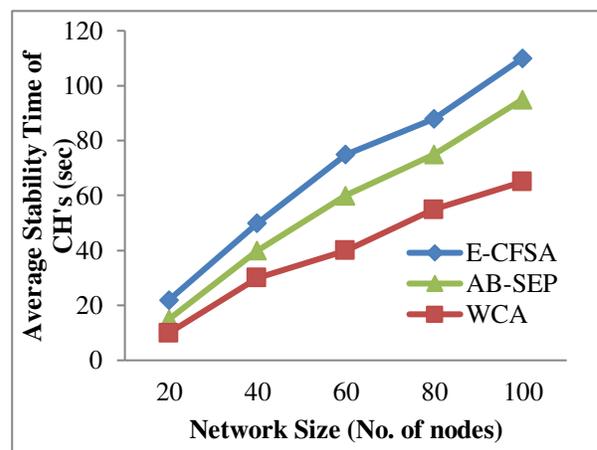


Figure 13: Average Stability Time of CH's (sec) versus no. of nodes

Figure 12 and 13 highlights the stability time period of the cluster head when the network size and node speed is varied for E-CFSA, WCA and AB-SEP. Experimental results very specifically highlights that E-CFSA based on k-means and ANN attain lower stability time under all the variations in node density and node speed of the network. The new integrations are helping E-CFSA to achieve better results as compared to conventional methods. These experimental results also signify that k-means is helping E-CFSA in cluster head selection which in turn helps in outperforming conventional state of the art.

5. Conclusion

This paper introduced efficient way for cluster formation and selection of stable cluster heads (E-CFSA) with high energy level among nodes in MANET. E-CFSA employs k-means algorithm to form cluster of nodes, based on distance among the nodes. Afterwards, artificial neural network is applied on each cluster for the selection of cluster head on the basis of mobility, packet drop, energy and number of neighbor nodes. E-CFSA identified nodes to be cluster heads that have longer stability and higher energy level. This procedure reduced the repetitive selection of cluster heads and re-affiliation of member nodes in a cluster. These new integrations in E-CFSA made it robust, make communication pervasive and achieved better packet delivery ratio, throughput and lower routing overhead and cluster head stability as compared to conventional WCA and AB-SEP. This work can be extended using bio inspired methods as they offer very enticing concepts, selection of cluster node can be considered as an intelligent and optimization problem for a more complex and heterogeneous networks.

Declaration

The Author(s) permits the publisher to publish this research article in this esteemed Journal. The Corresponding Author gives consent from individuals to publish the data associated with this research article.

Funding

The Author(s) declares that this research has not been funded by any agency.

Conflict of interest

The authors declare that they have no conflict of interest.

Availability of data and materials

All the data and the supplementary material can be made available from the corresponding author, upon reasonable request.

Code availability

Codes used in this study can be made available from the corresponding author, upon reasonable request.

Authors' contributions

- Dhananjay Bisen: Methodology; Writing original draft.
- Sneha Mishra: Literature Review; Editing; Reviewing the Manuscript.
- Praneeet Saurabh: Reviewing the Manuscript & final drafting.

Ethical Approval

All the author(s) agrees to the guidelines provided by the Journal. This research has considered all ethical issues associated with the methodology being employed.

Acknowledgements

The authors would like to extend our gratitude to Department of Computer Science & Engineering, Mody University of Science and Technology, Lakshmanagarh, Rajasthan and Department of Information Technology, University Institute of Technology, RGPV, Bhopal for support of this work.

References

1. Boukerche A, Turgut B, Aydin N, Ahmad MZ, Bölöni L, Turgut D (2011) Routing protocols in ad hoc networks: a survey. *Comput Netw* 55(13):3032–3080.
2. Hasdemir, S., Yilmaz, S. & Sen, S. A (2019) novel multi-featured metric for adaptive routing in mobile ad hoc networks. *Appl Intell* 49, 2823–2841. <https://doi.org/10.1007/s10489-018-01401-4>
3. Srungaram K., Krishna Prasad M.H.M. (2012) Enhanced Cluster Based Routing Protocol for MANETS. In: Meghanathan N., Chaki N., Nagamalai D. (eds) *Advances in Computer Science and Information Technology. Networks and Communications. CCSIT 2012. Lecture Notes of the Institute for Computer Sciences, Social Informatics and*

Telecommunications Engineering, vol 84. Springer, Berlin, Heidelberg.
https://doi.org/10.1007/978-3-642-27299-8_36.

