Project Report For CS667:
INTRODUCTION TO INTERNET OF THINGS & ITS INDUSTRIAL APPLICATIONS
2024-2025 Semester I
Project Title: ACO with Erasure Coding in IoT

Team members: Maurya Aryan Swaminath, Depanshu Sahu, Aditya Ajmera
Member emails: {aryanms21, depanshus21, adityaa21}@iitk.ac.in
IIT Kanpur

1. Introduction:

In the fast-evolving world of the Internet of Things (IoT), where billions of devices connect seamlessly across fields like healthcare, smart cities, agriculture, and industry, ensuring data reliability has become a crucial challenge. As IoT networks expand, they face mounting pressure to remain scalable, reliable, and energy-efficient, particularly in resource-constrained environments where devices are often vulnerable to failures.

After reading the following research papers [1], [2], [3], [4], [5], [6], [7], [8], [9], [10] and their comparing them thoroughly, we thought of combining Ant Colony Optimization (ACO) and erasure coding (EC). ACO, inspired by the way ants find the shortest routes to food, adapts dynamically to network changes, making it a promising approach for selecting optimal routes in IoT networks. Erasure coding (EC) complements this by enhancing data reliability—splitting each message into smaller fragments and adding redundancy so that data can still be reconstructed even if parts are lost during transmission.

This project focuses on using ACO to ensure efficient routing, while EC safeguards data integrity, providing a solution that is both resilient and adaptable. By integrating these techniques, we aim to enhance the scalability and reliability of IoT communication networks, paving the way for robust, real-world applications where dependable data transmission is essential.

2. Tasks:

1. **Random Network Topography Generation:** To simulate a real-world IoT environment, we generated a random network topography. This allowed us to mimic how IoT nodes might be deployed in practice and created a realistic setting for testing our solution. Figure 1 shows the random topography of nodes generated through the Python code.

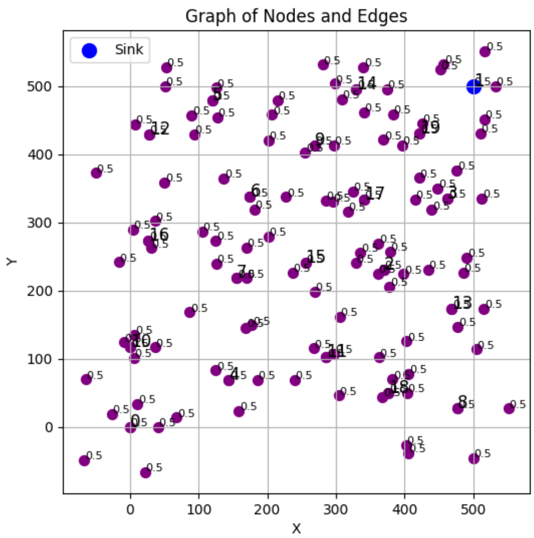

Figure 1: Network Topology

2. **Cluster Head Selection Mechanism:** To make the network energy-efficient, clusters with Cluster Heads(CHs) were made. Cluster heads were chosen dynamically, based on their energy levels and connectivity, to reduce the workload on individual nodes and streamline communication. Figure 2 shows the topology of randomly scattered cluster with their nodes and cluster heads. Figure 3 shows connection between neighbors that have been formed between Cluster Heads(CHs).

