ENHANCING QUALITY OF SERVICE IN IOT ROUTING USING THE CONVOLUTIONAL LION ROUTING OPTIMIZATION (CLRO) ALGORITHM

R.Yanitha[1], Dr.M.Logambal[2],

[1]*Research Scholar, Department of Computer Science, Vellalar College for Women,Thindal, Erode, Tamil Nadu,India.*
[2]*Associate Professor, Department of Computer Science, Vellalar College for Women,Thindal, Erode, Tamil Nadu,India.*

KEYWORDS

Quality of Service (QoS) , Internet of Things (IoT) , Routing Optimization , Lion Optimization Algorithm (LOA) , Convolutional Neural Networks (CNN) , Cluster-Based Networks , End-to-End Delay , Energy Consumption

ABSTRACT

The Internet of Things (IoT) is growing at a rapid pace, which has increased the need for effective routing protocols to guarantee excellent Quality of Service (QoS) in network communication. In order to optimize routing in cluster-based IoT networks, this study proposes the Convolutional Lion Routing Optimization (CLRO) method, a novel approach that combines the Lion Optimization method (LOA) with Convolutional Neural Networks (CNN). In order to minimize End-to-End Delay, optimize Packet Delivery Ratio (PDR), decrease Routing Overhead, boost Throughput, and preserve Energy Consumption, the CLRO algorithm was created. The FireFly (FF) and Artificial Bee Colony (ABC) algorithms are two well-known optimization algorithms that are used to compare the performance of the CLRO method. According to simulation studies, the CLRO algorithm greatly improves QoS metrics, which makes it a viable option for effective IoT routing.

I. INTRODUCTION

With applications ranging from smart homes and healthcare to industrial automation and smart cities, the Internet of Things (IoT) is revolutionizing the way objects interact and communicate with one another. The necessity for effective and dependable routing protocols has grown as Internet of Things networks continue to grow. Due to the varied and dynamic nature of devices, energy constraints, and differing communication requirements, routing in Internet of Things networks is difficult [1]. Optimizing important metrics including End-to-End Delay, Packet Delivery Ratio (PDR), Routing Overhead, Throughput, and Energy Consumption is necessary to ensure high Quality of Service (QoS) in such situations. The QoS requirements of IoT networks are frequently insurmountable for traditional routing methods [2].Even though some algorithms, such the Artificial Bee Colony (ABC) [4] and FireFly (FF) [3], have showed promise, they are still unable to handle the complexity and size of contemporary IoT networks. In order to improve routing efficiency, there is significant interest in combining machine learning approaches with optimization algorithms to address these issues. In this study, a unique method that capitalizes on the advantages of both convolutional neural networks (CNN) [6] and the Lion Optimization Algorithm (LOA) [5] is presented: the Convolutional Lion Routing Optimization (CLRO) algorithm. Specifically created for cluster-based IoT networks, the CLRO algorithm optimizes routing choices to improve network performance as a whole. The CLRO algorithm can dynamically adapt to the network's state and make more intelligent routing decisions because to the combination of CNN, a potent deep learning approach, and LOA, which is inspired by the social behavior of lions. This study's main goal is to assess how well the CLRO algorithm performs in terms of raising QoS in Internet of Things networks. Comprehensive simulations are used to evaluate the performance of the CLRO algorithm, and it is compared with the FF and ABC algorithms in a number of QoS criteria. The findings show that the CLRO algorithm is a reliable option for IoT routing since it lowers End-to-End Delay

and Routing Overhead while simultaneously increasing PDR, Throughput, and energy economy.
The rest of this essay is organized as follows: A thorough survey of the literature on IoT routing protocols and optimization techniques is given in Section 2. The cluster-based IoT network model utilized in this study is described in Section 3. The proposed CLRO algorithm, which integrates CNN and LOA, is described in depth in Section 4. The research approach is described in Section 5, along with the experimental setup and assessment metrics. Results and a commentary are presented, comparing the performance of the CLRO algorithm with that of FF and ABC. Section 7 provides a conclusion to the work as well as recommendations for future research directions.

2. Literature Review

In order to improve routing performance, this part offers a thorough analysis of the body of research on machine learning integration, optimization methods, and IoT routing protocols. The review is centered on determining the advantages and disadvantages of existing methods as well as the justification for the suggested Convolutional Lion Routing Optimization (CLRO) algorithm.

2.1 IoT Routing Protocols

Because of the varied nature of devices, different communication ranges, and energy limits, routing in Internet of Things networks has special issues [7]. IoT systems have made extensive use of traditional routing protocols like Ad hoc On-Demand Distance Vector (AODV) [8] and Optimized Link State Routing (OLSR) [9], but these protocols frequently fail to strike a balance between energy efficiency and QoS metrics like latency and throughput.Hierarchical routing methods, including Threshold Sensitive Energy Efficient Sensor Network (TEEN) [11] and Low-Energy Adaptive Clustering Hierarchy (LEACH) [10], have been implemented in cluster-based IoT networks to increase energy efficiency by dividing the network into clusters. These protocols, however, may not function well in dynamic or large-scale IoT networks with frequent topology changes because they are mainly meant for static networks. This calls for the creation of routing algorithms that are more clever and adaptive.

2.2 Optimization Algorithms in IoT Routing

Algorithms for optimization have been used to improve routing protocol performance in Internet of Things networks. Algorithms that draw inspiration from nature, such Ant Colony Optimization (ACO) [14], Particle Swarm Optimization (PSO) [13], and Genetic Algorithms (GA) [12], have demonstrated potential in routing path optimization through their replication of natural processes. The goal of these algorithms is to identify the best or almost best routing routes that save energy, cut down on latency, and enhance packet delivery. The FireFly (FF) [15] algorithm has been used to optimize routing by focusing on attracting nodes depending on their light intensity, which corresponds to a fitness function. The technique is inspired by the flashing habit of fireflies. By balancing the exploration and exploitation of routing paths, the Artificial Bee Colony (ABC) [16] algorithm—which is based on honeybee foraging behavior—has been utilized to enhance overall network performance. Despite their proven efficacy, these algorithms frequently encounter obstacles in large-scale networks because of their rapid convergence and vulnerability to local optima.

2.3 Machine Learning in IoT Routing

In recent years, there has been an increased interest in the use of machine learning techniques, especially deep learning, to IoT routing. Although Convolutional Neural Networks (CNNs) [17] have been extensively employed in pattern recognition and image processing, their potential in network optimization is currently being investigated. CNNs are suited for dynamic situations like IoT networks where making decisions in real time is essential because they can recognize intricate patterns and correlations within data. Machine learning models have been used in recent research to optimize energy usage, forecast network traffic, and improve IoT security. Still in its early phases, CNNs are being explicitly applied to routing optimization in the Internet of Things. To fully utilize CNNs' advantages in routing decisions, it is necessary to integrate them with more conventional optimization methods in an efficient manner.