4. Huang, J., Fan, X., Xiang, X., Wan, M., Zhuo, Z., Yang, Y. A (2016) Clustering Routing Protocol for Mobile Ad Hoc Networks, *Mathematical Problems in Engineering*, Hindawi Publishing Corporation, volume 2016, Article ID 5395894, DOI: <https://doi.org/10.1155/2016/5395894>.
5. Roopa M, Selva Kumar R (2016) Artificial neural network using back propagation algorithm in distributed MANETs, *International Conference on Information Communication and Embedded System*.
6. Torkestani AJ, Meybodi MR (2011) A mobility-based cluster formation algorithm for wireless Mobile Ad-Hoc Networks, *Cluster Computing*, 14(4):311–324.
7. Cheng H, Yang S, Cao J (2013), Dynamic genetic algorithms for the dynamic load balanced clustering problem in Mobile Ad Hoc Networks, *Expert Systems with Applications*, 40: 1381–1392.
8. Cheng ST, Li JP, Horng G.J (2013), An Adaptive Cluster-Based Routing Mechanism for Energy Conservation in Mobile Ad Hoc Networks, *Wireless Personal Communications: An International Journal*, 70(2): 561-579.
9. Govil K, Gupta SK, Agarwal A (2014), Cluster head selection technique for optimization of energy conservation in MANET', *International Conference on Parallel, Distributed and Grid Computing (PDGC), IEEE, Solan, India*, 39-42.
10. Basurra SS, Vos MD, Padget J, Ji Y, Lewis T, Armour S (2015), Energy efficient zone based routing protocol for MANETs, *Ad Hoc Networks*, 25: 16-37.
11. Periyasamy S, Khara S, Thangavelu S (2016), Balanced Cluster Head Selection Based on Modified k-Means in a Distributed Wireless Sensor Network, *International Journal Distrib. Sen. Network*, <https://doi.org/10.1155/2016/5040475>.
12. Prince T, Kannan ST (2017), Bat-inspired cluster head selection and on-demand cluster head gateway routing for prolonged network lifetime in MANET. *Int. J. Wire. Mob. Comput.*, 12(4) , 419-427.
13. Gupta A, Bisen D (2015), Review of different routing protocols in mobile ad-hoc networks. *International Journal of Computer Sciences and Engineering*, 3(3), 105–112.
14. Agarwal R, Gupta R, Motwani M (2015), Performance optimization through EPT-WBC in mobile ad hoc networks, *International Journal of Electronics*, 103(3), 355–371.
15. Bisen D, Sharma S (2018), An enhanced performance through Agent-based secure approach for mobile ad hoc networks, *International Journal of Electronics*, 105(1), 116-136.
16. Bisen D, Sharma S (2018), EE-SRA: An approach for MANET with unique feature of agent estimation, *National Academy Science Letters*, 42(2), 115–121.
17. Bisen D, Sharma S (2018), An energy efficient routing for performance enhancement of MANET through adaptive neuro-fuzzy inference system, *International Journal of Fuzzy System*, 20(8), 2693–2708.
18. Malar, A.C.J., Kowsigan, M., Krishnamoorthy, N. (2020), Multi constraints applied energy efficient routing technique based on ant colony optimization used for disaster

resilient location detection in mobile ad-hoc network., J Ambient Intell Human Comput, Springer.

19. Rao DS, Rao PN (2016), A cluster based scalable and energy efficient multi path routing protocol for MANET, *International Journal of Engineering Science*, 6(3), 3093-3099.
20. Kar S, Das S, Ghosh PK (2014) Applications of neuro fuzzy systems: A brief review and future outline, *Appl. Soft Comput.* 15, 243–259.
21. Chatterjee M, Das SK, Turgut D (2002), WCA: A Weighted Clustering Algorithm for Mobile Ad Hoc Networks. Vol. 5. Kluwer Academic.
22. Python, scikit-learn library (2021): <https://scikit-learn.org/stable/modules/clustering.html>.
23. The Network Simulator ns-2. (2021). Information Sciences Institute, USA. Viterbi School of Engineering, September. Retrieved from <http://www.isi.edu/nsnam/ns/>.