ENERGY-EFFICIENT QOS AWARE DATA TRANSMISSION IN MANET USING KIF-SDKMEANS AND ZS-AOA

Gajendra Kumar Ahirwar [1,*], Ratish Agarwal [2], Anjana Pandey[3]

[1,*]*Department Of Information Technology, RESEARCH SCHOLAR*
University Institute of Technology, RGPV, RGPV BHOPAL
[2]*Department of Information Technology, Associate Professor*
University Institute of Technology, RGPV, RGPV BHOPAL
[3]*Department of Information Technology, Associate Professor*
University Institute of Technology, RGPV,RGPV BHOPAL
[1,*]*gajendrakumarahirwar213@gmail.com*

KEYWORDS
Energy-efficient; FUZZY algorithm; KMeans algorithm; Ad Hoc On-Demand Distance Vector; Archimedes Optimization Algorithm; Fruit fly Optimization

ABSTRACT
Mobile Ad-hoc Network (MANET) is one of the recent fields in wireless communication that involves a large number of wireless nodes. Energy efficiency in data transmission is a major problem in MANET because the infrastructure of a network changes frequently, which results in data loss and communication overheads. The problem of energy efficiency in routing in mobile Ad-hoc networks has been approached in different methods. However, they suffer to achieve the required performance in the quality of service. So, to tackle such problems, the work proposed a novel energy-efficient QoS data transmission in MANET using KIF-SDK Means and ZS-AOA. Initially, the number of mobile nodes is initialized randomly and the mobile nodes are clustered together using efficient clustering techniques such as Kriging Interpolation based Fuzzy (KI-FUZZY) algorithm and Supremum Distance-based KMeans (SD-KMeans). Cluster heads are selected by using the KI-FUZZY algorithm and clustering was done by using SD-KMeans. Next, the test data transmission was done in order to invoke the routing between the nodes using Butterfly Effect Fruit fly Optimization Algorithm based Ad Hoc On-Demand Distance Vector (BEF-AODV). After test transmission, multiple routing paths are created. Finally, from the multiple created paths, the optimal routing paths are selected by using ZScore based Archimedes Optimization Algorithm (ZS-AOA) to perform energy-efficient base data transmission. The results show that the proposed model gives better performance for energy-efficient QoS data transmission in MANET than existing models.

1. INTRODUCTION

The continuous self-arranging, infrastructure-less network of mobile devices, which are linked wirelessly is called as Mobile Ad-hoc Network (MANET) [1, 2, 3]. It is a multi-hop short-term network without using centralized administration and infrastructure [4, 5]. The MANET has a great impact in different sectors and there are a number of applications using it. It can be deployed to perform data collection in different situations such as war fields, metrological, etc. It can be deployed quickly and the data collection can be performed in an efficient manner [6, 7]. The MANET mainly contains mobile nodes, and sensor nodes, along with a base station (BS) [8]. Nodes rely on the working strategy of receiving the packets and transmitting to the next-hop without making any delay unless and until the packet is an intended receiver. In MANET protocol, the stack comprises four layers like physical layer that performs bit by bit transmission of packets, the Network layer that is responsible for transmitting packets by identifying the route between source to destination, the transport layer that plays connection-oriented and connection-less transmission, and the Application layer, which is an acting application protocol for packet transmission. Based on the transmitting strategy, the packets get to travel from the source node to the destination node through intermediate nodes [9].

Compared to other network systems, MANET has significant advantages such as scalability, flexibility, restricted communication value, power parameters, wireless distributed radio medium as well as less infrastructure cost. MANET has been widely utilized in some applications such as device networks, wireless sensor-based systems, tactical networks, data networks, and so on. However, energy consumption, congestion, and fault tolerance are the different challenges in satisfying efficient communication in the MANET [10, 11]. Mainly, energy consumption is the key challenge in MANET [12, 13].

The energy consumption and transmission range of nodes in MANET are directly proportional to their transmitting power. Due to mobility, the distance between nodes and the network topology changes frequently; but, the transmission power of each node remains the same during communication. As an outcome, the energy consumption of nodes is not uniform across the network, which may leave the network disconnected [14] because the mobile nodes usually operated on limited batteries, often switching is leading to more battery consumption, and it could affect the nodes in a significant manner, as the data transmission between the nodes might get impacted and it could lead to more challenges [15]. Using the power-aware routing to examine the power consumption of the nodes when making routing decisions improves the performance of routing in MANETs [16].

In order to keep the network functional as long as possible, energy-efficient routing algorithms should be developed [17]. Many machine learning algorithms and deep learning algorithms have been developed to face this problem. However, the existing algorithm has a problem such as a network congestion, extra overhead, data cannot be transferred simultaneously and higher delay [18]. So, to overcome these problems, the work has proposed a novel technique for energy-efficient QoS aware data transmission in MANET using KIF-SDK Means and ZS-AOA.

The rest of the paper is organized as follows: Section 2 surveys the associated works regarding the proposed method, Section 3 explains the proposed methodology, and Section 4 illustrates the results and discussion of the proposed method based on performance metrics. Finally, section 5 concludes the paper with future work.

2. LITERATURE SURVEY

K. Karthick and R. Asokan [19] developed a Mobility Aware Routing Protocol for MANET using a hybrid optimization (MARP-HO) algorithm and Improved Ant Colony Optimization (IACO) algorithm, which maximized the QoS in data transmission. The model was used to achieve the exact quality requirement of data transmission. The result showed that the MARP-HO algorithm achieved better performance based on delay, packet delivery ratio, energy consumption, network lifetime, loss ratio, link stability, number of dead nodes, and throughput over existing state-of-art protocols. However, the IACO algorithm has the limitation of exploration and exploitation rate.

Neenavath veeraiah *et al.* [20] developed a trust-based secure energy-efficient navigation in MANETs. Initially, fuzzy clustering and CH selection utilized maximum value of direct, indirect, and recent confidence; the second was intruded node identification using a predefined threshold value of 0.5J; if the trust value of any node exceeded the predefined threshold value, it was considered a regular node; otherwise, it was considered an intruder node. The result showed that the method obtained minimal energy of 0.11m joules, a negligible delay of 0.005 ms, a maximum throughput of 0.74 bps, a maximum packet delivery ratio of 0.99 %, and a maximum detection rate of 90%. However, more security attacks were not focused on while analyzing the performance of the model.

