Received: May 16, 2024. Revised: August 29, 2024.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Enhancing Data Routing in Wireless Sensor Networks through Predictive Mobile Sink Trajectory Modeling with ALDS-Net

Kaumudi Keerthana[1]* **A. Mahesh Babu[1]**

[1]*Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Hyderabad-500075, Telangana, India*
* Corresponding author's Email: keerthana24k@rediffmail.com

Abstract: In the realm of Wireless Sensor Networks (WSNs), enhancing network longevity and accurately predicting mobile sink trajectories are critical challenges. This research introduces ALDS-Net (Adaptive Moment Estimation Long Short-Term Memory DeepSink Network), a novel model specifically designed to address these challenges by leveraging advanced deep learning techniques. The novelty of ALDS-Net lies in its unique integration of the ADAM optimizer with LSTM layers, enabling it to effectively capture and model complex mobility patterns of mobile sinks. The ADAM optimizer is utilized to manage noisy gradients, enhancing the stability and convergence of the model, while LSTM layers are employed to handle the long-term dependencies in sink mobility data. A key innovation of ALDS-Net is its three-stage simulation methodology, which includes initializing the simulation environment, training the predictive model using a simulated dataset, and deploying the trained model for real-time sink trajectory prediction in WSNs. This comprehensive approach ensures accurate modeling of sensor node interactions and sink mobility, resulting in improved data routing efficiency and extended network lifetime. Our experimental results demonstrate that ALDS-Net outperforms existing methods such as Adjacency Based Cell Score, Differential Moth Flame Optimization (DMFO), and Instantaneous Clustering Algorithm (ICP) in terms of energy efficiency and network stability. Specifically, ALDS-Net achieves a significant extension in network operational lifetime, reaching 498.5 seconds with 400 nodes over 750 rounds. This study highlights the potential of ALDS-Net to provide robust, efficient, and scalable solutions for data routing in dynamic WSN environments.

Keywords: Wireless, Prediction, Nodes, Rounds, Lifetime, Efficiency, Mobile sink.

1. Introduction

Wireless sensor networks (WSNs) have become prevalent in many areas such as environmental monitoring, health care and industrial automation [1-2]. These are made up of small energy-constrained sensor nodes that use mobile sinks strategically located to collect and transfer data [3]. Therefore, due to this need for optimization of data collection as well as network performance it became necessary to predict the movement of these mobile sinks [4, 5] and their unpredictable moves. Still, few studies have focused on forecasting the trajectory of a moving sink within a WSN [6].

Mobile Sinks represent either vehicles or mobile nodes with higher processing power and storage capacity which are used to collect information from energy constrained or stationary sensors [7]. Accurate prediction of trajectories can greatly enhance efficiency in gathering intelligence leading to longer lifetimes for networks coupled with better quality collected data [8]. They play an important role in today's world where real time monitoring is needed on critical parameters. This however depends on how best they can be managed within WSNs thus more research needs to be undertaken in this area [9].

Two issues drive the need for predictive models for mobile washbasin management; energy saving and route optimization. Number one, the reason battery life is a big deal in most sensors is because they run on limited power sources [10, 11]. Number two, that sink passes through different locations as it moves around collecting data during its roving period; hence some if not all parts of this information might be influenced by where the sink goes through. For this reason, forecasting trajectories of sinks is necessary for intelligent route planning that guarantees full coverage while saving energy.

Several new features have been brought out by this particular research; it uses a three stage simulation method for WSNs involving initialization, training models using gathered information and then collecting these initially missed pieces themselves. The proposed ALDS-Net model uses deep learning techniques to accurately simulate interactions between sensor nodes and mobility patterns among sinks. Sink position prediction accuracy is enhanced through utilization of simulated based dataset during predictive modeling. Scalability evaluation demonstrates how suitable this framework can be for larger networks besides showing that it significantly prolongs WSN lifetime over existing methods hence indicating better energy efficiency and network survivability brought about by ALDS-Net model.

The paper is divided into sections that build up towards giving an understanding on what has been done before and why it was necessary to undertake this research. Section 2 - Literature Review provides broad coverage of previous studies in order to set context for the current investigation. Section 3 - Methodology elaborates more on how things were done including detailed process flow and algorithm descriptions while section 4 - Experimental Investigations presents findings from different tests carried out which prove the superiority of proposed model. Finally, this paper concludes with a summary of the works carried out.

2. Literature review

A prominent focus of study in wireless sensor networks (WSNs) involves the management of mobile sinks and routing issues. In 2018, Sarwar et al. proposed distributed algorithms to calculate the minimum number of sinks needed for deployment in Wireless Sensor Networks (WSNs). It was discovered that the quantity of sinks is inversely proportional to the transmission range. Nevertheless, this approach relies on a fixed network structure and particular environmental circumstances, which may not consistently reflect reality due to the movement of nodes and obstacles in the environment. In practical situations, the movement of nodes can cause unexpected alterations in the structure of a network, which can affect the effectiveness of sink placement strategies.

In 2018, Naween et al. [13] proposed a heuristic technique to optimize the placement of data collectors over time in a wireless sensing network. This approach proposed that the mobile collector could read different points at different times to maximize data collection efficiency. Nevertheless, this approach fails to consider the dynamic fluctuations in node positions or the varying densities of the network, resulting in suboptimal data collection and heightened energy consumption. The limitations are addressed by our proposed model through the dynamic prediction of sink trajectories, resulting in improved efficiency in data collection.

According to Maurya S. et al. [14], the majority of existing routing protocols for homogeneous sensor networks fail to address multiple concerns at the same time, resulting in gaps in coverage and higher energy consumption. The Delay-Aware Energy-Efficient Reliable Routing (DAEERR) algorithm was suggested. However, it is not well-suited for large-scale networks due to its high computational cost, despite its effectiveness. The practical applications of such algorithms can be limited by their computational complexity and energy requirements, especially in environments with limited resources. The integration of the ADAM optimizer and LSTM layers in our approach offers a computationally efficient solution without compromising prediction accuracy.

Thomson et al. [15] created an energy optimization algorithm named mobility aware duty cycling for WSNs, which adjusts the duty cycles of nodes based on their movement patterns. Although this algorithm has the potential to be effective, it has demonstrated considerable latency in transmitting data, which can be crucial in time-sensitive applications. Our ALDS-Net model addresses these delays by utilizing LSTM layers to predict sink movements, guaranteeing prompt data collection and transmission.