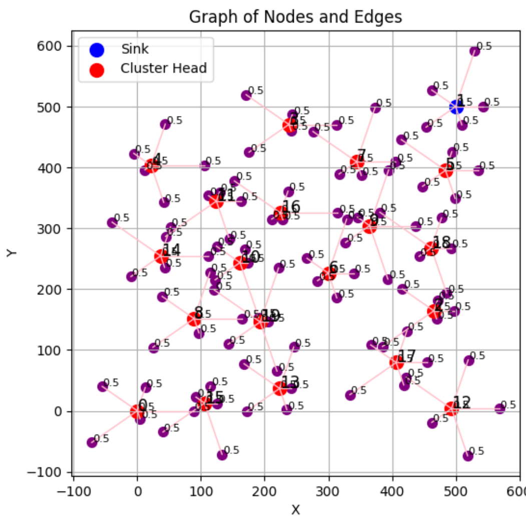

Figure 2: Network Topology with Cluster Heads

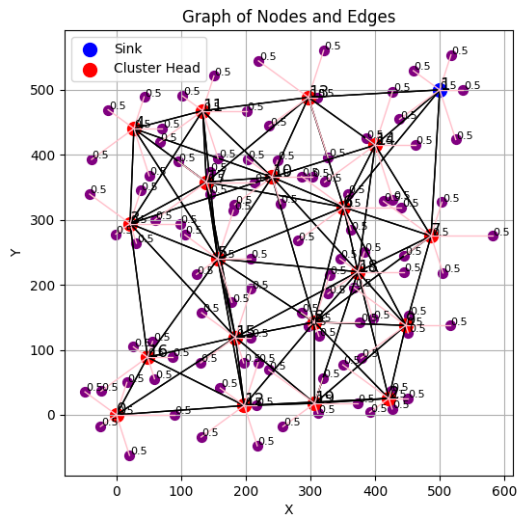

Figure 3: Network connection between Cluster Heads

3. **Trivial ACO Implementation:** We started by building a basic Ant Colony Optimization (ACO) algorithm. This version served as a benchmark for testing our improved ACO algorithm

4. **Energy-Efficient Next-Hop Selection:** To make the routing smarter and more energy-conscious, we designed a mechanism for choosing the next hop in a way that balances energy usage and ensures the IoT devices can operate for longer periods. It considers key factors like remaining battery life, signal quality, and proximity.

5. **Adding Erasure Coding During Data Transmission:** To improve reliability, we integrated erasure coding into the data transmission process. It involved breaking the data into smaller, redundant pieces, so even if some fragments were lost, the original data could still be reconstructed at the destination.

6. **Comparison Over Various Metrics:** It will help determine the performance of our combined network in different situations. An analysis of network performance over various situations helps us understand how it performs under varied conditions.

3. Proposed Solution:

The proposed solution **EC-ACO** integrates Ant Colony Optimization (ACO) with erasure coding to tackle the challenges of routing, energy efficiency, and data reliability in IoT networks. Figure 4 shows the flow diagram of the improved ACO algorithm.

1. **Erasure Coded Ant Colony Optimization (EC-ACO) for Routing:** Inspired by the natural behavior of ants, ACO is used to discover optimal routes in the network. The following is a step-by-step breakdown of the Erasure Coded Ant Colony Optimization (EC-ACO) algorithm:

   (a) **Initialize Cluster Heads**: Start the process by initializing the cluster heads, which may serve as routing points in the network for efficient data transfer.

   (b) **Generate Ants from Source**: Create a set of "ant"agents at the source node. These ants will explore paths in the network to find an optimal route to the sink (destination) node.

   (c) **Select Next Hop and Move Forward**: Each ant chooses the next hop (node) based on certain criteria (e.g. pheromone levels or distance). The ant then moves to the selected node, progressing toward the sink node.

   (d) **Check if Sink Node is Reached**: If the ant reaches the sink node, proceed to step (e). If not, return to step (c) and continue moving forward by selecting the next hop.

   (e) **Retrace the Path Back to Source**: Once the ant reaches the sink node, it retraces its path back to the source node. This step helps reinforce the chosen path with pheromone, marking it as a potentially efficient route.

   (f) **Check if K Ants are Generated**: Verify if the required number of ants (K) has been generated and completed their paths. If not, return to step (b) to generate more ants. If K ants are generated, proceed to the next step.