Nguyen Minh Quy *et al.* [21] developed a QoS-Aware on-Demand Routing Protocol (QoS-ADRP) for urban-MANET applications. The QoS-ADRP protocol was used to support QoS and enhanced the whole system performance of urban-MANETs. The results showed that the QoS-ADRP improved the QoS flows, packet delivery ratio, latency, and throughput compared to existing protocols. However, the model was not focused on optimizing routing algorithms, which resulted in less network performance.

R. Manikandan *et al.* [22] improved energy efficiency in Mobile Adhoc Networks using mobility-based routing. The mobility-based routing algorithm was used for efficient data transmission by avoiding path disruptions. The results showed that the model obtained decreased power usage, better packet distribution, lower packet error rate, higher network lifetime, and lower end-to-end latency compared to conventional methods. However, the node position and motion angles towards the destination node were not focused on in this model.

Saleh A. Alghamdi [23] developed a cuckoo search-inspired meta-heuristic-based attempt for an optimized load-balancing energy-efficient routing protocol. The model employed the cuckoo search technique to designate an optimum routing path based on individual nodes and residual energy to balance the routing overhead among the individual nodes participating in routing. The model showed significant enhancements in packet delivery ratios, better battery life, and minimal packet delay time. However, there was a need for modification of the algorithm to ratify inherent time efficiency.

3. PROPOSED METHODOLOGY

In this paper, an energy-efficient QoS aware data transmission in MANET using KIF-SDKmeans and ZS-AOA is proposed. For energy-efficient data transmission, the proposed work undergoes the following steps; first, the sensor nodes are initialized. Then, the distance between the sensor nodes is identified by using Euclidean distance. Then, the clustering of nodes is performed. For an efficient cluster formation, cluster heads are selected first by using the KI-FUZZY algorithm, and then the nodes with the least distance to the cluster heads are grouped together by using the SD-Kmeans algorithm. Then, the network parameters are extracted from the clustered nodes. After that, to check the active state of the sensor nodes, the test transmission is performed. Thereafter, the multiple routing paths are created by using BEF-AODV. From the available paths, the most optimal energy-efficient path is selected by the means of ZS-AOA. Once the optimal path is identified, via the optimal path, the data gets transmitted to the destination. The architecture of the proposed framework is given in figure 1.

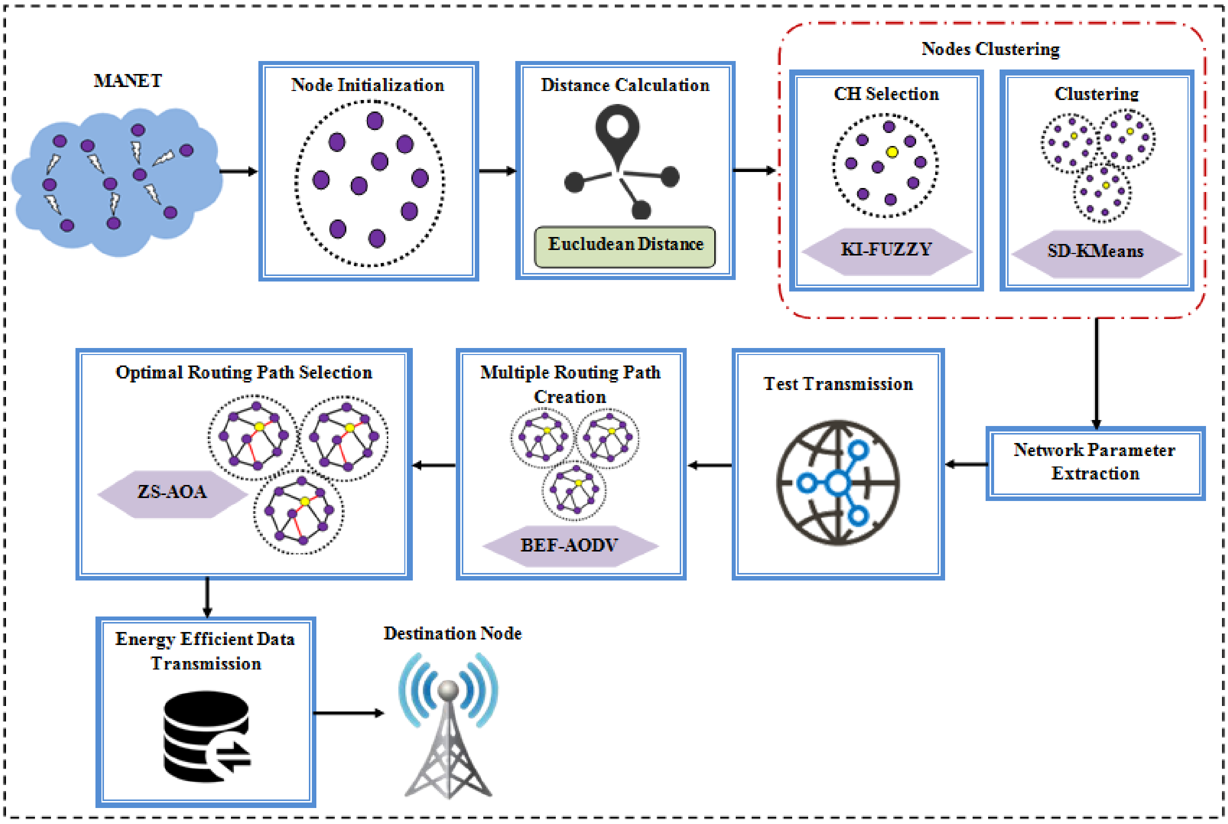

Figure 1: Architecture of the proposed framework

3.1. Initialization of nodes

In the initial step, the sensor nodes of the MANET environment are initialized randomly. The initialized sensor nodes are mathematically formulated as follows,

$$Sn_i = \{Sn_1, Sn_2, Sn_3, \ldots\ldots\ldots, Sn_n\} \tag{1}$$

3.2. Distance calculation

Once the sensor nodes are initialized, the distances between the sensor nodes are calculated. This distance calculation proficiently improves the clustering process in terms of accuracy and time. Hence, for an efficient distance calculation, the Euclidean distance is used. The expression for the Euclidean distance is given by,

$$E_{Dist}(Sn_1, Sn_2) = \sqrt{\sum_{i=1}^{n} (Sn_2 - Sn_1)^2} \tag{2}$$

Where, Sn_1 and Sn_2 defines the sensor nodes in Euclidean n space.