In 2020, Saunhita Sapre and S. Mini [24] proposed the use of Differential Moth Flame Optimisation (DMFO) to enhance the placement of mobile sinks. This approach resulted in an improved network lifetime, although it did lead to an increase in communication overhead. Elie et al. [25] introduced an instant clustering algorithm (ICP) that increased the lifespan of the network but also resulted in substantial communication overhead during its execution. These methods emphasize the balance between the long-term sustainability of the network and the efficiency of communication. Our model accurately predicts sink positions, which reduces the need for frequent re-clustering and communication overhead. This improves both the network's lifetime and efficiency.

Al-kaseem et al. [18] devised a heuristic clustering method to enhance energy efficiency. This

Received: May 16, 2024. Revised: August 29, 2024.

method involves the utilization of multiple mobile sinks and an optimized path planning strategy. Nevertheless, the intricacy of this method and the requirement for precise real-time network data can restrict its efficacy. The ALDS-Net model we propose streamlines this process by employing a predictive methodology for sink mobility, thereby diminishing the reliance on real-time data and simplifying path planning.

Furthermore, apart from the literature review mentioned earlier, other works were also examined, such as the Adjacency Based Cell Score (ABCS) technique [23]. This technique aims to optimize the placement of mobile sinks in Wireless Sensor Networks (WSNs) by minimizing the distance traveled by the sink. However, its static nature restricts its effectiveness in dynamic environments. The Differential Moth Flame Optimization (DMFO) algorithm [24] improves the lifespan of a network by optimizing the paths of the sink nodes. However, it has drawbacks such as increased communication overhead and high computational expenses, which make it less suitable for large-scale networks. The Instantaneous Clustering Algorithm (ICP) [25] clusters nodes in order to optimize the placement of the sink and enhance the lifespan of the network. However, it results in significant communication overhead during the clustering process and lacks the ability to adapt to real-time changes. The Clustered DVGOR [26] algorithm integrates geographical opportunistic routing with clustering. It performs efficiently in static scenarios but is less effective in dynamic networks due to its inflexible clustering approach. The Stochastic Bat Algorithm [27] uses a probabilistic method to optimize routing. It achieves high packet delivery ratios but has drawbacks in terms of computational complexity and energy consumption. Hybrid Optimization [28] combines various methods to improve throughput but requires substantial computational resources and faces challenges when network conditions change. Finally, the Greedy Strategy [29] prioritizes optimizing energy usage at each hop, which reduces immediate energy consumption but may result in less optimal global routing paths and higher overall energy usage.

To summarize, previous research has investigated different approaches to enhance the positioning and routing of mobile sinks in wireless sensor networks (WSNs), each with its own disadvantages and constraints. The ALDS-Net model we propose overcomes these limitations by incorporating sophisticated deep learning methods to forecast the trajectories of mobile sinks, optimize data routing, and improve the longevity of the network. The comprehensive demonstrations and data presented in our research offer compelling evidence of these enhancements and emphasize the scientific contributions of our work.

3. Methodology

The block diagram shown in Figure 1 illustrates the three-stage WSN simulation methodology for mobile sink prediction. The first stage configures the simulation environment with essential information, variables, and mobility settings. It also sets up the network topology for later simulations, initializes the parameters of the simulation as well as those related to mobility especially needed by mobile sinks. The second stage deals with gathering data and creating models. The data is collected and trained on an ALDS-Net deep learning model that uses the ADAM optimizer and LSTM layers to model sensor node interactions with sink mobility. Once trained, this model is saved for use in simulations. The actual WSN simulation occurs in Stage 3. It involves network setup, sensor nodes addition, mobile sinks addition then running the simulation where key parameters such as dead nodes, operating nodes, contact time etc are visualized for analysis. Finally, exporting simulation data in a suitable format for further analysis or storage. This three-stage process uses data-driven modeling and visualization to create an inclusive framework of studying and understanding WSN behavior and performance.

Let us elaborate the discussion with necessary equations.

– Initialize the simulation environment by clearing any existing figures or data and starting a timer to measure the execution time of the simulation.

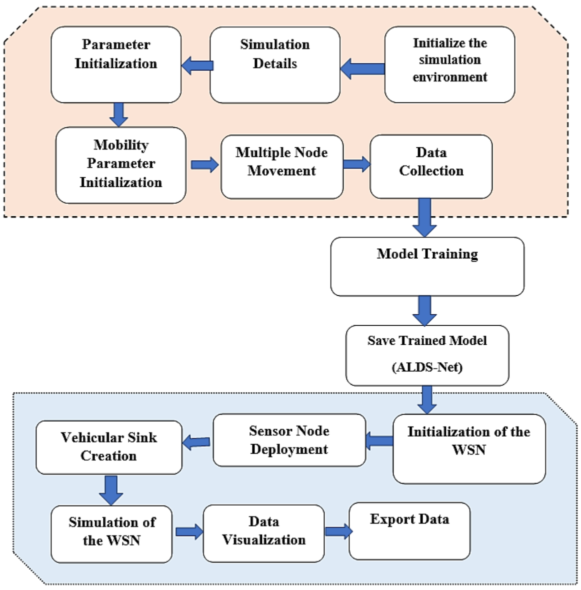

Figure. 1 Proposed Block Diagram

– Display a welcome message to let user know that WSN simulation has started.

Number of Nodes – N, Number of Mobile Sinks - SN , Number of Training Rounds – T, Number of Rounds per Simulation – R, Bits Transmitted per Packet – K, Number of Clusters – C.

- Initialize various parameters and configurations required for the WSN simulation. These parameters include values related to the network's dimensions, energy levels, mobility parameters, and simulation settings.