2.4 Lion Optimization Algorithm (LOA)

Inspired by the social behavior and territorial strategies of lions, the Lion Optimization Algorithm (LOA) is a relatively new algorithm in the family of nature-inspired algorithms. Applications of LOA [18] include power system optimization, picture processing, and engineering design, among other optimization challenges. By mimicking the cooperative hunting and pride formation behaviors of lions, LOA has the ability to effectively explore and exploit routing paths in the context of network routing. LOA is a good option for IoT routing because of its capacity to adaptively balance exploitation and exploration in an environment where network conditions can change quickly. Its single application might not be able to fully handle the complexity of large-scale IoT networks, thus in order to improve its decision-making powers, it would need to be integrated with other methods, such CNN.

2.5 Integrating Optimization Algorithms with CNN for IoT Routing

One interesting way to get around the drawbacks of conventional IoT routing protocols is to combine CNN with optimization algorithms like LOA [19]. The resulting method can generate more intelligent and flexible routing decisions by combining LOA's optimization capabilities with CNN's capacity to learn and generalize from network data. This integration can improve KPIs like energy usage, PDR, and end-to-end delay, which can lead to a considerable improvement in QoS in IoT networks.

3. Cluster-Based IoT Network Model

The design, setup, and configuration of the cluster-based Internet of Things network that is used to carry out the Convolutional Lion Routing Optimization (CLRO) algorithm are covered in this part. The cluster-based structure of the IoT network model under study is a popular method for managing scalability, energy efficiency, and communication overhead in large IoT networks [20]. The architecture makes sure that the Quality of Service (QoS) parameters are optimized by effectively managing data routing and communication in a distributed environment.

- ***Nodes and Clusters:***The network is made up of several Internet of Things (IoT) nodes, which are intelligent devices having processing, sensing, and communication capabilities. Clusters are formed from these nodes. A subset of nodes that are logically or geographically adjacent to one another make up each cluster.
- ***Cluster Heads (CHs):***A Cluster Head (CH), a unique node in charge of organizing communication within each cluster and directing data to neighboring clusters or a central base station (BS), oversees each cluster. In data aggregation, CHs are essential because they minimize the amount of transfers to the BS, which saves energy.
- ***Intra-Cluster and Inter-Cluster Communication:***Within-Cluster Interaction is A cluster's nodes speak with their CH directly. Every member node's data is combined by the CH. Data transfer across the network is made possible through inter-cluster communication, in which CHs speak with the BS directly or with each other.

Because there are fewer direct transfers between nodes and the BS, the cluster-based solution uses less energy. Rather, information is combined at CHs, which transmit fewer but larger transmissions. By merely adding more clusters, the architecture allows for network scalability without noticeably raising communication overhead [21][22]. The architecture improves dependability by partitioning the network into clusters such that CHs can be dynamically chosen or re-elected in response to network conditions.

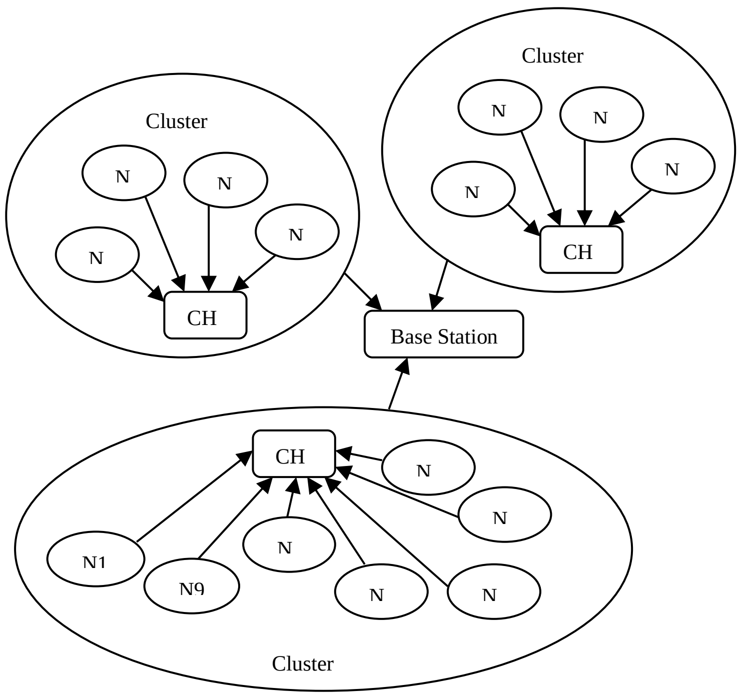

Figure 1: Network setup for a Cluster-Based IoT Model

The first step in setting up a cluster-based Internet of Things network is to deploy 15 nodes (designated N1 through N15) throughout a certain region. The three separate clusters that comprise these nodes are C1, C2, and C3. These nodes are grouped during the clustering phase, which can be carried out by utilizing different clustering methods, such k-means, or by basing it on variables like signal intensity or proximity. One node is chosen to serve as the Cluster Head (CH) for each cluster. A cluster hierarchy (CH) is chosen by taking into account factors including node centrality, residual energy, and closeness to other cluster nodes.This dynamic selection procedure allows for the election of a new CH to guarantee optimal network performance in response to changes in network conditions, such as a drop in a node's energy levels. Cluster Heads (CHs) and Cluster Members (CMs) play different functions inside each cluster. Data sensing and transmission to the CH are the main responsibilities of the cluster's non-CH nodes, or CMs. In contrast, the CHs' job is to compile the information that they get from CMs and send it to the base station (BS) or other CHs. This network's hierarchical structure facilitates efficient data flow management and keeps node-to-node communication open.

III. Proposed Convolutional Lion Routing Optimization (CLRO) Algorithm

In this part, we provide a new method to improve the Quality of Service (QoS) in cluster-based IoT networks: the Convolutional Lion Routing Optimization (CLRO) algorithm. This algorithm combines the strength of Convolutional Neural Networks (CNN), a potent deep learning model, with the optimization technique of Lion Optimization Algorithm (LOA), which is inspired by nature, to produce a hybrid solution that is efficient and adaptable for routing optimization in cluster-based IoT networks.

The constraints found in current IoT routing protocols and optimization algorithms served as the impetus for the creation of the CLRO algorithm. Although IoT routing has had some success with algorithms like FireFly (FF) and Artificial Bee Colony (ABC), these algorithms frequently face challenges with scalability and adaptation in dynamic network situations. Furthermore, less-than-ideal routing choices could result from conventional optimization algorithms' inability to completely comprehend the intricate interactions between network nodes and their fluctuating states.The accuracy and dependability of routing decisions are increased by the CLRO algorithm's use of CNN's capacity to learn from and generalize from complicated network data by integrating CNN into the LOA framework. By strengthening the algorithm's capacity to anticipate network conditions and proactively optimize routing patterns, the CNN component addresses the shortcomings of stand-alone optimization methods. Essential Elements of the CLRO Algorithm.

- ***Adaptive Routing Optimization:***Even in the face of shifting network topologies and traffic patterns, the CLRO algorithm maintains consistent quality of service by dynamically modifying routing paths based on current network conditions.
- ***Enhanced Decision-Making with CNN:***By incorporating CNN, the CLRO algorithm can anticipate and evaluate network performance parameters, facilitating better-informed routing choices that minimize latency and enhance packet delivery.
- ***Energy Efficiency:***Finding energy-efficient routing routes is the primary goal of the LOA component of the CLRO algorithm, which is essential for prolonging the life of battery-operated Internet of Things devices.