3.3. Clustering of nodes

After the distance estimation, the clustering of nodes is performed. Clustering is the process that groups the sensor nodes into several clusters so that the nodes in the same cluster have maximum similarity and the least distance as compared to the other clustered nodes. In the proposed work, the clustering of nodes is performed by undergoing two different steps. One is cluster head selection and another one is cluster formation.

3.3.1. Cluster head selection

Cluster head selection is an important step in clustering. Clustering involves the grouping of sensor nodes into a cluster and assigns the Cluster Head (CH) for each cluster. The CH is responsible to gather the data from the cluster sensor node and passes them to the base station. Hence, it is important to select an optimal (Ideal) CH. The ideal cluster head is the one that has maximum residual energy, the least distance of the nodes from the base station, and a maximum number of neighbor nodes. In the proposed methodology, the ideal cluster heads are selected by using the Kriging interpolation-based Fuzzy algorithm (KI-FUZZY). Here, to improve the rule generation process, the Kriging Interpolation membership function is used in the existing fuzzy system.

Basically, the Fuzzy logic system mainly performs three stages, which are input, processing, and output stage. First, the data are inputted, and inputted values are mapped for the operation. Then, the rules are generated by therule base operation, which was performed in the processing section. Finally, the defuzzifier performs a defuzzification system and sends the result to the output. Generally, the membership function is required for the fuzzy system; this membership function helps in mapping the elements.In this work, the Kriging Interpolation membership functionis used to avoid working on inaccurate inputs in the existing Fuzzy Logic System.The steps of the KI-FUZZYsystem are given as follows,

Initially, the distance vectors between the sensor nodes $Dt(Sn)$ are inputted to the fuzzy system. These inputsare converted to fuzzy sets using the fuzzification process as,

$$Dt(Sn)_e \xrightarrow{fuzzification} \Delta Dt(Sn)_e \tag{3}$$

Where, $\Delta Dt(Sn)_e$ denotes the converted fuzzy set.Then, the fuzzy set is applied to fuzzy interference where a set of rules are defined for mapping the input to the output phase. Then, the rules are generated, these rules are defined in the form of "If-Then" under linguistic variables such as low, high, and medium. For example, a rule R_r can be expressed as,

$$R_r = \begin{cases} IF\ Dt(Sn_1) < Dt(Sn_n) & THEN\ Sn_1 = OptimalNode \\ IF\ Dt(Sn_1 > Dt(Sn_n) & THEN\ Sn_1 \neq OptimalNode \end{cases} \tag{4}$$

Where, $Dt(Sn_1) < Dt(Sn_n)$ defines the initial sensor node that has minimum distance from the other sensor node, thus it is considered as an optimal node and the second condition is vice versa.

During this process, each input contains its membership functions to calculate the output ofeach rule. In order to improve the accuracy of this process, the Kriging Interpolation membership function (KI) is used. The expression for the KI is given by,

$$K_Z = \sum_{i=1}^{n} \beta_i K_i \tag{5}$$

Where, K_Z is a distance-vector between the nodes, which is estimated by kriging, β_i represents the random weight for K_i and K_Z is a variable.

Finally, the Defuzzification is performed according to the membership function. This Defuzzification process is used to convert the Fuzzy values into the output results. The architecture of the KI-FUZZY is given in figure 3.

Thus, the selected cluster heads Ch are mathematically formulated as follows,

$$Ch_i = \{Ch_1, Ch_2, Ch_3, ..., Ch_n\} \tag{6}$$

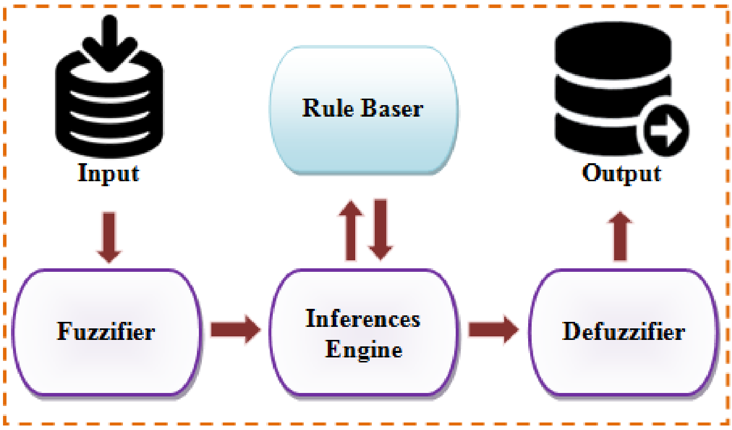

Figure 2: Illustration of Fuzzy algorithm

3.3.2. Cluster formation

Cluster formation is another step of the clustering phase. In this phase, the nodes with similar properties and the nodes with the least distance with CH are grouped together. The proposed method uses Supremum Distance-based KMeans (SD-KMeans) algorithm to form an efficient cluster. The conventional K-Means algorithm uses the Euclidean distance to calculate the distance between the data points (Sensor nodes) and the centroids (CH). But the Euclidean distance is not suitable for a large amount of data, due to which the clustering performance can be degraded. So, in this work, the Supremum Distance is used instead of the Euclidean distance. The Supremum distance metric is more advantageous for the handling of a huge amount of data points. Thus, the SD-KMeansform the ideal clusters with better clustering accuracy. The steps of the SD-KMeans are explained as follows,

Step 1: Initialize the number of data points as well as the centroids (CHs). Here, the data points represent the sensor nodes. The mathematical expression is given by,

$$Sn_i = \{Sn_1, Sn_2, Sn_3, ..., Sn_n\} \quad (7)$$

$$Ch_i = \{Ch_1, Ch_2, Ch_3, ..., Ch_n\} \quad (8)$$

Where, Sn_i be the number of sensor nodes and Ch_i is the set of centroids (CHs).

Step 2: Select the initial cluster centroid randomly. Thereafter, the data points, which have a minimum distance from the cluster centroid are assigned to the clusters. The distance is calculated by the means of theSupremum Distance metric. The expression for Supremum distance is given by,

$$S_{Dist}(Sn_i, Ch_i) = \lim_{M \to \infty} \left(\sum_{f=1}^{P} |Sn_i - Ch_i|^M \right)^{\frac{1}{M}} \quad (9)$$

$$\lim_{M \to \infty} \left(\sum_{f=1}^{P} |Sn_i - Ch_i|^M \right)^{\frac{1}{M}} = \underset{f}{\overset{P}{Max}} |Sn_i - Ch_i| \quad (10)$$

Where, M^{∞} is the uniform norm, f is the attribute that renders the maximum difference in values between Sn_i and Ch_i.