Let us give notations for the Initialization parameters

- Maximum Dimension of the WSN Plot - D
- Initial Node Energy – $\boldsymbol{E_{init}}$ **,**
- Energy Required for Transmission and Receiving of Packet (Transceiver Energy) – $\boldsymbol{E_{trx}}$ **,**
- Amplification Energy – $\boldsymbol{E_{amp}}$ **,** Aggregation Energy **-**$\boldsymbol{E_{agg}}$

Initialize dimension and Energy related values by following equation

$$P_{DE} = [dims, ener] = Init_{parameters}(D, E_{init}, E_{trx}, E_{agg}, E_{amp}) \quad (1)$$

- Use the `init_mobility_params` function to initialize mobility-related parameters. These parameters include:

- $\boldsymbol{M_d}$: Minimum mobility for sensor nodes
- $\boldsymbol{M_x}$: Maximum mobility for sensor nodes
- $\boldsymbol{M_{s_{min}}}$ **:** minimum mobility for sink nodes
- $\boldsymbol{M_{s_{max}}}$: Maximum mobility for sink nodes
- $\boldsymbol{V_{min}}$ Minimum distance to affirm visitation by sink nodes (mobile sinks) (in meters).

- These parameters control how nodes move within the WSN during simulation**.**

- Utilize the data Gathering function to collect data for training predictive models of mobile sink positions. This step is crucial for the accuracy of the simulation.

- Displays a message indicating the start of data gathering.

$$G(t) = "Start\ of\ Data\ Gathering\ at\ time\ t \quad (2)$$

Here, **G(t)** symbolizes the process of starting data gathering at time **t**.

- Initializes the WSN with sensor nodes and mobile sinks.

- Group the WSN into clusters.

$$Cg(ni) = kj \quad (3)$$

This equation states that a node **n_i** is assigned to a cluster **k_j** by the function **C_g**.

- Simulates multiple rounds of node movement and data collection.

$$Sr(t, n, M, K) = \begin{Bmatrix} New\ Position(n, t, M), \\ Data\ Collected(n, t, K) \end{Bmatrix} \quad (4)$$

- **t** is the current round of the simulation, ranging from 1 to **R** (the total number of rounds per simulation).
- **n** is a node in the set of all nodes **N**.
- **M** represents the mobility parameters, potentially including **M_d**, **M_x**, **M_(s_min)**, and **M_(s_max)**.
- **K** is the number of bits transmitted per packet.

- Collects data regarding the positions of mobile sinks and store it in a structured format and it is given by equation

$$Ds(t, SN, P) = \{(sni, P(t, sni)) \mid sni \in SN \quad (5)$$

- **SN** represents the set of all mobile sinks, where **SN = {sn_1, sn_2, ..., sn_{SN}}**.
- **P** is a function that gives the position of a mobile sink at time **t**, such that **P(t, sn_i)** gives the position of sink **sn_i** at time **t**.

- Depending on the availability of pretrained models, the simulation either loads the existing models or creates new predictive models for mobile sink positions.

- The predictive model is trained using the data collected during the data gathering step. Displays a message to indicate the start of model training. Configures a neural network model for regression, specifying architecture, hyperparameters, and training options. Prepares the training data extracted from the collected data. Trains separate models for predicting the X and Y positions of mobile sinks. Saves the trained models for future use and the Model for Mobile Sink Poitions can be given by following equation

$$sn_{model} = modeltraining(Ds(t, SN, P), T, S_N) \quad (6)$$

- Check if pretrained predictive models for mobile sink positions exist.

Received: May 16, 2024. Revised: August 29, 2024.

- Attempts to load a pretrained model for X-coordinates and Y-coordinates of mobile sinks.

- Initialize the WSN by creating sensor nodes as given below.

For Network Creation ($CR_{network}$- Function to create newtork):

$$WSN = CR_{network}(N, D, E\_init, R) \tag{7}$$

Building the Sensor Nodes of the WSN
Let us say
Sensor Node ID (id):

$$SN.n(i).id = i \tag{8}$$

Maximum Dimension of the WSN Plot (D):

$$dims = [x_{min}, y_{min}, x_{max}, y_{max}] \tag{9}$$

X-axis coordinates of Sensor Node (x):

$$SN.n(i).x = dims('x'_{min}) + rand(1,1) * \left(dims('x'_{max}) - dims('x'_{min})\right) \tag{10}$$

Y-axis coordinates of Sensor Node (y) :

$$SN.n(i).y = dims('y\_min') + rand(1,1) * \left(dims('y'_{max}) - dims('y'_{min})\right) \tag{11}$$

- Create vehicular sink nodes (mobile sinks) within the WSN using the following function.

For Vehicular Sink Node Creation:

$$V_{SN}(SN, D, M\_s\_min, M\_s\_max) = \{(sni, P\_init(sni), M(sni)) \mid i = 1,2,\ldots,SN\} \tag{12}$$

Where:

- **sn_i** represents the ith mobile sink.
- **P_{init}(sn_i)** denotes the initial position of sink **sn_i**, where each position is determined within the bounds of **D**, ensuring that **x_{init} ∈ [x_min, x_max]** and **y_{init} ∈ [y_min, y_max]**.
- **M(sn_i)** represents the mobility range for sink **sn_i**, bounded by **[M_{s_min}, M_{s_max}]**.

-The above equationidentifies suitable locations for mobile sinks and assigns specific attributes to these nodes and vehicular sink nodes (mobile sinks) to the WSN and returns their IDs in *ms_{ids}*. Mobile sinks are placed strategically within the WSN based on their roles and directions and its algorithm in stepwise manner is provided below.

Algorithm1: Creation of the Vehicular Sinks

```
Function createVehicularSinks(SN, dims):
  // Step 1: Determine Sink Placement
  For i from 1 to 4:
     // Calculate axis mid-point and comparison operator based on i
     If (i mod 2 == 0):     axis_mid = dims.x / 2
        op = "<=" if i <= 2 else ">="
     Else:
        axis_mid = dims.y / 2
        op = "<=" if i <= 2 else ">="
     // Find eligible sensor nodes and their distances from axis mid-point
     eligible_nodes = [(j, abs(SN.n(j).x - axis_mid if 'x' in locals() else SN.n(j).y - axis_mid)) for j in SN if SN.n(j).role == 'N' and (SN.n(j).y op axis_mid if 'y' in locals() else SN.n(j).x op axis_mid)]
     // Step 2: Find Closest Node
     I = min(eligible_nodes, key=lambda x: x[1])[0]
     // Set properties for the selected node
     SN.n(I).E = float('inf')
     SN.n(I).role = 'S'
     SN.n(I).cluster = NaN
     SN.n(I).col = "k"
     SN.n(I).size = 30
     SN.n(I).alpha = 1
     // Step 3: Set Sink Direction
     if (i mod 2 == 0):
        SN.n(I).direction = 'down' if i <= 2 else 'up'
        SN.n(I).direction_moved = 1
     else:
        SN.n(I).direction = 'left' if i <= 2 else 'right'
        SN.n(I).direction_moved = 1 if i <= 2 else -1
     // Store Sink ID
     ms_ids[1, i] = I
  Return SN, ms_ids
End Function
```

- Conduct the core simulation process, which includes multiple rounds of communication, data exchange, and data aggregation among nodes.