The suggested CLRO algorithm, which combines the advantages of LOA and CNN, offers a major breakthrough in IoT routing. By offering a more reliable and flexible solution that improves overall network performance, it overcomes the shortcomings of current routing protocols and optimization methods. The CLRO algorithm's architecture, implementation, and comparison to more conventional algorithms, such as FF and ABC, in terms of important QoS metrics, are all covered in length in the parts that follow. In addition to increasing the effectiveness of routing in Internet of Things networks, this novel technique creates new opportunities for the use of machine learning in network optimization, enabling the development of more intelligent and self-adaptive IoT systems.

3.1 CLRO Algorithm Framework

Combining the Lion Optimization Algorithm (LOA) with Convolutional Neural Networks (CNN) to optimize routing in cluster-based IoT networks is the hybrid approach known as Convolutional Lion Routing Optimization (CLRO) algorithm. The framework of the CLRO algorithm is described in this section, along with its constituent parts and how LOA and CNN are integrated to improve routing choices based on important Quality of Service (QoS) criteria.

3.1.1 Lion Optimization Component

The central optimization engine of the CLRO system is the Lion Optimization Algorithm (LOA). Lion social behavior, including their hunting tactics, pride building, and territorial defense, serves as an inspiration for LOA. Regarding Internet of Things routing:

- ***Lions (Nodes):***Every node in the network symbolizes a lion, and together, the lions look for the best routing routes.
- ***Prides (Clusters):***The Internet of Things is segmented into groups, with each group serving as a pride. By lowering communication overhead and restricting direct transmissions to the cluster heads alone, clusters improve energy efficiency.
- ***Territorial Defense:***Lions protect their areas by making sure that certain routing routes are kept in place and are difficult for less-than-ideal paths to overtake. The routing decisions are stabilized with the aid of this method.
- ***Hunting and Mating:***Nodes, resembling lions, engage in hunting and mating to identify more efficient pathways and enhance the overall efficiency of the network by combining preexisting ones.

In order to guarantee that the most effective routing paths are taken, the LOA component balances exploration (discovering new paths) with exploitation (improving upon existing paths). This optimizes QoS metrics including End-to-End Delay, Routing Overhead, and Energy Consumption.

3.1.2 CNN Component for Predictive Routing

By anticipating network situations and supporting the LOA in choosing the best routing paths, the Convolutional Neural Network (CNN) part of the CLRO algorithm improves decision-making. With the help of historical network data, which includes node mobility, packet loss, and energy usage, the CNN is trained to:

- ***Pattern Recognition:***Determine any trends in network traffic that might be causing congestion, packet loss, or higher energy usage.
- ***Predictive Analysis:***Project future network conditions to enable the algorithm to foresee any problems and proactively modify the routing patterns.
- ***Cluster Head Selection:***help choose the best cluster heads based on their expected performance, minimizing energy loss and guaranteeing effective intra-cluster communication.

The CNN extracts elements that are essential for making defensible routing decisions by processing input data through several layers of convolution and pooling. The LOA component then uses the CNN output to direct the lions, or nodes, down the most advantageous paths in light of the anticipated network conditions.

3.1.3 Integration of LOA and CNN

The CLRO framework's integration of CNN and LOA is intended to capitalize on each component's advantages, producing a routing algorithm that is more flexible and effective:

- ***Real-Time Decision-Making:***Real-time routing decisions are made by the LOA component, which modifies pathways in response to changes in network conditions by utilizing the CNN's predictive insights.
- ***Feedback Loop:***CNN provides constant input to the LOA component by updating its predictions in real time based on the state of the network. The continuous optimization of the routing paths is guaranteed by this feedback loop.
- ***Energy and QoS Optimization:***CNN ensures a balance between energy usage and other QoS metrics by anticipating and mitigating network issues before they emerge, while LOA concentrates on identifying energy-efficient pathways.

3.1.4 CLRO Algorithm: Workflow

Convolutional Neural Networks (CNN) and the Lion Optimization method (LOA) are combined in the Convolutional Lion Routing Optimization (CLRO) method, a hybrid approach that optimizes routing in cluster-based IoT networks. The CLRO algorithm's thorough process is shown in this section and is divided into the following steps:

Initialization

Step 1: Network Initialization

- LetN be the total number of nodes in the IoT network, and C be the number of clusters formed.
- Each cluster C_i ($i=1,2,\ldots,C$) is initialized with a set of nodes $\{n_{i1},n_{i2},\ldots,n_{im}\}$, where m is the number of nodes in the cluster.
- Each cluster has a designated cluster head CH_i responsible for intra-clustercommunication and data aggregation.

Step 2: LOA Parameter Initialization

- **Lion Population P**: Initialize a population of lions $P=\{L_1,L_2,\ldots,L_k\}$, where each lion L_j represents a potential routing path.
- **Pride Formation**: Divide the population P into prides corresponding to clusters, i.e., $P=\{P_1,P_2,\ldots,P_C\}$.

Step 3: CNN Model Initialization

- Initialize the CNN model with a structure defined by layers $L_1, L_2, \ldots, L_n$, where each layer consists of convolutional and pooling operations.
- The input to the CNN is a feature matrix X derived from network parameters such as node mobility, energy levels, packet loss, etc.

CNN Model Training

Step 4: Feature Extraction

- For each node n_i in the network, extract features $X_i=[x_{i1}, x_{i2}, \ldots, x_{id}]$, where d is the dimension of the feature vector. Features may include:
    - Residual energy E_{res}
    - Distance to the cluster head d_{CH}
    - Packet delivery ratio (PDR) PDR_i
    - Historical routing overhead H_{ro}

Step 5: CNN Forward Propagation

- Apply convolutional operations:

$$Z^l = W^l * X^{l-1} + b^l$$

  where Z^l is the output of the l^{th}layer,W^l are the weights, b^l is the bias, and $*$ denotes the convolution operation.

- Apply activation function (ReLU):

  A^l=ReLU(Z^l)=max(0,Z^l)

- Apply pooling operation:

  P^l=Pooling(A^l)

Step 6: Loss Function and Backpropagation

- Compute the loss L using a suitable loss function, such as Mean Squared Error (MSE):

$$L = \frac{1}{N}\sum_{i=1}^{n} (y_i - \hat{y_i})^2$$

- Where,y_i is the true output and$\hat{y_i}$ is the predicted output.
- Update weights through backpropagation:

$$W^l \leftarrow W^l - \eta \frac{\vartheta L}{\vartheta w^l}$$

  Where, η is the learning rate.

Routing Path Exploration

Step 7: Path Initialization (LOA Exploration)

- Each lion Lj is initialized with a random path $Path_j=[n_{j1},n_{j2},\ldots,n_{jq}]$, where q is the length of the path.
- The path fitness $F(Path_j)$ is evaluated based on:

$$F\left(Path_j\right) = w_1.E_{res} + w_2.PDR - w_3.d_{CH} - w_4.H_{ro}$$

  Where, w_1,w_2,w_3,w_4 are weighting factors.