Step 3: Thus, the sensor nodes with minimum distance to the centroid (CHs) are assigned to the same cluster. The above steps are repeated until convergence is achieved. Thus, the SD-KMeans algorithm efficiently groups the sensor nodes, thus the clustered sensor nodes are defined by.

$$Cs_i = \{Cs_1, Cs_2, Cs_3, ..., Cs_n\} \quad (11)$$

Where, $\{Cs_1, Cs_2, Cs_3, ..., Cs_n\}$ denotes the number of formed clusters.

3.4. Network parameter extraction

From the formed clusters, the most required and informative network parameters like range, distances between the nodes, nodes location, and so on are extracted. The network parameters provide detailed information about the clustered nodes. Hence, these extracted features are highly contributed to the energy-efficient data transmission process. The extracted network parameters Np_i are mathematically formulated as follows,

$$Np_i = \{Np_1, Np_2, Np_3, ..., Np_n\} \quad (12)$$

3.5. Test transmission

Then, the paths are created between the source nodes to the destination nodes. Before that, the test transmission is performed. The test transmission is the process in which each node sends the request to the nearest neighbor nodes and gets the response from it. In doing so, the active state of each and every node from the network is identified; thereby the data transmission can be performed without any interruption.

3.6. Multiple routing path creations

After the successful test transmission, multiple routing paths are created. Paths are nothing but the routes by which the data packets reach their destination from the source. During path creation, it analyses the available paths for data transmission. Then, multiple paths arecreated between the sourceand destination. In the proposed work, themultiple routing paths are created by the means of Butterfly Effect Fruit fly Optimization Algorithm based Ad Hoc On-Demand Distance Vector (BEF-AODV).

3.5.1. Multiple routing path creations using BEF-AODV

AODV is an on-demand, single path, loop-free distance vector protocol. It combines the on-demand route discovery mechanism in Dynamic Source Routing (DSR) with the concept of destination sequence numbers from Destination Sequenced Distance Vector (DSDV). Generally, the AODV uses a hop-by-hop routing approach. But, in AODV, there is a single path from the source to the destination in the routing table. When the path is disconnected, the source node again initiates the routing request, this leads to network overhead and waste of bandwidth. So to avoid that, possible optimal routes are discovered first. For an optimal route selection, an optimization algorithm called Butterfly EffectFruit fly optimization (BEF) is used. Thus, the steps of BEF-AODV are briefly explained as follows.

In AODV, before the route discovery process begins each node maintains the routing table. The routing table contains essential information like source, destination, Hop count, Destination sequence number, and broadcast ID.

- ***Source:*** The node initiates the routing process by flooding RREQ.
- ***Destination:*** The node where the RREQ is destined to.
- ***Hop count:*** This represents the number of transmissions required to reach the destination.
- ***Destination sequence number:*** This parameter is used to reduce the transient routing loops.
- ***Broadcast ID:*** This ensures that each message is not transmitted more than once.

(a) Route discovery

In AODV, nodes discover routes in request-response cycles. Whenever a source node needs a route to a destination, it initiates a route discovery by flooding a route request (RREQ) message to all its neighbors. When a node receives an RREQ message, it checks whether there is a valid route available for the destination. If the route is available then it unicasts an RREP back to the source via the reverse path; otherwise, it rebroadcasts the RREQ packet.

This process repeats until the RREQ reaches a node that has a valid route to the destination. Thus, at the end of this request-response cycle, a bidirectional route is established between the requesting node and the destination. During the route discovery process, the available optimal paths are selected by using BEF. The steps of BEF are elucidated below.

The Fruit fly optimization (FOA) is a new swarm intelligent method for optimization, which is inspired by the food foraging behaviours of the fruit flies. In order to find the best fruit fly, the butterfly effect (BE) is incorporated with FOA; this improvisation efficiently helps to identify the best fruit fly among the population. Here, the fruit fly defines the number of sensor nodes in the network. The food foraging behaviours of the fruit flies are summarized in the following steps.

- Initially, the flies smell the food through olfactory organs and fly towards that location.
- Using the sensitive visions, the flies get closer to the food location.
- Lastly, other flocks of fruit flies fly towards that direction where the food is abundant.

Step 1: At first, the parameters of the FOA are initialized. The parameters include population size, Maximum iteration, and initial fruit fly swarm location (x^0, y^0).

Step 2: The fruit flies are randomly distributed in the search space. The individual fruit fly from the population is initialized as follows.

$$x_i = x^0 + Rd \tag{13}$$

$$y_i = y^0 + Rd \tag{14}$$

Where, Rd represents a random vector and that was sampled from a uniform distribution.

Step 3: Calculate the distance between the individual fruit fly (dis_i) and the smell concentration judgment value of the fireflies (δ_i) using the below expressions.

$$dis_i = \sqrt{x_i^2 + y_i^2} \tag{15}$$

$$\delta_i = \frac{1}{dis_i} \tag{16}$$

Step 4: For each individual fly, the smell concentration judgment value is computed by subjecting (δ_i) to the fitness function.

$$Sm_i = Fitness(\delta_i) \tag{17}$$

Step 6: Find the individual fruit fly with the best fitness function among the fruit fly swarm, which means the fruit fly with maximum smell concentration (BSm_i) by using the following expression.

$$[BestSm \; BestIx] = Max(Sm_i) \tag{18}$$

Where, (δ_i) defines the fruit fly with the best smell.

Step 7: The fruit flies keep the best smell concentration value and use their vision to fly towards that location. This process is mathematically written as.

$$Sm_{Best} = BestSm \tag{19}$$

$$x_{Best} = \left(x_j^i + \left(\Re^2 \times b^* - x_j^i\right) \times Sm_i\right) \tag{20}$$

$$y_{Best} = \left(y_j^i + \left(\Re^2 \times b^* - y_j^i\right) \times Sm_i\right) \tag{21}$$

Where, x_j^i defines the solution vector x_j for j_{th} firefly in iteration i, b^* defines the current best solution found among all the solutions in current iteration. Smell concentration of i_{th} firefly is signified as Sm_i.

The iteration continues until the maximum iteration is achieved.

(b) Route maintenance

Route maintenance is done by means of route error (RERR) packets. When an intermediate node detects a link failure, it generates a RERR packet. The RERR propagates towards all source nodes having a route via the failed link and erases all broken routes on the way. When a source node receives the RERR, it initiates a new route discovery if it still needs the route.