   (g) **Update Pheromone Matrix**: Update the pheromone levels along the paths taken by the ants based on their performance (e.g. shorter paths may receive more pheromone). This update helps optimize future path selections by guiding ants toward promising routes.

   (h) **Send Erasure Coded Data**: Once optimal paths are established, transmit the erasure-coded data along these paths. Erasure coding adds redundancy, allowing data recovery even if parts of it are lost during transmission.

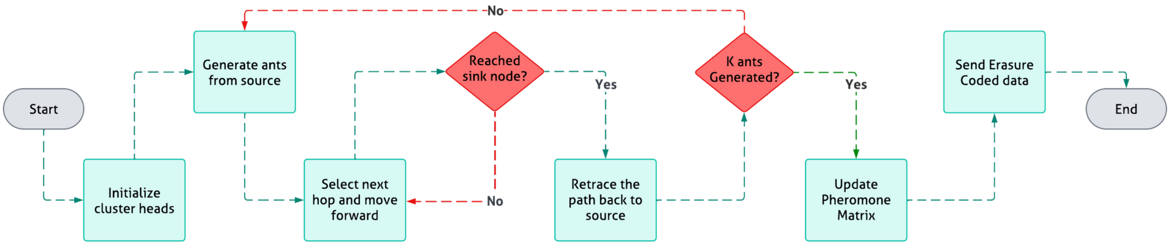

Figure 4: EC-ACO Routing

2. **Energy-Efficient Next-Hop Selection:** The routing process prioritizes energy efficiency by selecting the next hop based on:

    - Residual energy of neighboring nodes to prevent energy depletion.
    - Relative energy of the candidate node compared with the current node.
    - Distance to the sink node to minimize transmission power.
    - Distance to the candidate node to which the next hop may occur.
    - Number of neighbors of the candidate node.

    This formula gives next hop selection probability for an ant

$$nextHop_k = \frac{p_k^\alpha n_k^\beta}{\sum_{i=1}^{n} p_i^\alpha n_i^\beta}$$

    where:
    $\alpha, \beta = constants$
    p = pheromone value of edge

$$n = \frac{1}{d_{i,j} + d_{j,sink}} \cdot \frac{E_j}{E_i} \cdot \frac{N_j}{maxN}$$

    where:
    $d_{i,j}$ = Distance between candidate node and transmitter node
    E_j = Energy level of node j
    N_j = Number of neighbors of node j
    This approach prolongs the network lifetime while maintaining robust communication.

3. **Pheromone update mechanism:** We have used the same pheromone update mechanism as used in [3]. We use $\tau_{ij}(t)$ to denote the pheromone concentration between node i and node j at time t. Additionally, t is the iteration counter. Moreover, the pheromone volatilizes with time. After all ants have completed a path search, the pheromone matrix should be updated. The global pheromone update rule is presented as follows:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t),$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^{q} \Delta\tau_{ij}^k(t),$$

$$\Delta\tau_{ij}^{k}(t) = \begin{cases} \frac{Q}{L_k}, & \text{if } (i,j) \text{ is in the tour by ant } k \\ 0, & \text{otherwise} \end{cases}$$

where $0 < \rho < 1$ is the evaporation parameter, q is the total number of ants, Q is the total amount of pheromone, and L_k is the total length of the path that the k-th ant passes during this time.

To prevent excessively high or low pheromone concentrations, which may cause the algorithm to stagnate or fail to attract other ants, the method from the max–min ant system (MMAS) is employed to limit the pheromone values:

$$\tau_{ij}(t+1) = \begin{cases} \tau_{\max}, \text{if } \tau_{ij}(t+1) > \tau_{\max} \\ \tau_{\min}, \text{if } \tau_{ij}(t+1) < \tau_{\min} \\ (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t), \text{otherwise} \end{cases}$$

where $\tau_{\max}$ and $\tau_{\min}$ represent the maximum and minimum pheromone values, respectively. Limiting the pheromone values helps avoid stagnation and improves the global convergence of the algorithm.