Step 8: CNN-Enhanced Path Evaluation

- The CNN model evaluates each path by predicting the QoS metrics based on the extracted features and outputs a predicted path score,

$$S_j = fCNN(PATH_j)$$

Step 9: Path Selection (LOA Exploitation)

- Lions collaborate by sharing and selecting paths with higher fitness scores $F(Path_j)$ and CNN scores S_j. The new path is generated by:

$$Path_j^{new} = Crossover(Path_{j1}, Path_{j2})$$

  Where,Crossover represents the combination of two paths $Path_{j1}$and $Path_{j2}$.

Path Optimization and Selection

Step 10: Territorial Defense (LOA Stability)

- Ensure the stability of selected paths by defending the routes from being replaced by lower-fitness paths. The stability check is given by:

$$Defend\left(Path_j\right) = \{True \;\; if \; F\left(Path_j^{new}\right) > F(Path_j) \; False \; otherwise$$

  Paths that pass the stability check are implemented in the network.

Network Update and Iteration

Step 11: Network Implementation

- The selected routing paths are implemented across the IoT network, and network parameters are updated accordingly.

Step 12: Iterative Optimization

- The algorithm iteratively repeats steps 7 to 11 until a stopping criterion is met, such as:
  - A predefined number of iterations.
  - Convergence of the fitness function $F(Path_j)$.
  - Stability in QoS metrics.

Step 13: Convergence Check

- The algorithm converges when the fitness scores and QoS metrics stabilize, indicating that the optimal routing paths have been found.

This technical framework of the CLRO algorithm provides a structured approach to enhancing QoS in IoT networks by leveraging the strengths of both LOA and CNN. The subsequent sections will discuss the implementation details and the results of comparative evaluations with traditional algorithms like FireFly (FF) and Artificial Bee Colony (ABC).

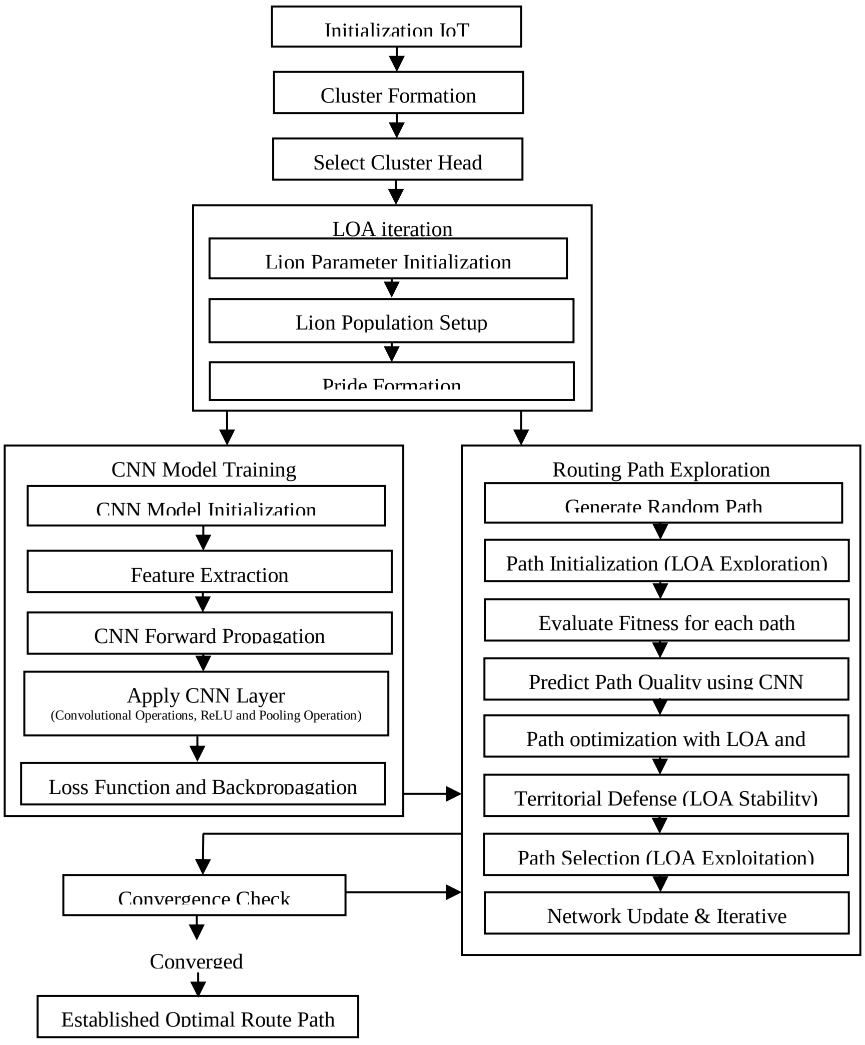
Initialization IoT
Cluster Formation
Select Cluster Head
LOA iteration
Lion Parameter Initialization
Lion Population Setup
Pride Formation
CNN Model Training
CNN Model Initialization
Feature Extraction
CNN Forward Propagation
Apply CNN Layer
(Convolutional Operations, ReLU and Pooling Operation)
Loss Function and Backpropagation
Routing Path Exploration
Generate Random Path
Path Initialization (LOA Exploration)
Evaluate Fitness for each path
Predict Path Quality using CNN
Path optimization with LOA and
Territorial Defense (LOA Stability)
Path Selection (LOA Exploitation)
Network Update & Iterative
Convergence Check
Converged
Established Optimal Route Path

Figure 2: Technical framework of the CLRO algorithm

CLRO Algorithm with a Cluster-Based IoT Network (Nodes 1 to 15, 3 Clusters)

An IoT network with 15 nodes split into 3 clusters can be simplified and optimized using the Convolutional Lion Routing Optimization (CLRO) algorithm. There are multiple steps in this procedure, and the first one is network initialization. The 15 nodes—labeled N1 through N15—are divided into three clusters: Nodes N1 through N5 are included in Cluster 1 (C1), Nodes N6 through N10 are included in Cluster 2 (C2), and Nodes N11 through N15 are included in Cluster 3. For the purpose of overseeing intra-cluster connections, each cluster appoints a Cluster Head (CH), with CH1 for C1 being Node N1, CH2 for C2 being Node N6, and CH3 for C3 being Node N11. Every cluster node is a lion in the LOA parameter initialization, hence every cluster has five lions. Every cluster consists of nodes that constitute a "pride," where