- The simulation process performs for a specified number of rounds and it iterates through each round and performs various operation

Let us initiate process

for round=1 to rounds 'R'

i. Reset sensor node roles to normal and sink

$$Reset_{SN} = [SN] = resetWSN(SN) \quad (13)$$

ii. Appoint Priority Nodes

$$Priority_N = [SN,\ pn_{ids}] = prioritynodeselection\begin{pmatrix} SN, ms_{ids}, sn_{model}, \\ pastdata \end{pmatrix} \quad (14)$$

iii. Simulate packet transfer and energy dissipation and Update various simulation parameters, including dead nodes, operating nodes, total energy, packets, stability period, lifetime, stability period round, lifetime round, contact time, and interconnect time.

- Now once data is collected, visualize various aspects of the simulation results. These aspects include the number of dead nodes, operating nodes, total energy, packets, contact time, and interconnect time per round.

i. Dead Nodes (DNo):

$$DNo_{new} = DNo_{old} + CDN(SN) + CDN(V_{SN}) \quad (15)$$

- Where DNo_{old} is the count of dead nodes from the previous round.
- $CDN(SN)$ and $CDN(V_{SN})$ represent the count of dead nodes in the sets SN and V_{SN}, respectively.

ii. Operating Nodes (ON):

$$ON_{new} = N - DN_{new} \quad (16)$$

Where N is the total number of nodes.

iii. Total Energy ($E_{totalnew}$)

$$E_{totalnew} = sum\big(energies(SN)\big) + sum\big(energies(V_{SN})\big) \quad (17)$$

Where energies(nodes) is a function that calculates the total energy of the given set of nodes.

iv. Packets Received (P_{rcvdn}):

$$P_{rcvdn} = sum\big(P_{rcvd}(SN)\big) + sum\big(P_{rcvd}(V_{SN})\big) \quad (18)$$

Where $P_{rcvd}(nodes)$ is a function that calculates the total received packets for the given set of nodes.

v. Contact Time (CT):

$$CT = Comp_{CT}(SN, V_{SN}) \quad (19)$$

Where $Comp_{CT}(node1\ ,node2)$ is a function that calculates the total contact time between the two sets of nodes.

vi. Interconnect Time (ICT):

$$ICT = Comp_{ICT(SN,V_{SN})} \quad (20)$$

The function $Comp_{CT}(node1\ ,node2)$ is responsible for calculating the sum of all interconnection times between two groups of nodes.

This important data can contain network settings, energy levels, movement parameters as well as any other necessary details.

- Use this function to determine how long it took to run the whole simulation.

$$Elapsed\ Time\ (\Delta t) = Ending\ Time\ (t_{end}) - Starting\ Time\ (t_{start}) \quad (21)$$

WhereΔt is the elapsed time, t_{end} is the ending time and $t_{start}$is the starting time.

The core parts of the proposed ALDS-Net model are well reflected in its name which is also known as Adaptive Moment Estimation Long Short-Term Memory DeepSink Network. This indicates about the first component as Adaptive moment Estimation and it has Adam optimization process induced in the beginning so as to eliminate the noisy gradients which are unwanted. Secondly, the term "Long Short-Term Memory (LSTM)" is added into the model where it works mainly in the situations where long term dependencies arise as this network leans the components in a sequential manner.

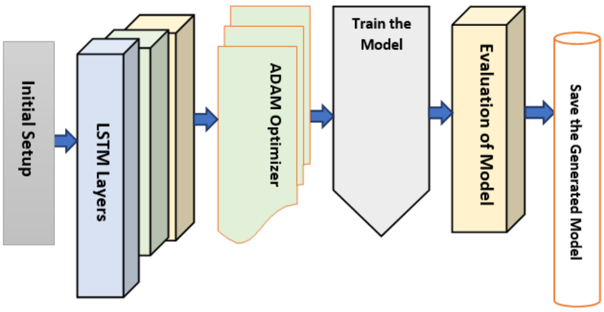

Figure. 2 Proposed Model Architecture

Additionally, 'DeepSink Network' refers to problems associated with mobile sinks within wireless sensor networks; based on deep learning approach should accurately predict direction and position of such mobile sinks. Here organization coverage and also the routing efficiency are covered.

ADAM optimizer and LSTM layers are used in combination as shown in figure 2 in a manner of self-transferred by sensor nodes interacting with mobile sinks for achieving better results with ALDS-Net than any other system. Therefore ALDS-Net can make accurate decisions thereby enhancing overall performance of a WSN. In short, 'ALDS-Net' implies breakthroughs so big they will surmount bottlenecks thus promoting growths within WSNs.

The algorithm of the proposed model ALDS-Net is provided below which contains the main components of Adam Optimization and then LSTM network which works on Long term dependencies. The algorithm is provided in step by step procedure providing insights into each stage operation in a clear manner.

Algorithm 2 : Proposed ALDS-Net Model

% Step 1: Setup
clear; close all; clc; rng('default'); % Clear environment, set seed
% Step 2: Define Model
inputSize = [features, timesteps];
layers = [sequenceInputLayer(inputSize), lstmLayer(128, 'OutputMode', 'sequence'), lstmLayer(64), fullyConnectedLayer(2), regressionLayer];
% Step 3: Training Options
options = trainingOptions('adam', 'MaxEpochs', 100, 'MiniBatchSize', 32, 'InitialLearnRate', 0.001, 'GradientThreshold', 1, 'Shuffle', 'never', 'Plots', 'training-progress');
% Step 4: Train Model
model = trainNetwork(XTrain, YTrain, layers, options);
% Step 5: Evaluate Model
YPred = predict(model, XTest); % Predict with test data
% Step 6: Save Model
save('ALDS_Net_Model.mat', 'model');

4. Result and analysis

The Dead Nodes Per Round plot in Figure 3 demonstrates how well the ALDS-Net can keep nodes alive over a long time. Until about round number 400, we see that there are no dead nodes, which means energy management is strong and the network is working well. But after 400 rounds this changes dramatically with node failures happening per round.