4. **Cluster Head Selection Mechanism:** To make the network more organized and energy-efficient, we used the cluster-based approach mentioned in [3]. Cluster heads were chosen dynamically, based on their energy levels and connectivity, to distribute the workload among the cluster nodes. When the energy of the current cluster head falls below half the energy of the other cluster members, an algorithm is run to find the new Cluster Head(CH). Cluster heads (CHs) were selected based on:

   - Residual energy of nodes to ensure longevity.
   - Proximity to cluster members for efficient communication.
   - Distance of the candidate node from the sink node.

   Cluster head is selected on the basis of I value which is given by the equation:

$$I_i = \frac{\lambda \cdot E_i}{d_{i,sink} \cdot d_{avg}}$$

   where:
   λ : constant
   E_i : Energy level of node i
   $d_{i,sink}$: Distance between node i and sink
   d_{avg} : Average distance between neighboring nodes

$$new\_CH = argmax_{i \in N}(I_i)$$

   CHs facilitate intra-cluster communication and act as intermediaries for inter-cluster routing.

5. **Erasure Coding for Data Reliability:** Erasure coding is implemented to enhance fault tolerance during data transmission. The methodology includes:

- Dividing data packets into **k** fragments and generating **r** redundant fragments for reconstruction.
- Distributing the fragments across diverse paths selected by ACO to reduce dependency on any single path.
- Reconstructing data at the destination even in cases of partial fragment loss.

We compared Reed-Solomon and Cauchy Reed-Solomon techniques for our algorithm.

6. **Energy Calculation Model:** We considered both transmission and receiving energy calculations(as given in [4]), defined as follows:

$$E_{TX}(l, d) = l \times E_{elec} + l \times \epsilon_{mp} \times d^4$$

$$E_{RX}(l, d) = l \times E_{elec}$$

- $E_{TX}(l, d)$: Transmission Energy
- $E_{RX}(l, d)$: Receiving Energy
- E_{elec}: Transmission Energy per bit
- l: Number of bits in the packet
- ϵ_{mp}: Amplification Energy
- d: Distance between nodes

In the equations above:

- **Transmission Energy (E_{TX})** accounts for the energy required to transmit a data packet. It includes two components:
  (a) The energy consumed per bit ($l \times E_{elec}$).
  (b) The amplification energy is proportional to the distance raised to the power of 4 (i.e. $l \times \epsilon_{mp} \times d^4$), assuming a multi-path fading model.
- **Receiving Energy (E_{RX})** is the energy consumed to receive a packet, which depends only on the number of bits and the energy consumed per bit ($l \times E_{elec}$).

7. **Performance Evaluation:** The performance of the proposed solution was evaluated against the energy efficient ACO called **ACOCR** implemented in [3]. Key metrics include:

- **Packet Loss Ratio:** Comparison of packet loss under lossy network conditions will demonstrate the effectiveness of erasure coding in maintaining data integrity.
- **Network Lifetime:** It will help determine which network will have prolonged operation and hence a higher lifetime.
- **Energy Consumption:** Analysis of energy usage per transmission highlighted the sustainability of the proposed solution.
- **Comparing Different Erasure Coding Techniques:** This comparison will provide valuable insights into selecting the best erasure coding method for IoT networks.

4. Results:

1. **Reliable Data Packet Comparison:** The integration of Cauchy Reed-Solomon Erasure Coding significantly reduced packet loss under varying network conditions. The number of successful packets received increases relatively as the packet loss ratio increases compared to the trivial ACO network. In scenarios with packet loss rates ranging from 0% to 20% (figure 7), the data reliability improved notably compared to simple ACO implementations. For figures 5, 6, 7 we took erasure coding parameters k=10, r=12. For figure 8 the parameters are k=100, r=20.