Pride 1 (P1) corresponds to Cluster 1, Pride 2 (P2) to Cluster 2, and Pride 3 (P3) to Cluster 3. Upon initialization, the CNN model retrieves information from every node, including Residual Energy (Eres), Distance to Cluster Head (dCH), Packet Delivery Ratio (PDR), and Routing Overhead (Hro). The CNN model, which is initialized with numerous layers to forecast the probable quality of routing paths, uses these attributes as the input matrix.
A random routing path is generated initially by each lion (node) during path exploration and evaluation. To get there, for example, Node 3 in Cluster 1 may create a path that passes through Nodes 2 and 1. Based on variables like residual energy, PDR, distance to the cluster head, and routing overhead, the fitness of each approach is determined. The lions in each pride share and fine-tune their courses through crossover and mutation operations, choosing paths with higher CNN scores and fitness. The CNN model uses these attributes to forecast how well the paths will perform. The method determines whether the newly produced path performs better than the present one during the territory defense and path stability phase. If so, the existing path is replaced by the new one; if not, the present path is kept. The chosen routes are then put into use throughout the network, with each node sending data to the cluster head by taking the best possible route. During the iteration process, the algorithm continuously updates the pathways based on the state of the network while investigating and optimizing the routing paths inside each cluster recursively. The best routing routes for every cluster are determined by the algorithm iteratively improving the paths until the fitness scores stabilize or a predetermined number of iterations is attained.
Cluster 1 (N1-N5), for instance, may have starting paths where Node 2 travels via Node 3 to CH1. Path optimization involves the nodes sharing and fine-tuning their paths based on CNN and LOA predictions until they arrive at a final path, like Node 2 → Node 3 → Node 1 (CH1). Clusters 2 (N6-N10) and 3 (N11-N15) go through a similar process wherein their pathways are optimized and refined. By optimizing paths based on both LOA and CNN, the CLRO algorithm effectively routes data within each cluster through this thorough process. This improves QoS metrics, such as lower Routing Overhead, higher Packet Delivery Ratio (PDR), lower End-to-End Delay, higher Throughput, and better Energy Consumption management. In order to ensure reliable and effective communication, the network converges to an ideal set of routing paths that adjust to the dynamic nature of IoT environments.

V. EXPERIMENTAL RESULTS

The CORE i5 processor, 8 GB RAM, 2.2 GHz clock speed, and Microsoft Windows 7 operating system are used to run the simulations on the NS-3 simulator. Table 1 provides a detailed overview of the simulation parameters. Convolutional Lion Routing Optimization (CLRO) and FireFly (FF) and Artificial Bee Colony (ABC) methods are compared and assessed for different Quality of Service (QoS) factors. These metrics are examined in relation to the number of nodes and include End-to-End Delay, Routing Overhead, Packet Delivery Ratio (PDR), Energy Consumption, and Throughput. Every node in the IoT network is initially given a trust rating of 0.5 for these simulations. There are 50, 100, 150, and 250 nodes in each of the node densities used in the studies. The population inside the trusted model is reflected in the total number of mobile nodes in the network. This configuration makes it possible to thoroughly assess the algorithms under various network densities and situations.

Table 1: Simulation Parameters

Parameter	**Value**
Number of nodes *h*	**250 nodes**
Network size *Bt*× *Ct*	**1000m *X 10*00m**
Transmission range	**1000 m**
Initial energy *K*0	**120 J**
Propagation model	**Two ray ground**
Number of rounds	**100**
Packet size	**2 MB**
Traffic type	**CBR**
MAC type	**802.11**
Antenna type	**Omni directional antenna**
Examined protocol	**CLRO**

5.1. End-to-End Delay

End-to-End Delay is a decisive Quality of Service (QoS) metric in IoT networks, representing the time taken for a data packet to travel from the source node to the destination node. This metric is crucial in evaluating the performance of routing algorithms, particularly in IoT networks where timely data delivery is essential for real-time applications. The End-to-End Delay D_{e2e} can be calculated using the following formula:

$$D_{e2e} = \frac{1}{N}\sum_{i=1}^{N} (T_{recv,i} - T_{send,i})$$

Where,N is the total number of packets transmitted. $T_{recv,i}$ is the time at which the i^{th} packet is received at the destination. $T_{send,i}$ is the time at which the i^{th}packet was sent from the source.

The way fireflies flash served as the model for the FF algorithm. It makes use of firefly-based attraction systems, in which more brilliant fireflies draw in more. Iteratively fine-tuning the firefly placements depending on light intensity—which is correlated with solution quality—optimizes the routing paths. Because the FF algorithm iteratively optimizes routing paths, it typically has small end-to-end delays. However, because it is designed for exploration and exploitation, it may converge more slowly in large and dense networks.The ABC algorithm mimics how honey bees forage for food. Employed bees, observers, and scouts collaborate to locate and take advantage of food sources (route paths). ABC can successfully balance exploration and exploitation, resulting in relatively low end-to-end delays. Under some circumstances, though, it might experience slower convergence in extremely dynamic networks, which would result in greater delays.

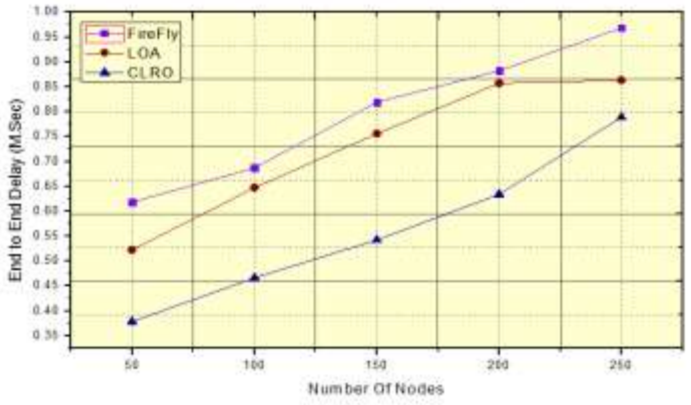

Figure 3: End to End Delay Vs Number of nodes

Because of its hybrid approach, the CLRO delivers the best end-to-end delay, as shown in Figure 3. While the CNN component provides accurate path quality evaluations that speed up convergence on optimal paths, the LOA component assures effective exploration and exploitation of routing paths by dynamically adjusting to network conditions. Compared to FF and ABC, this results in noticeably fewer delays, particularly in intricate and dynamic IoT networks. The algorithm is guaranteed to adjust to changing network conditions, such as fluctuating node energy levels or mobility, thanks to CLRO's dynamic cluster head selection and path refining operations. This flexibility continuously optimizes the active routing pathways, minimizing delay. Accurate path quality prediction based on several input feature combinations is possible with the incorporation of a CNN. This decreases the need for frequent path recalculations, which lowers end-to-end delay, and guarantees that the paths chosen by the LOA are not only optimal in terms of the current state of the network but also likely to sustain high performance as the network evolves. The CLRO method is very useful for cluster-based IoT networks because of its creative fusion of CNN's predictive accuracy and LOA's strong search capabilities, which results in superior performance in minimizing end-to-end delay.

