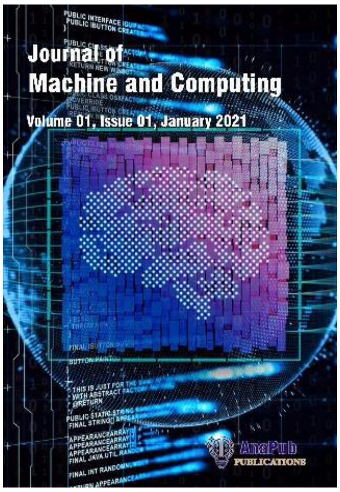
Journal Pre-proof



Energy-Efficient Reinforcement Learning-Based Adaptive Resource Allocation for LoRa Networks

Suchitra N Shenoy, Ganesh V Bhat and Manoj T Gadiyar H

DOI: 10.53759/7669/jmc202505186

Reference: JMC202505186

Journal: Journal of Machine and Computing.

Received 18 June 2025

Revised from 28 July 2025

Accepted 03 August 2025

AnaPub
PUBLICATIONS

Energy-Efficient Reinforcement Learning-Based Adaptive Resource Allocation for LoRa Networks

[1]Suchitra N Shenoy, [2]Ganesh V Bhat and [3]H. Manoj T. Gadiyar

[1]Research Scholar, Department of Electronics and Communication Engineering,
Canara Engineering College, Mangalore – 574219, Visvesvaraya Technological University, Belagavi.
[2]Research Supervisor, Department of Electronics and Communication Engineering,
Canara Engineering College, Mangalore – 574219, Visvesvaraya Technological University, Belagavi.
[3]Research Supervisor, Department of Information Science and Engineering,
Canara Engineering College, Mangalore – 574219, Visvesvaraya Technological University, Belagavi.

Correspondence should be addressed to Suchitra N Shenoy: suchiprab_16@yahoo.co.in

Abstract - In the last few years, a surge in IoT applications has ramped up the need for effective and dependable data transmission in LoRa-based systems. Yet, traditional resource allocation methods in LoRa systems face major drawbacks such as higher packet loss, interference, excessive energy use, limited coverage, slow transmission speeds, and increased operational expenses. To tackle these issues, this study introduces a new hybrid optimisation framework that combines Hybrid Reinforcement Learning, named as Double Deep Q-Learning based Actor-Critic mechanism (Hy-DeoQ-AC), with a hybrid Levy Flight Assisted Rabbit optimisation algorithm (Hy-LevRBO). Hy-DeoQ-AC mechanism learns optimal network configurations dynamically by engaging with the environment, concentrating on key transmission parameters like spreading factor, transmission power, and channel selection to satisfy strict Quality of Service (QoS) requirements of IoT devices. Additionally, the hybrid optimisation gains from Hy-LevRBO, which fine-tunes chosen parameters and boosts capability to evade local optima. Thus, this combined strategy greatly enhances energy efficiency, maximises throughput, extends transmission range, and reduces latency in LoRa networks. The Comprehensive experimental analysis attains a throughput of 56.8471(bits/s), energy efficiency of 16.1364 (bits/J), which confirms the proposed model's superiority and achieves better performance across various metrics. This research offers an energy-efficient solution for IoT communications.

Keywords - Q learning, long-range network, Levy Flight, Rabbit optimisation, Actor critic approach.

I. INTRODUCTION

The rise of Internet of Things (IoT) assurances to combine 22 billion devices, and long-range (LoRa) will successfully manage it in 2025 [1]. LoRa is efficiently used in cellular data networks and areas such as manufacturing and academia to offer improved communication [2]. LoRa appliance offers lower development cost and power consumption. The chirp spread spectrum approach uses various spreading factors (SF) in LoRa with lower energy consumption [3]. It helps enhance the efficiency of the network. In LoRa technology, the network performance and battery life of LoRa network devices are improved by using physical layer operations [4]. In IoT, customisation and resource allocations are critical parts. Some example limitations, like fewer shared resources, inaccurate radio influences, and limited intrinsic networks, disturb resource allocation [5].

These causes enhance heterogeneity and quality of service (QoS) about hardware diversity. Concurrence problems have increased in the development of LoRa [6]. By considering signal-to-noise ratio (SNR), the server improves communication power and modifies SF for enhancing energy efficiency, airtime, and data transmission speed [7]. To increase the efficiency of the resource, the transmit power is changed at each stage [8]. A wide range of IoT technologies requires several connected devices for data transmission based on resource allocation [9]. To increase resource efficiency, a convincing resource allocation scheme is needed for channel conflict avoidance, and an intelligent resource allocation framework is essential [10].