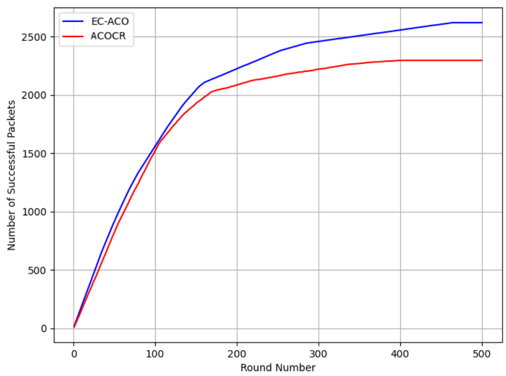

Figure 5: Data Success Ratio with each node packet loss probability 0-10%

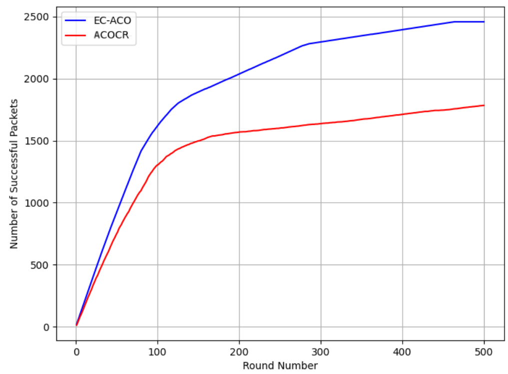

Figure 6: Data Success Ratio with each node packet loss probability 0-15%

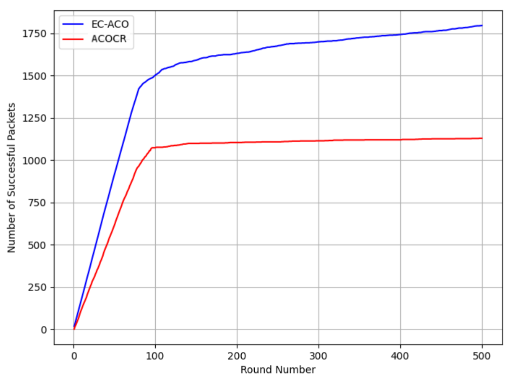

Figure 7: Data Success Ratio with each node packet loss probability 0-20%

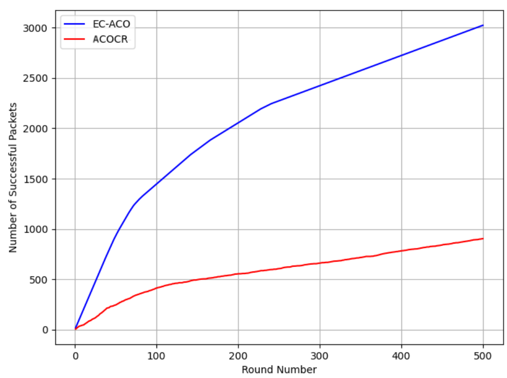

Figure 8: Data Success Ratio with each node packet loss probability 0-10% (with k=100)

2. **Network Lifetime Comparison:** By optimizing cluster head selection and balancing energy usage among nodes, the algorithm successfully prolonged the operational duration of the network. But as we require to send more number of packets after erasure coding, we can see from the results in figures 9, 10, 11, 12, (all having parameters k=12, r=2) the overall network lifetime of the EC-ACO is less than trivial ACO, but the difference is comparable.