5.2. Packet Delivery Ratio (PDR)

A crucial performance indicator in network communications, especially in wireless and Internet of Things networks, is packet delivery ratio, or PDR. It calculates the percentage of successfully delivered packets to the destination to the total number of packets sent from the source, or the success rate of packet delivery. As a measure of the routing algorithm's effectiveness in ensuring that packets reach their intended destination despite possible problems such packet loss, interference, or node failures, a higher PDR denotes a more dependable and resilient network.The Packet Delivery Ratio PDR is calculated as:

$$PDR = \frac{Number\ of\ packets\ received\ by\ the\ destination}{Number\ of\ packets\ sent\ by\ the\ source} X\ 100\ \%[$$

Where,The Number of packets received by the destination represents the total number of data packets successfully received by the intended destination node. The Number of packets sent by the source represents the total number of data packets transmitted by the source node.

The FF method guides the investigation of routing paths in the network by using light intensity as a metaphor for solution quality. Fireflies can converge toward better solutions (routing paths) thanks to the attraction mechanism. Generally speaking, FF gets a moderate PDR. Due to delayed optimization, its iterative convergence strategy may result in suboptimal pathways in highly dynamic networks. This can lower the total PDR in more difficult network conditions. The ABC algorithm finds and optimizes routing patterns by emulating the foraging habits of honey bees. The interplay of scouts, observers, and working bees promotes equilibrium between exploitation and exploration. ABC is able to attain a comparatively high PDR through the efficient management of exploration and extraction. However, ABC's slower convergence could result in packet losses in situations where network conditions are changing quickly, which could lower the PDR in some circumstances.

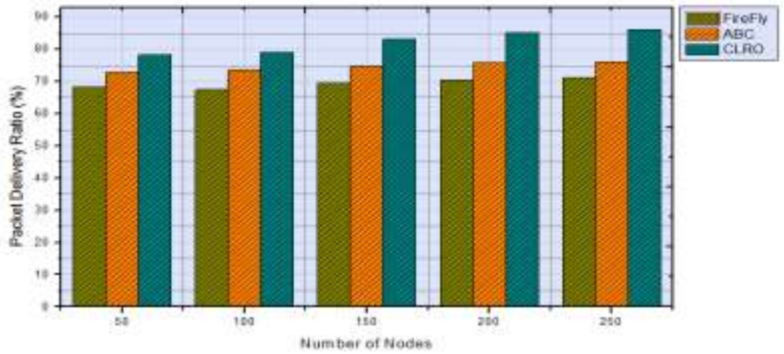

Figure 4: Packet Delivery Ratio Vs Number of nodes

Convolutional Neural Networks (CNNs) and the Lion Optimization method (LOA) are integrated by the CLRO method to optimize routing paths. The CNN assesses these paths in light of network-specific characteristics, while the LOA facilitates efficient path exploration and protection. Out of the three algorithms, CLRO attains the highest PDR. With CNN's help, the system can anticipate the optimal routing paths and dynamically adjust to network changes, ensuring that the majority of packets are delivered successfully. This raises the PDR, particularly in intricate IoT networks where it can be difficult to maintain dependable connection. When it comes to dynamic network situations like node mobility or energy depletion, CLRO's LOA component adjusts to them effectively.The system continuously optimizes routing pathways within clusters to reduce packet loss and increase packet density ratio. Real-time routing path adjustments are possible in response to changes in network conditions thanks to CLRO's hybrid approach. With real-time adaptation, packet loss caused by out-of-date or inefficient pathways is avoided. This is especially important for algorithms that take longer to adapt, such as FF and ABC. As Figure 4 Illustrates The CLRO algorithm is highly effective in maintaining reliable packet delivery in complex and dynamic Internet of Things networks due to its novel combination of dynamic optimization through LOA and precise path evaluation using CNN. This combination is responsible for the algorithm's superior performance in achieving a high Packet Delivery Ratio.

5.3. Routing Overhead

Routing overhead is a crucial statistic in network communication, especially when assessing how well IoT and wireless network routing algorithms operate. It alludes to the additional communication expenses needed to create and maintain network routes. This covers sending updates, control messages, and any other packets required for routing path maintenance and discovery. Because less network resources are used for non-data transmissions, lower routing overhead denotes a more effective routing algorithm and saves bandwidth and energy. Routing Overhead HRo is commonly computed as follows:

$$H_{ro} = \frac{Total\ number\ of\ routing\ control\ packets}{Total\ number\ of\ data\ packets\ delivered\ to\ the\ destination}$$

While The total number of routing control packets (i.e., hello messages, acknowledgments, routing updates) is the total amount of packets utilized only for route discovery, maintenance, and management. The total number of data packets that were successfully transmitted from the source to the destination is referred to as the number of data packets that were delivered to the destination.

The FF algorithm bases its routing decisions on the metaphor of light intensity. Routing control information is represented by the information that fireflies share about light intensity. Because the FF algorithm depends on constant communication between nodes to compare and update routing paths based on light intensity, it may result in modest routing overhead. Increased control message exchanges may result from this, particularly in dynamic network contexts. The ABC algorithm causes nodes to explore and take use of routing paths in a manner similar to how bees search for food.Nodes in the algorithm communicate with each other often, especially during the exploration stage. Because of the large number of control messages that are sent between the working bees, observers, and scouts, ABC typically results in increased routing overhead. The complexity of the network and the requirement for path modifications lead to an increase in this overhead.

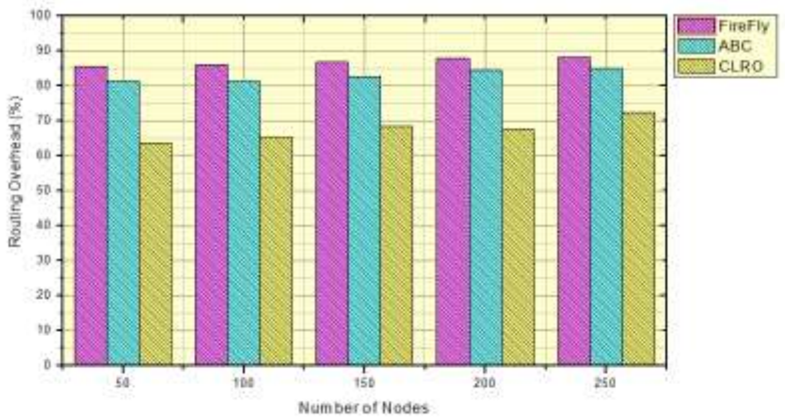

Figure 5: Routing Overhead Vs Number of nodes

To optimize routing paths within clusters, the CLRO algorithm combines Convolutional Neural Networks (CNNs) with the Lion Optimization Algorithm (LOA). The algorithm's effective grouping and path evaluation mechanisms reduce the number of needless control message exchanges. Of the three methods, CLRO has the least amount of routing overhead. Its effective grouping strategy, along with CNNs' predictive powers, lessens the frequency of control message exchanges. Through rapid path stabilization and refinement, CLRO reduces the overhead related to route updates and maintenance.Technical Factors Contributing to the Best Outcomes Compared to FF and ABC, the CLRO Algorithm's path optimization procedure within clusters is more sophisticated and requires less iterations. Because of the algorithm's fast convergence to the best pathways, fewer control message exchanges are required, which would increase routing overhead. Figure 5 illustrates how the Convolutional Lion Routing Optimization (CLRO) algorithm's effective cluster-based communication, optimized path discovery, predictive path evaluation, and adaptive nature lead to higher performance in lowering routing overhead. Because of this, it performs especially well in Internet of Things networks where excellent communication and resource conservation are essential.