Thus, the numbers of possible or available optimal routes for the destination are defined by,

$$P_i = \{P_1, P_2, P_3, \ldots\ldots\ldots, P_n\} \tag{22}$$

3.7. Optimal routing path selection

Once the available paths are discovered, the most optimal or finest paths are selected. The cost-efficient routes with the least distance to the destination are considered as the optimal paths. Generally, in WSN, the data packets are transmitted from the source to the destination. For the data transmission, the nodes require the assistance of neighboring nodes to relay the data packets to the destination. Thus, the energy levels of the nodes that are involved in the routing process are drastically reduced. Due to this, the draining of energy happens and that leads to the network lifetime degradation. Hence, to increase network lifetime and also for reducing energy consumption, the optimal routing path selection is required. In this proposed work, the optimal paths are selected by the means of the ZScore-based Archimedes Optimization Algorithm (ZS-AOA).

3.7.1. Optimal routing path selection using ZS-AOA

Archimedes Optimization Algorithm (AOA) is the population-based algorithm, which is inspired by Archimedes' principle. Here, the population defines the immersed objects. It describes the behavior of the exerted force when a particle is partially or completely immersed in a fluid. Each particle has its own volume, density, and acceleration, which are updated in an iterative process. Here, to improve the acceleration of the object, the Z-Score normalization is used. Thus, the updating procedure of the ZS-AOA is detailed in the following steps:

Step 1: In the beginning step, the positions of all objects are initialized. Here, the objects signifies the number of paths. Thus, the objects are initialized by using.

$$P_i = Lb_i + Rand \times (Ub_i - Lb_i); \quad i = 1,2,3,........,n \tag{23}$$

Where, P_i defines the i_{th} object in a population of n objects, Lb_i and Ub_i are the lower and upper bounds of the search-space,respectively.

Then, the density Dn_i and the volume Vl_i for each i_{th} object are determined by using,

$$Dn(i) = Rand \tag{24}$$

$$Vl(i) = Rand \tag{25}$$

Where, $Rand$ is a d dimensional vector, that randomly generates a number between [0, 1].

At last, the acceleration Acl of the i_{th} object is determined by using,

$$Acl(i) = Lb_i + Rand \times \left(Ub_i - Lb_i\right) \tag{26}$$

In this phase, the fitness value of each object is computed and the solution with the best value is selected. Thus, the best values are denoted as $Y(b)$, $Dn(b)$, $Vl(b)$, and $Acl(b)$.

Step 2: For each iteration, the density and volume of each solution i is updated. Thus, the updation process for the iteration $k+1$ is expressed as.

$$Dn_i^{k+1} = Dn_i^k + Rand \times \left(Dn_{Best} - Dn_i^k\right) \tag{27}$$

$$Vl_i^{k+1} = Vl_i^k + Rand \times \left(Vl_{Best} - Vl_i^k \right) \tag{28}$$

Where, $Vl(b)$ and $Dn(b)$ are the volume and density of the global solution, and $Rand$ denotes a random number in the range from 0 to 1.

Step 3: In this phase, the collision occurs between the objects, after that, the objects try to reach the equilibrium state. This function is performed by the transfer operator Tf, which transforms the search from exploration to exploitation. Thus, Tf is expressed by,

$$Tf = Exp\left(\frac{k - k_{Max}}{k_{Max}} \right) \tag{29}$$

Where, the Tf increase gradually with time until it reaches 1, k and k_{Max} denotes the iteration number and maximum iterations, respectively.

The, density decreasing factor Df helps the AOA to reach global search. This factordecreases with time using the following expression.

$$Df^{k+1} = Exp\left(\frac{k_{Max} - k}{k_{Max}} \right) - \left(\frac{k}{k_{Max}} \right) \tag{30}$$

Here, the value Df^{k+1} of gradually decreases to help the solutions to converge in the promising search area.

Step 4 - Exploration phase: If the transfer operator Tf is less than or equal to 0, 5, then the collision occurs between the objects. In this case, a random material Rm is selected and updates the acceleration of the object for iteration $k+1$ by using.

$$Acl_i^{k+1} = \frac{Dn_{Rm} + Vl_{Rm} \times Acl_{Rm}}{Dn_i^{k+1} \times Vl_i^{k+1}} \tag{31}$$

Where, Dn_{Rm}, Vl_{Rm}, and Acl_{Rm} signifies the randommaterial's density, volume, and acceleration respectively.

Step 5 - Exploitation phase: If the transfer operator Tf is greater than 0.5, then there is no collision. Thus, the object's acceleration for iteration $k+1$ is updated using.

$$Acl_i^{k+1} = \frac{Dn(b) + Vl(b) \times Acl(b)}{Dn_i^{k+1} \times Vl_i^{k+1}} \tag{32}$$

Here, $Acl(b)$ represents the acceleration of the best object.

Step 6 - Normalize acceleration: The acceleration of the i_{th} object is normalized by using below expression.

$$Z_{Score} = \frac{Acl_i^{k+1} - \mu(Acl_i^{k+1})}{SD(Acl_i^{k+1})} \tag{33}$$

Where, Acl_i^{k+1} defines the acceleration of the i_{th} object, μ represents the mean, and $k+1$ signifies the standard deviation.

Step 6 - Update position: If the transfer operator $Tf \le 0.5$ then in the exploration phase, the i_{th} object updates the position for the next iteration $k+1$ using.

$$Y_i^{k+1} = Y_i^k + C_1 \times Rand \times Acl_{iNorm}^{k+1} \times Df \times \left(Y_{Rand} - Y_i^k\right) \tag{34}$$

Where, C_1 represents a constant value of 2.