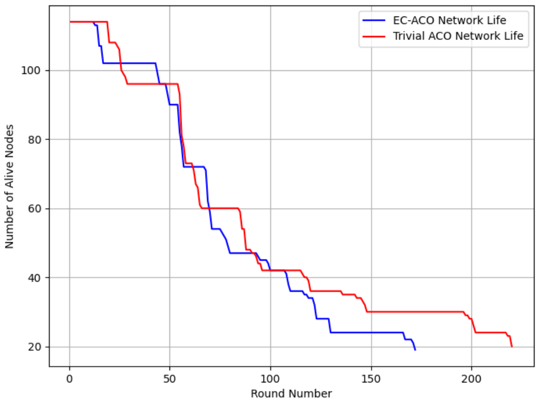

Figure 9: Network Life with each node packet loss probability 0-10%

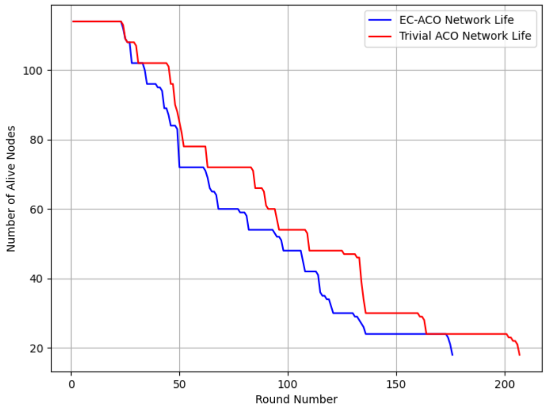

Figure 10: Network Life with each node packet loss probability 0-15%

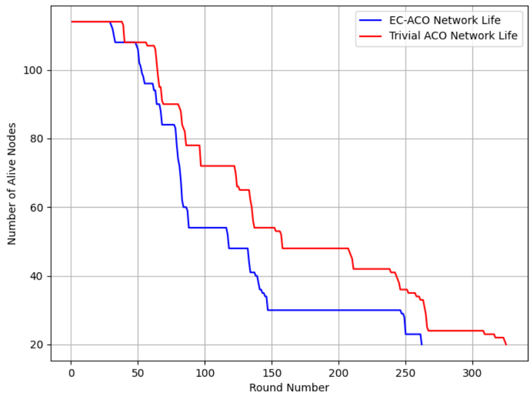

Figure 11: Network Life with each node packet loss probability 0-20%

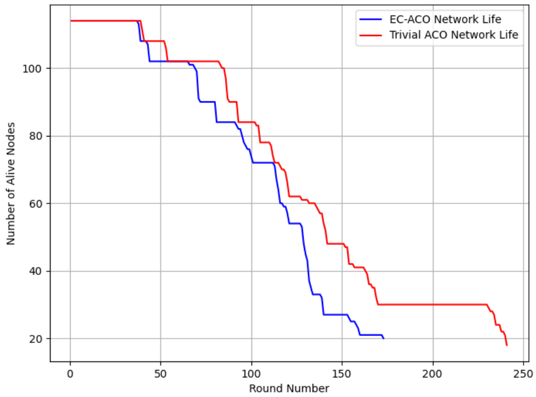

Figure 12: Network Life with each node packet loss probability 10-20%

3. **Erasure Coding Energy Consumption Analysis:** We have used Cauchy Reed-Solomon Erasure Coding in our algorithm, compared to the Reed-Solomon EC Algorithm that uses multiplication and addition operations, the Cauchy Reed-Solomon EC Algorithm uses only XOR-based operations which significantly reduces energy consumed for the Erasure Coding.
   Assumptions used for Energy Usage calculations in Encoding and Decoding Processes (as mentioned in [11]):

   - XOR operation consumes approximately 0.1 pJ.
   - Addition operation consumes approximately 0.1 pJ.
   - Multiplication operation consumes approximately 3 pJ.