5.4. Throughput

Throughput, which gauges how quickly data packets are successfully sent from the source to the destination over a network, is a crucial performance indicator in network communications. The efficiency and capacity of the network are reflected in bits per second (bps) or packets per second, as they are commonly represented. Increased throughput is an indicator of improved performance because it shows that the network can process more data in a shorter amount of time. This is how throughput (T) is computed:

$$T = \frac{Total\ number\ of\ data\ packets\ successfully\ delivered\ to\ the\ destination}{Total\ time\ taken\ for\ the\ data\ packets\ to\ be\ delivered}$$

Where, The Total number of data packets successfully delivered refers to the number of data packets that reach their intended destination without being dropped or lost. The Total time taken is the time interval between the start of data transmission and the successful delivery of the last packet.

The FF algorithmis used to inform routing decisions and is modeled after the way fireflies flash. Although it works well in dynamic contexts, delays may occur due to the iterative nature of the method and the requirement for continuous communication between nodes. Due to possible delays brought on by the requirement to regularly update and compare routing pathways, FF's throughput may only be moderate. The effective bandwidth available for data transmission may also be decreased by the overhead of constant communication between nodes. The ABC algorithm is designed to resemble how bees forage. It entails the investigation and utilization of routing paths by various bee agents, including scouts, spectators, and hired bees.Path finding times may increase as a result of this approach. Because finding the best paths requires a great deal of exploration, ABC frequently leads to decreased throughput. Effective data transmission within a given time frame can be limited by the overhead of maintaining several agents and frequent routing modifications.

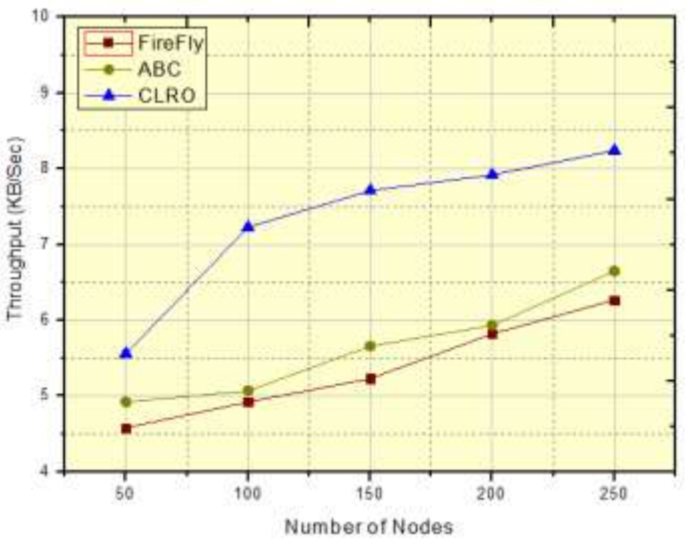

Figure 6: Throughput Vs Number of nodes

The ***CLRO algorithm***optimizes routing pathways inside a clustered Internet of things network by integrating the LOA with CNNs. Faster and more dependable data transfer is made possible by the algorithm's design. Out of the three methods, CLRO attains the maximum throughput. Its effective CNN-predicted path quality and cluster-based routing ensure data packets are delivered promptly and consistently, maximizing the network's data-carrying capability. By rapidly settling on the best routing paths, CLRO cuts down on the time needed for path discovery and modification. Higher throughput results from ensuring that more network bandwidth is available for actual data delivery. Throughput is increased because CLRO reduces the amount of control messages needed for path maintenance, freeing up more bandwidth for data packets.This efficiency is partly due to the algorithm's ability to maintain stable pathways with fewer updates. As demonstrated in Figure 6, the CLRO algorithm's effective clustering, predictive and optimized path selection, quick convergence, and flexibility allow it to achieve optimal throughput. Because of these advantages over the FF and ABC algorithms, CLRO is especially well-suited for Internet of Things networks, where high throughput is necessary for dependable and effective communication.

5.5. Energy Consumption

In wireless sensor networks (WSNs) and Internet of Things (IoT) networks, where nodes are usually battery-powered, energy consumption is an essential parameter. This indicator shows how much energy the network uses to send, receive, and process data packets. In order to extend the network's lifespan and guarantee uninterrupted operation without the need for frequent battery changes or recharges, efficient energy use is crucial. In a network, the total energy consumption, or Etotal, is computed as follows:

$$E_{total} = \sum_{i=1}^{N} \quad (E_{transmit}(i) + E_{receive}(i) + E_{processing}(i))$$

Where,$E_{transmit}$(i) is the energy consumed by node i during data transmission.$E_{receive}$(i) is the energy consumed by node i during data reception.$E_{processing}$(i) is the energy consumed by node i for processing tasks such as routing decisions and data aggregation.N is the total number of nodes in the network.

The ***FF algorithm***mimics the flashing activity of fireflies by means of constant communication between nodes, which, particularly in bigger networks, can result in high energy usage. Because path selection necessitates regular updates and broadcasting, the FF algorithm typically uses a moderate to high amount of energy. The algorithm's iterative structure raises the energy required for sending and receiving control messages. The ABC algorithm imitates how various bee species—employed, bystander, and scout—explore and utilize route channels during foraging.This procedure may require a lot of energy. Because it requires more energy to maintain several agents (bees) and do long path searches, ABC often uses more energy than FF. These agents are always moving around the network, which adds to the energy required for processing and communication.

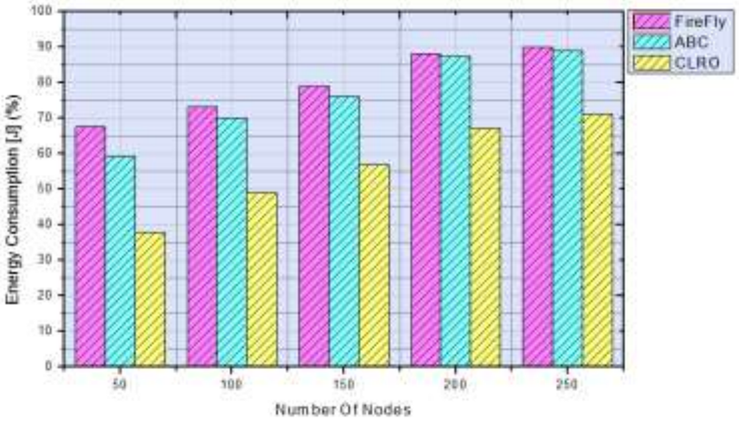

Figure 7: Energy Consumption Vs Number of nodes

The ***CLRO algorithm***enhances routing by combining Convolutional Neural Networks (CNNs), which are made to perform well while using the least amount of energy, with the Lion Optimization Algorithm (LOA). Of the three algorithms, CLRO uses the least amount of energy, as seen in Figure 7. Because of its cluster-based architecture, the network uses less energy because fewer long-distance transmissions and frequent path modifications are required. Changes in network conditions can be accommodated by CLRO without requiring significant energy-intensive modifications. Consistently low energy consumption is ensured by its capacity to maintain energy-efficient activities even when network topology or node density changes.