This plateau in Figure 4 shows a network that has a lot of operational nodes over many rounds. The result is an illustration of resilience and strong operational capacity since all the nodes are working. This part of the chart represents what can be known as the peak or highest point for any given system; everything after it will be downhill.

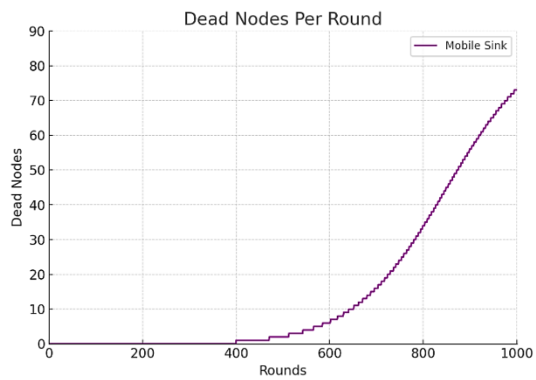

Figure. 3 Dead Nodes Per Round

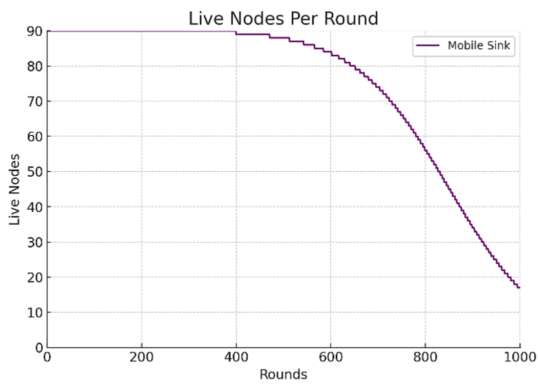

Figure. 4 Live Nodes Per Round

Received: May 16, 2024. Revised: August 29, 2024.

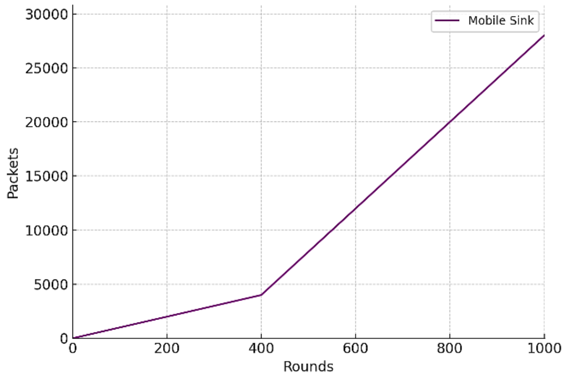

Figure. 5 Packets Transmission per Round

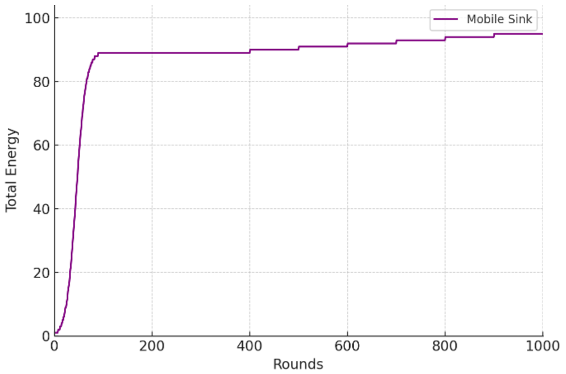

Figure. 6 Total Energy per Round

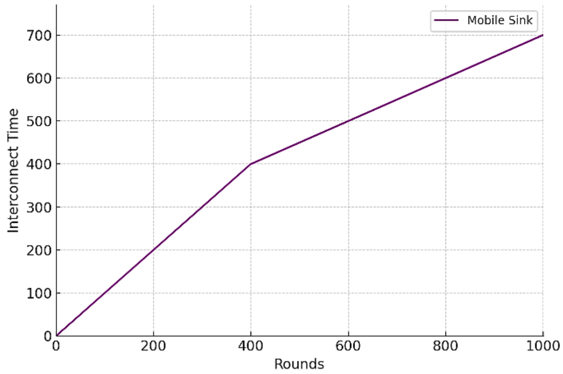

Figure. 7 Interconnect Time per Round

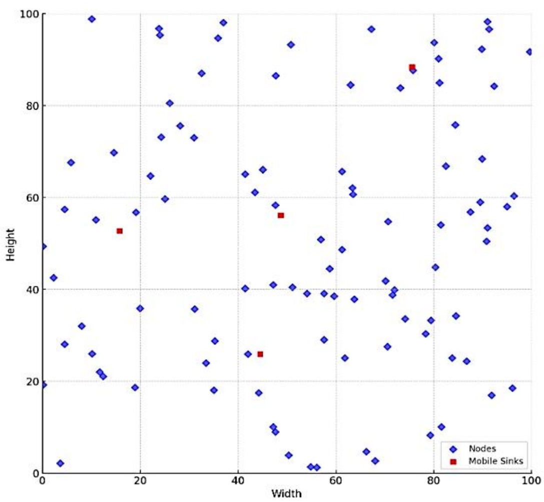

Figure. 8WSN Mobility at the initial rounds

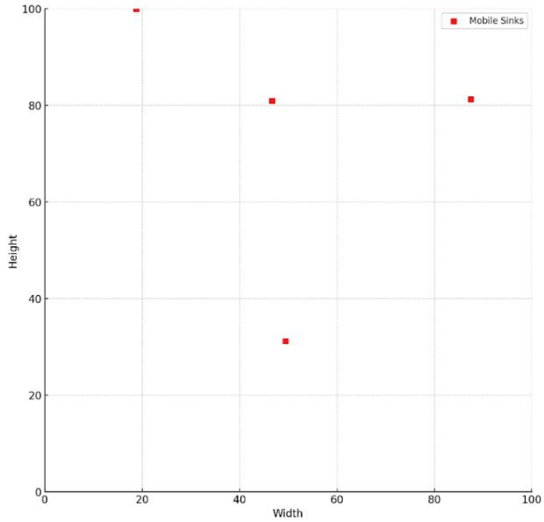

Figure. 9 WSN Mobility of 4 Mobile Sinks

In the given plot, Figure 5, it can be seen that as time goes by, more packets are being sent per round. The chart shows a rising trend of activity in the network which means that this system can process larger amounts of information with each passing moment.