Process	**XORs**	**ADD**	**MUL**	**Energy Used (µJ)**
Reed-Solomon Encoding	798,840	798,840	798,840	2.556288
Reed-Solomon Decoding	866,840	866,840	866,840	2.773888
Cauchy Reed-Solomon Encoding	800,000	0	0	0.080000
Cauchy Reed-Solomon Decoding	800,000	0	0	0.080000

Table 1: Operations and Energy Usage for Different Encoding and Decoding Processes

5. Conclusion:

Integrating Ant Colony Optimization (ACO) with Cauchy Reed-Solomon Erasure Coding effectively addresses the dual challenges of ensuring data reliability and enhancing energy efficiency in IoT networks. Cauchy Reed-Solomon Erasure Coding approach significantly mitigates packet loss, enhancing data reliability even under high packet loss conditions at the trade-off of sending more packets and hence lower overall network lifetime. Simultaneously, ACO's adaptive routing mechanism discovers energy-efficient paths and dynamically responding to changes in network topology. Cluster head selection, guided by residual energy distribution, balances energy consumption across nodes, thereby increasing the network's lifespan. Furthermore, the extent of erasure coding can be fine-tuned to match network conditions, providing a customizable approach to managing packet loss and energy efficiency in IoT environments.

6. Link to source code: github.com/aryanmaurya383/ACO-with-Erasure-Coding

7. Contribution of Each Team Member:

Task Done By	**Team Member(s)**
Random Network Topography Generation	MA, AA
Trivial ACO Implementation	MA, DS
Energy-Efficient Next-Hop Selection	DS
Erasure Coding During Data Transmission	MA, AA
Erasure Coding Techniques	AA
Cluster Head Selection Mechanism	DS
Comparing Data Loss in a Lossy Network	MA
Comparing Network Lifetime	MA, AA
Presentation	MA, DS, AA
Final Report	MA, DS, AA

Table 2: Task Distribution

Member	**Full Name**	**Final Contribution (%)**
MA	Maurya Aryan Swaminath	33.33
DS	Depanshu Sahu	33.33
AA	Aditya Ajmera	33.33

Table 3: Contribution

References

[1] Zongshan Wang, Hongwei Ding, Bo Li, Liyong Bao, and Zhijun Yang. An energy efficient routing protocol based on improved artificial bee colony algorithm for wireless sensor networks. *IEEE Access*, 8:133577–133596, 2020.

[2] Tianli Zhou and Chao Tian. Fast erasure coding for data storage: A comprehensive study of the acceleration techniques. In *17th USENIX Conference on File and Storage Technologies (FAST 19)*, pages 317–329, Boston, MA, February 2019. USENIX Association.

[3] Xingxing Xiao and Haining Huang. A clustering routing algorithm based on improved ant colony optimization algorithms for underwater wireless sensor networks. *Algorithms*, 13(10), 2020.

[4] Prachi Maheshwari, Ajay K. Sharma, and Karan Verma. Energy efficient cluster based routing protocol for wsn using butterfly optimization algorithm and ant colony optimization. *Ad Hoc Networks*, 110:102317, 2021.

[5] N. Moussa, E. Nurellari, and A. El Belrhiti El Alaoui. A novel energy-efficient and reliable aco-based routing protocol for wsn-enabled forest fires detection. *Journal of Ambient Intelligence and Humanized Computing*, 14:11639–11655, 2023.

[6] Zunli Kou, Chan Wang, and Ming Lei. Multipath routing with erasure coding in underwater delay tolerant sensor networks. In *2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*, pages 1–5, 2020.

[7] Chunchao Liang, Chang Liu, Ying Li, and Zhengchuan Liang. Erasure coding based efficient communication for internet of things. In *2020 IEEE International Systems Conference (SysCon)*, pages 1–6, 2020.

[8] Qiong Wu, Hai Huang, Xinmiao Lu, Jiaxing Qu, Juntao Gu, and Cunfang Yang. E-reinformif routing algorithm based on energy selection and erasure code tolerance machine. *Electronics*, 12(11), 2023.

[9] Yuya Uezato. Accelerating xor-based erasure coding using program optimization techniques. 2021.

[10] Afsah Sharmin, F. Anwar, and S. M. A. Motakabber. A novel bio-inspired routing algorithm based on aco for wsns. *Bulletin of Electrical Engineering and Informatics*, 8(2):718–726, 2019.

[11] Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Y. Bengio. Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or -1. 02 2016.