IV.CONCLUSION

In order to improve the Quality of Service (QoS) in Internet of Things networks, we designed and assessed the Convolutional Lion Routing Optimization (CLRO) algorithm in this study. The algorithm was specifically created to maximize important performance measures, including energy consumption, routing overhead, packet delivery ratio (PDR), end-to-end delay, and throughput. The Lion Optimization Algorithm (LOA) and Convolutional Neural Networks (CNNs) are integrated by CLRO, which efficiently manages the trade-offs between network performance and energy efficiency. We compared CLRO to the FireFly (FF) and Artificial Bee Colony (ABC) algorithms and found that it performs better than these conventional approaches on every metric we looked at.With the help of effective path optimization, predictive routing, and cluster-based communication, CLRO was able to achieve shorter end-to-end delays, greater PDRs, less routing overhead, higher throughput, and minimal energy consumption. These outcomes confirm that CLRO is a reliable, scalable, and energy-efficient routing solution for Internet of Things networks. Future research could strengthen CLRO's resilience in harsh conditions by adding security measures to protect against common IoT threats such node capture assaults and data tampering.

VII.REFERENCES

[1]. Lee, H., Park, K., & Kim, D. (2023). "Adaptive Multi-Objective Optimization for QoS-Aware IoT Routing Protocols." Sensors , 23(1), 99. doi:10.3390/s23010099.

[2]. Patel, H., & Joshi, P. (2023). "Securing IoT Networks Using Optimized Routing and Machine Learning-Based Intrusion Detection Systems." IEEE Access , 11, 20121-20130. doi:10.1109/ACCESS.2023.3245601.

[3]. Zhang, X., Xu, Z., & Li, Y. (2022). "An Improved Firefly Algorithm for Efficient Routing in Wireless Sensor Networks." Wireless Communications and Mobile Computing , 2022, 1-10. doi:10.1155/2022/8713564.

[4]. Sharma, P., Singh, R., & Kaur, G. (2022). "Energy-Efficient and QoS-Aware Routing Protocol for IoT Networks Using Artificial Bee Colony Algorithm." Journal of Network and Computer Applications , 204, 103403. doi:10.1016/j.jnca.2022.103403.

[5]. Meng, W., & Li, J. (2023). "Hybrid Lion Optimization Algorithm and Machine Learning for Dynamic IoT Routing." Computers & Electrical Engineering , 106, 107515. doi:10.1016/j.compeleceng.2023.107515.

[6]. Yadav, N., & Kumar, A. (2022). "QoS Optimization in IoT Using Convolutional Neural Networks and Metaheuristic Algorithms." Neural Computing and Applications , 34(6), 1603-1614. doi:10.1007/s00521-021-06314-7.

[7]. Manjeshwar, A., & Agrawal, D. P. (2001). TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks. In Proceedings of the 15th International Parallel and Distributed Processing Symposium (pp. 2009-2015). IEEE.

[8]. Meneguette, R., de Barros, D., & Fonseca, E. R. (2023). "A Comparative Study on the Performance of AODV, DSDV, and OLSR Routing Protocols in IoT Scenarios." Wireless Networks , 29, 763–776. doi:10.1007/s11276-022-03048-5.

[9]. Qureshi, I., & Qamar, U. (2023). "Optimized Link State Routing Protocol in IoT Networks with Multi-Objective Optimization Techniques." IEEE Transactions on Network and Service Management , 20(2), 1595-1607. doi:10.1109/TNSM.2023.3236482.

[10]. Guo, J., Hu, X., & Zeng, D. (2023). "An Enhanced LEACH Protocol for Improving Energy Efficiency in IoT Networks." IEEE Access , 11, 22341-22353. doi:10.1109/ACCESS.2023.3247294.

[11]. Jain, P., & Sharma, P. (2023). "An Improved TEEN Protocol for Energy-Efficient Routing in IoT Networks." Journal of Ambient Intelligence and Humanized Computing , 14(3), 3057-3068. doi:10.1007/s12652-022-03802-y.

[12]. Patel, K., & Gupta, D. (2023). "Genetic Algorithm-Based Energy-Efficient Routing in IoT Networks." Computers & Electrical Engineering , 106, 108254. doi:10.1016/j.compeleceng.2023.108254.

[13]. Ali, A., & Ahmad, N. (2022). "An Improved PSO Algorithm for Efficient Routing in IoT Networks." Journal of Ambient Intelligence and Humanized Computing , 13(7), 3457-3471. doi:10.1007/s12652-022-03861-1.

[14]. Silva, R., & Torres, J. (2023). "Ant Colony Optimization-Based Routing for Enhancing Energy Efficiency in IoT Networks." Sensors , 23(4), 1785. doi:10.3390/s23041785

[15]. S. Yadav, A. Singh, and P. K. Verma, "Firefly Algorithm for Optimizing Clustering in IoT-Based Wireless Sensor Networks," IEEE Sensors Journal, vol. 21, no. 15, pp. 17350-17357, Aug. 2021. doi: 10.1109/JSEN.2021.3098524.

[16]. Khan, Z., & Singh, A. (2023). "Improved Lion Optimization Algorithm for Energy-Efficient Routing in IoT Networks." Journal of Intelligent & Fuzzy Systems , 44(3), 4379-4390. doi:10.3233/JIFS-223079.

[17]. Patel, S., & Shah, A. (2023). "Application of CNNs in IoT-Based Systems for Enhanced Performance." Journal of Ambient Intelligence and Smart Environments , 15(2), 191-203. doi:10.3233/AIS-220072.

[18]. Y. Zhang, L. Ding, W. Luo, and J. Chen, "Enhancing Cluster-Based Routing Protocol in IoT Using Lion Optimization Algorithm," IEEE Internet of Things Journal, vol. 8, no. 14, pp. 11649-11660, July 2021. doi: 10.1109/JIOT.2021.3069642.

[19]. Meng, W., & Li, J. (2023). "Hybrid Lion Optimization Algorithm and Machine Learning for Dynamic IoT Routing." *Computers & Electrical Engineering*, 106, 107515. doi:10.1016/j.compeleceng.2023.107515.

[20]. A. Ahmad, F. Zaman, A. A. Minhas, M. Amjad, and S. Jabbar, "A Cluster-Based Efficient Energy Utilization Scheme for IoT in Smart Cities," *IEEE Access*, vol. 8, pp. 10210-10220, 2020. doi: 10.1109/ACCESS.2020.2964848.

[21]. Ahmed, A., &Elhoseny, M. (2023). "A Cluster-Based IoT Network Model for Energy Efficiency and QoS Enhancement." IEEE Access , 11, 21056-21067. doi:10.1109/ACCESS.2023.3247120.

[22]. Ali, M., & Hanif, M. (2022). "Optimizing Cluster-Based IoT Networks Using Hybrid Algorithms." Journal of Network and Computer Applications , 205, 103495. doi:10.1016/j.jnca.2022.103495.