If the transfer operator $Tf > 0.5$ then in the exploitation phase, the solutions update their positions with respect to the following expression.

$$Y_i^{k+1} = Y(b)^k + G \times C_2 \times Rand \times Acl_{iNorm}^{k+1} \times Df \times \left(I \times Y(b) - Y_i^k\right) \tag{35}$$

Where, C_2 defines a constant value of 6, I signifies the increasing variable, which was calculated by using.

$$I = C_3 \times Tf \tag{36}$$

Here, I surges with time in range $[C_3 \times 0.3, 1]$ and considers the certain percentage from the best position. Initially, it starts with a low percentage, if this result has a large difference between the best position and the current position, then the step size of the random walk increases. The increase in this percentage decreases the difference between the best position and the current position. This achieves the balance between exploration and exploitation. Here, G defines a flag for changing the motion's direction in accordance with the following expression.

$$G = \begin{cases} +1 & if \ p \le 0.5 \\ -1 & if \ p > 0.5 \end{cases} \tag{37}$$

Here, $p = 2 \times Rand - C_4$

Finally, the global best solutions are obtained. Here, the best solutions represent the optimal path. Thus, the obtained optimal paths are mathematically written as follows.

$$O(P)_i = \{O(P)_1, O(P)_2, O(P)_3, ..., O(P)_n\} \tag{38}$$

Algorithm 1 Pseudo-code of the ZS-AOA algorithm

Input: Number of path

Output: Selection of optimal path

Begin
- **Initialize** the position of objects P_i, $Dn(i)$ and $Vl(i)$
- **Evaluate** the initial position and select the particle with best fitness value
- Set iteration counter $k = 1$
- **While** $k = k_{Max}$ **do**
  - **For** each object i **do**
    - **Update** density and volume of each object using Dn_i^{k+1} and Vl_i^{k+1}
    - **Update** transfer and density decreasing factors TF and DF using Tf and Df^{k+1}
    - **If** $TF \le 0.5$ **then** **//Exploration phase**
      - **Update** acceleration using Acl_i^{k+1}
    - **Else** **//Exploitation phase**
      - **Update** acceleration using Acl_i^{k+1}
      - **Update** position using Y_i^{k+1}
      - **Update** direction flag, G
    - **End if**
  - **End for**
  - **Evaluate** each object and select the one with the best fitness value
  - Set $k = k + 1$
- **End while**
  - **Return** object with best fitness value

End

Figure 3: Pseudo code for the ZS-AOA algorithm

Once the optimal paths are discovered, the data packets are transmitted to the destination with less energy consumption, this helps to increase the life of the networkand also improves the quality of services.

4. RESULTS AND ANALYSIS

This section analyses the performance of the proposed model by comparing its results with other existing models. The proposed model is implemented in the working platform of python and data are collected from the dataset MANET – CRAWDAD. In order to state the efficiency of the proposed work, the performance analysis, as well as comparative analysis, is carried out.

4.1Dataset Description

CRAWDAD is the Community Resource for Archiving Wireless Data At Dartmouth, a wireless network data resource for the research community. This archive has the capacity to store wireless trace data from many contributing locations, and staff to develop better tools for collecting, anonymizing, and analyzing the data.

4.2 Performance Analysis of Proposed KI-FUZZY Algorithm

The performance analysis of the proposed KI-FUZZY is validated with respect to cluster head selection time. Then, the obtained analysis results are compared with existing FUZZY algorithms.

Table 1: Performance analysis of the proposed KI-FUZZY based on cluster head selection time

No. of Nodes	FUZZY (ms)	Proposed KI-FUZZY (ms)
100	16266	11610
200	32117	22745
300	48560	34124
400	64551	45919
500	80431	58051

Table 1 shows the performance analysis of proposed KI-FUZZY and existing FUZZY algorithms based on cluster head selection time. The time taken to select the optimal node from the cluster is called cluster head selection. Lower the cluster head selection shows better performance of the model. The proposed technique has a cluster head selection time of 11610ms, 22745ms, 34124ms, 45919ms, and 58051ms for the nodes 100, 200, 300,400, and 500, whereas the existing FUZZY has a cluster head selection time of 16266ms, 32117ms, 48560ms, 64551ms, and 80431ms. On comparing these values it is concluded that the proposed technique attains lower cluster head selection time than the existing technique. So, it is concluded that the proposed technique shows better performance for energy-efficient QoS aware data transmission in MANET.

4.3 Performance Analysis of Proposed SD-KMeans Algorithm

The performance analysis of the proposed SD-KMeans is validated with respect to clustering time. Then, the obtained results are compared with other existing algorithms such as K-Means, Partition around Medoids algorithm (PAM), Clustering Large Application (CLARA), and Fuzzy C Means (FCM) in order to prove their effectiveness.

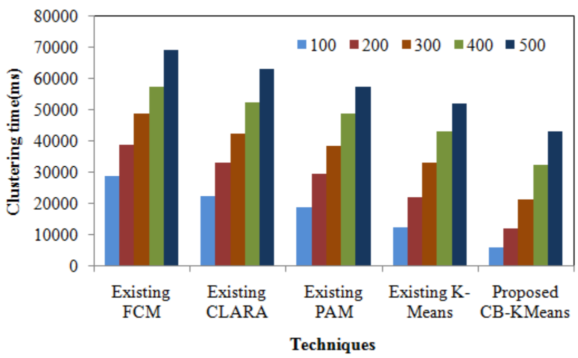

Figure 4: Comparative analysis of the proposed SD-KMeans in terms of clustering time

Figure 4 shows the clustering time of the proposed SD-KMeans and the existing techniques such as K-Means, PAM, CLARA, and FCM. For node 100 the clustering time of the proposed model attains 6054ms, whereas, the existing models attain the clustering time of 12484ms, 18764ms, 22324ms, and 28755ms. Similarly, for other nodes, the proposed model attains lower clustering time when compared to other existing models.

4.4 Performance Analysis of Proposed ZS-AOA Algorithm

The performance analysis of the proposed ZS-AOA algorithm is validated with respect to packet loss ratio, throughput, packet delivery ratio, network lifetime, energy consumption, and delay. Then, the obtained results are compared with various algorithms such as Angle of Arrival (AOA), Right Hand Side (RHS), Migrating Birds Optimization (MBO), and Crow Search Algorithm (CSA) in order to prove their effectiveness.