In Figure 6, the total energy usage each round in a network is shown. This shows that after an initial increase, energy consumption stabilizes significantly. The graph flattens at about 90 units of energy meaning the network can achieve power parity. Between rounds 400 and 1000 inclusive there is no change in energy use this implies that the network can sustain itself without increasing its budget for power consumption over time.

In this network, the duration of connection per round is shown by Figure 7. It implies that as time goes by, the time taken to connect linearly increases. This duration of interconnectivity starts at zero and then rises up to about 700 units at the 1000th round. The graph's linearity means that there will be an expected growth in how long nodes stay connected together hence showing that more communication can take place through the network over an extended period without any abrupt spikes or dips.

The figure 8 is about how a wireless sensor network (WSN) operates at its initial stages with respect to mobility where nodes are represented by blue diamonds and mobile sinks by red squares. Four mobile sinks are spread over one hundred units wide by one hundred units high area together with a hundred nodes which are uniformly distributed across it implying evenness of coverage and connection throughout the region. Figure 9 shows how four mobile sinks moved over a wireless sensor network during 1000 rounds on a unit grid measuring 100x100. The red squares represent the positions of

Received: May 16, 2024. Revised: August 29, 2024.

Table 1. Life Time assessment

Techniques used	Life Time (Secs)
Adjacency Based Cell Score (ABCS) [23]	111
DMFO [24]	102.86
ICP [25]	107.135
Proposed Method (ALDS-Net)	498.5

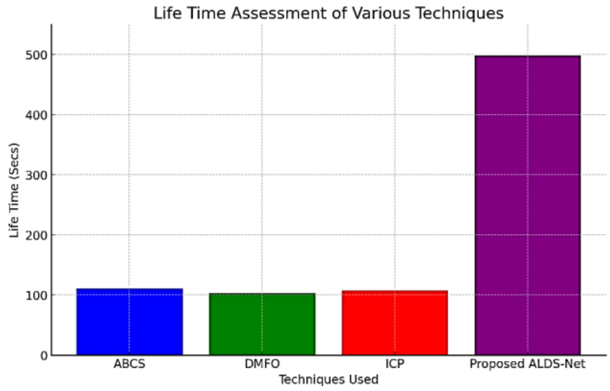

Figure. 10 Comparison of Lifetime for 400 Nodes

Table 2. Average Packet Delivery Ratio

Techniques used	PDR %
Clustered DVGOR [26]	85
Stochastic Bat [27]	92
Greedy [22]	91.3
Proposed Method (ALDS-Net)	93.2

the mobile sinks which are deployed throughout the network for data collection efficiency and coverage maximization. Placing them strategically after heavy activity on the network shows how adaptable it is in terms of reallocating its resources where necessary.

Table 1 provides an assessment on the lifetime of different techniques employed in wireless sensor networks with respect to how long they can function effectively before their nodes fail. Three methods are shown by this table; ALDS-Net, proposed method and their lifetimes given in seconds. Adjacency Based Cell Score method lasts for 111s, while DMFO (Differential Mutation Flower Optimization) lasts for 102.86s and ICP (Inductive Charging Protocol) lasts for 107.135s . Compared to these numbers, ALDS-Net has been able to operate up to 498.5 seconds more. ALDS-Nets' significant increase in operational lifetime points out at enhanced efficacy as well as management of resources within networks such as this one here being considered.

The bar plot in Figure 10, based on data from Table 1, compares lifetimes of different network techniques. The Adjacency Based Cell Score, DMFO, ICP, and the proposed ALDS-Net method were used as techniques. The lifetime of each technique is indicated below: Adjacency Based Cell Score at 111 seconds, DMFO at 102.86 seconds, ICP at 107.135 seconds and proposed ALDS-Net method at 498.5 seconds. It can be seen clearly from this diagram that ALDS-Net method has longer life than any other methods.

In wireless sensor networks, Table 2 presents different techniques used with their average packet delivery ratio (PDR) which is a fundamental evaluative measure for the efficiency of transmitting packets from source to destination. Some of the techniques highlighted are Clustered DVGORs that have 85% PDR and show good performances in delivering packets. Stochastic Bat technique and Greedy technique perform better than this with 92% and 91.3% PDRs respectively thus indicating more stable transmission of packets across the network.

The proposed ALDS-Net method has a better performance than all these methods as it achieves PDR of 93.2%, which is the highest among them. This shows that ALDS-Net can route most packets correctly to their intended destinations because of its high delivery rate. Also it means that this algorithm has stronger routing abilities than any other one included here. Such high PDR percentage indicates great improvement on reliability and efficiency in communication within networks by ALDS-Net thereby positioning it as one of the best solutions for critical applications demanding higher rates of packet delivery throughputs.

The figure 11 bar chart shows different methods' average packet delivery ratio (PDR). The Proposed Method (ALDS-Net) has the highest PDR of 93.2%. This diagram makes a clear comparison between each method's PDR performances and underscores that ALDS-Net is better at ensuring successful packet delivery in wireless sensor networks.

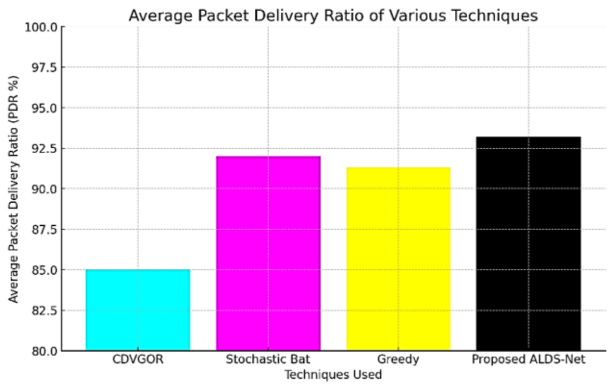

Figure. 11 Comparison of PDR

Received: May 16, 2024. Revised: August 29, 2024.