Table 2(a): Performance analysis of the proposed ZS-AOA based on packet loss ratio

No. of Nodes	CSA (%)	MBO (%)	RHS (%)	AOA (%)	Proposed ZS-AOA (%)
100	25.64	22.32	17.84	14.32	8.32
200	27.32	21.54	18.65	13.56	7.35
300	28.62	22.65	16.84	13.47	8.47
400	26.84	23.54	16.84	14.32	6.89
500	27.65	21.45	17.51	13.54	7.52

Table 2(b): Performance analysis of the proposed ZS-AOA based on packet delivery ratio

No. of Nodes	CSA (%)	MBO (%)	RHS (%)	AOA (%)	Proposed ZS-AOA (%)
100	81.24	84.79	88.33	91.33	97.53
200	82.15	85.63	89.25	92.46	95.45
300	80.33	83.55	87.46	91.76	96.33
400	81.46	84.76	89.65	92.52	95.87
500	80.33	83.66	88.65	92.76	96.23

Table 2 revealed the packet loss ratio and packet delivery ratio of the proposed ZS-AOA algorithm and the existing algorithms such as AOA, RHS, MBO, and CSA. Table 2(a) shows the packet loss ratio of the proposed technique. The packet loss ratio is the number of packet losses during transmission. Lower the packet loss ratio results in better performance of the model. In the proposed model the packet loss ratio for node 100 is 8.32%, whereas the existing techniques have the packet loss ratio for node 100 are 14.32%, 17.84%, 22.32%, and 25.64%. Similarly, the packet loss ratio of the proposed technique for nodes 200, 300, 400, and 500 are 7.35%, 8.47%, 6.89%, and 7.52%, which is lower when compared to the other existing techniques. The packet delivery ratio of the proposed algorithm and the existing algorithms are shown in table 2(b). The packet delivery ratio of the proposed technique for nodes 100, 200, 300,400, and 500 are 97.53%, 95.45%, 96.33%, 95.87%, and 96.23%, which are higher when compared to the existing techniques.

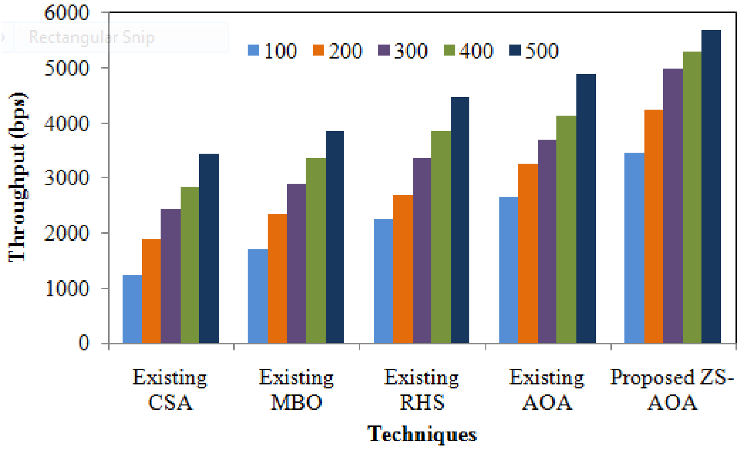

Figure 5: Comparative analysis of the proposed ZS-AOA in terms of throughput

Figure 5 shows the comparative analysis of the proposed ZS-AOA based on throughput. Throughput is a measure of total units of information a system can process in a given amount of time. Higher the throughput value results in better performance of the model. The throughput of the proposed technique for node 100 is 3457bps, whereas the throughputs of the existing techniques for node 100 are 2657bps, 2257bps, 1697bps, and 1234bps. Likewise, the proposed technique has the throughput for nodes 200, 300,400, and 500 are 4251bps, 4987bps, 5314bps, and 5685bps. On comparing these values the throughput of the proposed technique attains a higher value than the existing techniques.

Table 3(a): Performance analysis of the proposed ZS-AOA based on delay

No. of Nodes	CSA (ms)	MBO (ms)	RHS (ms)	AOA (ms)	Proposed ZS-AOA (ms)
100	2155	1694	1358	778	348
200	28675	2227	1857	1257	874
300	3377	2755	2256	1694	1247
400	3857	3264	2857	2258	1745
500	4225	3694	3261	2694	2143

Table 3(b): Performance analysis of the proposed ZS-AOA based on energy consumption

No. of Nodes	CSA (J)	MBO (J)	RHS (J)	AOA (J)	Proposed ZS-AOA (J)
100	9854	7497	5422	4137	3245
200	11557	9554	7433	6221	4578
300	13664	11484	9543	8134	6532
400	15758	13462	11668	10124	8475
500	17134	15257	13457	12421	10325

The performance analysis of the proposed ZS-AOA based on delay and energy consumption is shown in table 3. Table 3(a) shows the delay of the proposed algorithm and the existing techniques. When the delay of the technique decreases, it shows better performance of the model. The proposed technique has the delay for nodes 100, 200,300, 400, and 500 are 348ms, 874ms, 1247ms, 1745ms, and 2143ms, which are lower when compared to the existing techniques. Similarly, the energy consumption of the proposed techniques and the existing techniques are shown in table 3(b). The proposed technique attains the energy consumption for nodes 100, 200,300,400, and 500 are 3245J, 4578J, 6532J, 8475J, and 10325J, which are lower when compared to the existing techniques

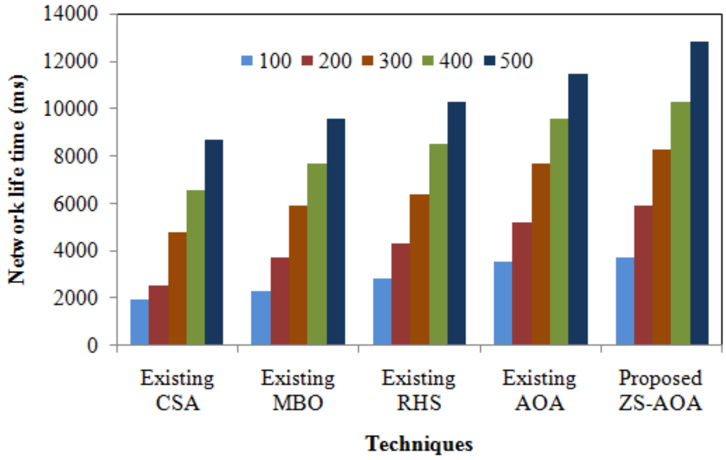

Figure 6: Comparative analysis of the proposed ZS-AOA in terms of Network lifetime

The comparative analysis of the network lifetime of the proposed ZS-AOA and the existing techniques are shown in figure 6. The proposed technique attains the network lifetime of 3658ms for node 100, 5847ms for node200, 8245ms for node 300, 10245ms for node 400, and 12784ms for node 500, which are higher when compared to the existing techniques. So, from the results of all metrics, it is concluded that the proposed ZS-AOA shows better performance for energy-efficient QoS-aware data transmission in MANET.

5. CONCLUSION

The work has proposed an energy-efficient QoS aware data transmission in MANET using the KIF-SDKMeans algorithm for the clustering process and the ZS-AOA algorithm for optimal routing path selection. The proposed method undergoes the following process such as random node initialization, distance calculation, node clustering, network parameter extraction, test transmission, multiple routing path creations, optimal routing path selection, and energy-efficient data transmission. After that, the experimental analysis is performed in which the performance analysis and the comparative analysis of the proposed techniques were done in terms of various metrics. The final outcomes reveal that the proposed clustering algorithm forms an efficient cluster head selection time and clustering time with a limited amount of time such as 11610ms and 6054ms. Furthermore, the proposed method achieves an 8.32% for packet loss ratio, 97.53% for packet delivery ratio, 3457bps for throughput, 348ms for the delay, and 3245J for energy consumption for node 100. Similarly, the proposed model achieves better performance for all other nodes. So, from the results, it is concluded that the proposed method shows better performance for energy-efficient QoS aware data transmission in MANET. In the future, the work will be extended with some advanced techniques for the detection of security during data transmission in MANET.

REFERENCES

1. Tamil Selvi P and Suresh Ghana Dhas C, "A novel algorithm for enhancement of energy efficient zone based routing protocol for MANET", Mobile Networks and Applications, vol. 24, no. 2, pp. 307-317, 2018.
2. Sheng Hao, Huyin Zhang and Mengkai Song, "A stable and energy-efficient routing algorithm based on learning automata theory for MANET", Journal of Communications and Information Networks, vol. 3, no. 2, pp. 43-57, 2018.
3. Deepa M, Krishna Priya P and Sivakumar S, "QoS enabled optimized adaptive multipath AODV protocol", SN Computer Science, vol. 1, no. 2, pp. 1-9, 2020.
4. Murugan Singaravelan and Bhuvaneswari Mariappan, "Reinforcement energy efficient ant colony optimization of mobile ad hoc multipath routing performance enhancement", International Arab Journal of Information Technology, vol. 19, no. 2, pp. 195-202, 2022.
5. Arockiasamy Karmel, Varadarajan Vijayakumar and Radhakrishnan Kapilan, "Ant-based efficient energy and balanced load routing approach for optimal path convergence in MANET", Wireless Networks, vol. 27, no. 8, pp. 5553-5565, 2019.
6. Kowshika Arumugham and Vivekanandan Chenniappan, "Least mobility high power (LMHP) dynamic routing for QoS development in MANET", Wireless Personal Communications, vol. 105, no. 1, pp. 355-368, 2019.
7. Meena Rao and Neeta Singh, "Energy efficient QoS aware hierarchical KF-MAC routing protocol in manet", Wireless Personal Communications, vol. 101, no. 1-2, pp. 635-648, 2018.
8. Prasad A. Y and Balakrishna R, "Implementation of optimal solution for network lifetime and energy consumption metrics using improved energy efficient LEACH protocol in MANET", Telkomnika, vol. 17, no. 4, pp. 1758-1766, 2019.
9. Anand M and Sasikala T, "Efficient energy optimization in mobile ad hoc network (MANET) using better-quality AODV protocol", Cluster Computing, vol. 22, no. 3-5, pp. 12681-12687, 2018.
10. Nanditha Boddu, Veeramallu Boba and Ramesh Vatambeti, "A novel georouting potency based optimum spider monkey approach for avoiding congestion in energy efficient mobile Ad-hoc network", Wireless Personal Communications, 2021, https://doi.org/10.1007/s11277-021-08571-4.
11. Dhananjay Bisen and Sanjeev Sharma, "An energy-efficient routing approach for performance enhancement of MANET through adaptive neuro-fuzzy inference system", International Journal of Fuzzy Systems, vol. 20, no. 1, pp. 1-16, 2018.
12. Rachna Jain and Indu Kashyap, "An QoS aware link defined OLSR (LD-OLSR) routing protocol for MANETS", Wireless Personal Communications, vol. 108, no. 6, pp. 1-14, 2019.
13. Vu Khanh Quy, "An adaptive on-demand routing protocol with QoS support for urban-MANETs", Preprint, 2022, https://doi.org/10.21203/rs.3.rs-873575/v1.
14. Rashmi Chaudhry and Shashikala Tapaswi, "Optimized power control and efficient energy conservation for topology management of MANET with an adaptive gabriel graph", Computers and Electrical Engineering, vol. 72, pp. 1021-1036, 2018.

15. Papanna N, Rama Mohan Reddy A and Seetha M, "EELAM energy efficient lifetime aware multicast route selection for mobile Ad Hoc networks", Applied Computing and Informatics, vol. 15, no. 2, pp. 120-128, 2017.
16. Ashraf Al Sharah, Mohammad Alhaj and Firas Al Naimat, "Trade-off between energy consumption and transmission rate in mobile Ad-Hoc network", (IJACSA) International Journal of Advanced Computer Science and Applications, vol. 12, no. 3, pp. 245-252, 2021.
17. Mohamed Er-rouidi, Houda Moudni, Hicham Mouncif and Abdelkrim Merbouha, "A balanced energy consumption in mobile Ad hoc network", Procedia Computer Science, vol. 151, pp. 1182-1187, 2019.
18. Mohd Naseem, Gulfam Ahamad, Sparsh Sharma and Ehtesham Abbasi, "EE-LB-AOMDV an efficient energy constraints-based load-balanced multipath routing protocol for MANETs", International Journal of Communication Systems, vol. 34, no. 16, pp. 1-13, 2021.
19. Karthick K and Asokan R, "Mobility aware quality enhanced cluster based routing protocol for mobile Ad-Hoc networks using hybrid optimization algorithm", Wireless Personal Communications, vol. 119, no. 3-4, pp. 3063-3087, 2021.
20. Neenavath Veeraiah, Osamah Ibrahim Khalaf, Prasad C. V. P. R, Youseef Alotaibi, Abdulmajeed Alsufyani, Saleh Ahmed Alghamdi and Nawal Alsufyani, "Trust aware secure energy efficient hybrid protocol for MANET", IEEE Access, vol. 9, pp. 120996-121005, 2021.
21. Nguyen Minh Quy, Nguyen Tien Ban and Vu Khanh Quy, "An adaptive on-demand routing protocol with QoS support for urban-MANETs", IAENG International Journal of Computer Science, vol. 49, no. 1, pp. 1-8, 2022.
22. Manikandan R, Muruganantham Ponnusamy, Raja K and Ramkumar M, "Improved energy efficiency in mobile adhoc networks using mobility based routing", ICTACT Journal on Communication Technology, vol. 12, no. 1, pp. 2380-2385, 2021.
23. Saleh A Alghamdi, "Cuckoo energy-efficient load-balancing on-demand multipath routing protocol", Arabian Journal for Science and Engineering, vol. 47, no. 2, pp. 1321-1335, 2021.