Table 3. Throughput

Techniques used	Throughput (bps)
ACO [29]	1150
Stochastic Bat [27]	1450
Hybrid Optimization [28]	1285
Proposed Method (ALDS-Net)	1620

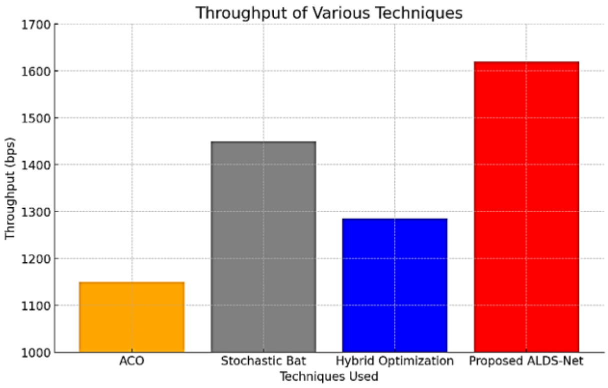

Figure. 12 Comparison of Throughput

3 compares the throughput capacities of different routing techniques in a wireless sensor network, measured in bits per second (bps). The Ant Colony Optimization (ACO) technique attains a capable throughput of 1150 bps thus setting up the foundation for network performance improvement. Comparatively, Stochastic Bat algorithm performs far much better with 1450 bps throughput while Hybrid Optimization gives us1285bps. However, the proposed method ALDS-Net has shown great improvement in terms of throughput which peaked at 1620 bps. This is very high given that the size of data packets is only about 200 KB. It therefore indicates that ALDS-Net can deliver data fast and efficiently within the network. Figure 12 shows the data of throughputs from Table 3 highlighting different routing methods by means of their data forwarding capabilities. ALDS-Net proposed method records the highest throughput (1620 bps) followed by Stochastic Bat technique having 1450bps, Hybrid Optimization with 1285 bps and ACO at 1150 bps among others. Such a diagram clearly demonstrates that ALDS-Net has competitive advantage over other approaches when it comes to smoothness of information traffic within wireless sensor networks.

5. Conclusion and summary

This research study examined Wireless Sensor Networks (WSNs) through a three-stage methodology for the prediction of mobile sink. This work includes specifying the network topology, configuring simulation parameters, and tuning mobility settings particularly for mobile sinks which serve as our groundwork for subsequent simulations to ensure accurate representation of WSN scenarios. The second stage of our methodology involves data collection and modeling. Here we collect required data then apply ALDS-Net deep learning model which consists ADAM optimizer and LSTM layers. This provides accurate views about sensor node interactions with sink mobility hence enhancing predictability. Trained models are saved cautiously for future simulations. Stage 3 is where actual WSN simulation occurs; creation of network setup, sensor nodes deployment, mobile sinks deployment, simulation running. It visualizes key parameters such as active nodes, dead nodes, contact time etc., thus enabling deeper understanding on how networks behave. It is observed that in a 400-node WSN over 750 simulation rounds ALDS-Net lifetime was 498.5 seconds while Adjacency Based Cell Score had a Lifetime of less than or equal to 15 seconds (DMFO), ICP lifetime was less than or equal to 6 seconds. These results indicate that if implemented well this technique could lead to higher sustainability rates within networks compared to other alternatives currently available. In addition, the findings open up new areas for investigation in relation with WSNs. However, it should be noted that adaptive routing may still pose some challenges even after applying machine learning methods like ALDS-Net which is a clear indication that more needs to be done towards developing energy efficient protocols for data aggregation within these kinds of networks.

Notations List:

- *D*: Maximum Dimension of the WSN Plot.
- *E*init: Initial Node Energy.
- *E*trx: Energy Required for Transmission and Receiving of Packet (Transceiver Energy).
- *E*agg: Aggregation Energy.
- *E*amp: Amplification Energy.
- *Md*: Minimum mobility for sensor nodes.
- *Mx*: Maximum mobility for sensor nodes.
- *Ms*min: Minimum mobility for sink nodes (mobile sinks).
- *Ms*max: Maximum mobility for sink nodes (mobile sinks).
- *V*min: Minimum distance to affirm visitation by sink nodes (mobile sinks).
- $G(t)$: Start of Data Gathering at time tt.
- *N*: Number of Nodes.
- *SN*: Number of Mobile Sinks.
- *T*: Number of Training Rounds.
- *R*: Number of Rounds per Simulation.
- *K*: Bits Transmitted per Packet.
- *C*: Number of Clusters.
- *D*No: Number of Dead Nodes.

- *O*No: Number of Operating Nodes.
- *E*total_new: Total Energy.
- *P*rcvd: Packets Received.
- CT: Contact Time.
- ICT: Interconnect Time.
- Δ*t*: Elapsed Time.

Conflicts of Interest

The authors affirm that they are aware of no personal or financial conflicts of interest that might have affected the research described in this paper

Author Contributions

Kaumudi Keerthana - Conceptualization, Data curation Formal analysis, Methodology, Software Writing, original draft, investigation, resources.

A. Mahesh Babu –Writing, review and editing, visualization, supervision, project administration.

References

[1] C. Ma, H. Wang, and Q. Li, "A Connectivity Aware Approximation Algorithm for Relay Node Placement in Wireless Sensor Networks", *IEEE Sensors Journal*, Vol. 16, No. 2, pp. 515-528, 2016.

[2] C. Bergonzini, D. Brunelli, and L. Benini, "Comparison of Energy Intake Prediction Algorithms for Systems Powered by Photovoltaic Harvesters", *Microelectron. J.*, Vol. 41, pp. 766-777, 2010.

[3] S. Kosunalp, "A New Energy Prediction Algorithm for Energy-Harvesting Wireless Sensor Networks with Q-Learning", *IEEE Access*, Vol. 4, pp. 5755-5763, 2016.

[4] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless Sensor Network Survey", *Comput. Netw*, Vol. 52, pp. 2292-2330, 2008.

[5] J.-S. Lee and T.-Y. Kao, "An Improved Three-Layer Low-Energy Adaptive Clustering Hierarchy for Wireless Sensor Networks", *IEEE Internet of Things Journal*, Vol. 3, No. 6, pp. 951-958, 2016.

[6] A. Saranya, R. Senthil Kumaran, and G. Nagarajan, "Enhancement of Network Lifetime Using Tree-Based Routing Protocol in Wireless Sensor Networks", In: *Proc. of the IEEE Sponsored International Conference on Electronics and Communication Systems (ICECS '15)*, Vol. 3, pp. 1392-1396, 2015.

[7] W.R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient Communication Protocol for Wireless Microsensor Networks", In: *Proc. of the 33rd Annual Hawaii International Conference on System Sciences*, pp. 1-10, 2000.

[8] S. Yu, B. Zhang, C. Li, and H.T. Mouftah, "Routing Protocols for Wireless Sensor Networks with Mobile Sinks: a Survey", *IEEE Commun. Mag*, Vol. 52, No. 7, pp. 150-157, 2014.

[9] L. Guezouli, K. Barka, S. Bouam, and A. Zidani, "Implementation and Optimization of RWP Mobility Model in W.S.N.s under Tossing Simulator", *Int. J. Commun. Netw. Inf. Secure*, Vol. 9, No. 1, 2017.

[10] S. Gao, H. Zhang, and S. Das, "Efficient Data Collection in Wireless Sensor Networks with Path-constrained Mobile Sinks", *IEEE Trans. Mob. Comput*, Vol. 10, No. 4, pp. 592-608, 2011.

[11] R.-S. Chang, S.-H. Wang, S.-L. Tsai, and W.-P. Yang, "Planning Rendezvous using the Halin Graph in Wireless Sensor Networks", *IET Wirel. Sens. Syst*, Vol. 2, No. 3, pp. 222-229, 2012.

[12] M.M.S. Sarwar and P. Chatterjee, "Optimal Sink Placement in Wireless Sensor Networks to Increase Network Performance", In: *Proc. of Industry Interactive Innovations in Science*, Engineering and Technology, pp. 423-433, 2018.

[13] K. Naween and D. Dinesh, "Mobile Data Sink-based Time-constrained Data Collection from Mobile Sensors: a Heuristic Approach", *IET Wirel. Sens. Syst*, Vol. 8, No. 3, pp. 129-135, 2018.

[14] S. Maurya, V.K. Jain, and D.R. Chowdhury, "Delay Aware Energy Efficient Reliable Routing for Data Transmission in Heterogeneous Mobile Sink Wireless Sensor Network", *J. Netw. Comput. Appl*, Vol. 144, pp. 118-137, 2019.

[15] C. Thomson, I. Wadhaj, and Z. Tan, "Mobility Aware Duty Cycling Algorithm (MADCAL): a Dynamic Communication Threshold for Mobile Sink in Wireless Sensor Network", *Sensors*, Vol. 19, No. 22, pp. 4930-4945, 2019.

[16] J. Anees, H.C. Zhang, B.G. Lougou, S. Baig, and Y.G. Dessie, "Delay Aware Energy-efficient

Opportunistic Node Selection in Restricted Routing", *Comput. Netw*, Vol. 181, 107536, 2020.

[17] D. Sethi, "An Approach to Optimize Homogeneous and Heterogeneous Routing Protocols in WSN using Sink Mobility", *MAPAN*, Vol. 35, No. 2, pp. 241-250, 2020.

[18] B.R. Al-Kaseem, Z.K. Taha, S.W. Abdulmajeed, and H.S. Al-Raweshidy, "Optimized Energy-efficient Path Planning Strategy in WSN with Multiple Mobile Sinks", *IEEE Access*, Vol. 9, pp. 82833-82847, 2021.

[19] C. Punriboon et al., "Fuzzy Logic-based Path Planning for Data Gathering Mobile Sinks in WSNs", *IEEE Access*, Vol. 9, pp. 96002-96020, 2021.

[20] P.P. Jadhav and S.D. Joshi, "Atom Search Sunflower Optimization for Trust-based Routing in Internet of Things", *Int J Numer Model*, Vol. 34, No. 3, e2845, 2021.

[21] U.K. Lilhore, O.I. Khalaf, S. Simaiya et al., "A Depth-controlled and Energy-efficient Routing Protocol for Underwater Wireless Sensor Networks", *International Journal of Distributed Sensor Networks*, Vol. 18, No. 9, 2022.

[22] S. Hao, Y. Hong, and Y. He, "An Energy-Efficient Routing Algorithm Based on Greedy Strategy for Energy Harvesting Wireless Sensor Networks", *Sensors (Basel)*, Vol. 22, No. 4, 1645, 2022.

[23] A.R. Aravind and C. Rekha, "Optimal Placement of Mobile Sink in Wireless Sensor Network Using Adjacency-Based Cell Score Factor", *International Journal of Intelligent Engineering & Systems*, Vol. 14, No. 2, pp. 22-32, 2021, doi: 10.22266/ijies2021.0430.03.

[24] S. Sapre and S. Mini, "A Differential Moth Flame Optimization Algorithm for Mobile Sink Trajectory", *Peer-to-Peer Networking and Applications*, pp. 1-14, 2020.

[25] T. Elie, A. Zaraket, and E. Tonye, "A Reliable and Efficient Path Discovery Method for Mobile Sink based Wireless Sensor Networks", *International Journal of Computer Applications*, Vol. 176, No. 25, pp. 17-22, 2020.

[26] K.-P. Shih, B. Ragavi, V. Baranidharan, and K. Ramash Kumar, "A Novel Hybridized Cluster-Based Geographical Opportunistic Routing Protocol for Effective Data Routing in Underwater Wireless Sensor Networks", *Journal of Electrical and Computer Engineering*, Vol. 2023, pp. 5567483, 2023.

[27] G. Diraco, J. Paruvathavardhini, and B. Sargunam, "Stochastic Bat Optimization Model for Secured WSN with Energy-Aware Quantized Indexive Clustering", *Journal of Sensors*, Vol. 2023, pp. 4237198, 2023.

[28] A. Vinitha, M.S.S. Rukmini, and Dhirajsunehra, "Secure and Energy Aware Multi-hop Routing Protocol in WSN using Taylor-based Hybrid Optimization Algorithm", *Journal of King Saud University - Computer and Information Sciences*, Vol. 34, No. 5, pp. 1857-1868, 2022.

[29] M. Rathee, S. Kumar, A.H. Gandomi, D. Kumar, B. Balusamy, and R. Patan, "Ant Colony Optimization Based Quality of Service Aware Energy Balancing Secure Routing Algorithm for Wireless Sensor Networks", *IEEE Transactions on Engineering Management*, Vol. 68, No. 1, pp. 170-182, 2021.