The existing frameworks for resource allocation in LoRa suffer from high channel usage, computational cost, and decreased network capacity while allocating the resources to large networks [11]. The previous resource allocation algorithms offer less QoS. It reduces the network's robustness. Some lightweight techniques for SFs produce inaccurate reliability [12]. The goal of the traditional system is to statistically minimise the probabilities of two or more communications overlapping in frequency and time [13]. Existing LoRa networks must increase reliability and control overhead since they are still implemented based on network size. Additionally, the main goal of a scheduling strategy is to enhance reliability by allocating transmission slots with minimal cost [14]. Yet, traditional networks enhance overhead and computational cost [15]. The interactions with educational settings, such as user volume, colour, and service quality requirements, are issues of existing approaches [15].

Motivation

The Internet of Things (IoT) will connect 30+ billion devices by 2030 with long-range (LoRa) technology as the primary management system. LoRa is widely utilised in cellular networks, industry, and academia to improve communication. Wireless sensor networks (WSN) adopting low-power wide area networks (LPWAN), such as long-range (LoRa) WAN, help to improve communication standards. LoRa has been used to gather sensor data for many applications, such as environmental monitoring. Owing to the presence of interference and congestion with the development of IoT devices, the existing LoRa system is also impacted by these issues. The conventional resource allocation-based data transmission in LoRa has many challenges, including security, server dependence, network connectivity, coverage, and limited resource capabilities. Existing systems mainly suffered from high computational load and communication latency issues. To overcome these issues, a novel optimal hybrid reinforcement learning-assisted resource allocation in LoRa is proposed. The main contributions of the work are:

- To review existing studies and analyse existing models and gaps.
- To develop an efficient reinforcement learning-based algorithm for optimal resource allocation in LoRa systems for energy efficiency and performance optimisation.
- To introduce a Double Deep Q-Learning-based actor critic mechanism with Levy-assisted bio-inspired optimisation for creating optimal resource allocation policies.
- To evaluate the proposed model's effectiveness through extensive simulation, we analyse key performance metrics such as energy consumption, latency, throughput, signal-to-interference-plus-noise ratio (SINR), and transmission rate.

The structure of the manuscript is organised as follows: Section 2 reviews the existing approaches and techniques, Section 3 provides a thorough explanation of the proposed model, Section 4 examines and deliberates outcomes obtained from the proposed approach, and Section 5 concludes with a summary of findings and future scope of this research.

II. RELATED WORKS

Azizi et al. [16] explored a reinforcement learning based resource allocation model to adjust their transmission parameter. The optimisation algorithm includes two stages, namely exploitation and exploration. These phases help to allocate the resources in LoRa. From the simulation results, it is evident that the suggested framework provides better results compared to traditional algorithms in terms of packet delivery ratio and convergence speed. Limitations of this study are high computational cost, scalability, and adaptability challenges. Rao and Sunder [17] minimised power transmission and increased data transfer rate in LoRa networks by using a reinforcement learning approach-based system. The approach was used for finding variables during the transmission of data. An effective hybrid coati with an energy valley strategy tuned these parameters. The tuned parameter was applied to the terminal hubs. Parameter optimisation was used to enhance the throughput and reduce the energy usage. However, the challenge of the suggested system includes supporting only low data rates and complex optimisation challenges. Minhaj et al. [18] implemented a new way of allocating the SF and transmission power to the devices by joining a decentralised and centralised technique with two independent learning procedures. Transmission power was allocated centrally by decreasing the contextual bandit problem using machine learning (ML) techniques. The reinforcement learning (RL) technique assigned the SF parameter to the network devices. The designed system provided higher accuracy and lower energy usage for large, congested networks than current state-of-the-art algorithms. High packet loss ratio is the main issue of this system. Gava et al. [19] provided a novel resource optimisation scheme in LoRa for maintaining costs and implementation complexity. Performance investigations were carried out in LoRa using LoRa repeaters to improve the coverage. Total execution time and energy usage were minimised by adjusting parameters like transmission power, spreading factor, and bandwidth. However, maintaining a balance between energy efficiency and data collecting takes time and effort. Mahesh et al. [20-23] suggested a co-optimal Q-reinforcement learning (CO-QL) model as a resource allocation mechanism. Here, Q-reinforcement learning was utilised to learn the information about nodes, and COA helps to choose the optimal action for enhancing the reward. This approach helps to improve packet delivery rate performance. The performances, such as packet success ratio (PSR), packet collision rate (PCR), time, delay, and energy, were evaluated and compared with recent research models. High latency was one of the major limitations in this model. Table 1 defines the performance evaluation of the proposed and existing models [24-25].

Table 1. Performance analysis of proposed and existing approaches

Author	Techniques	Merits	Demerits
Azizi et al. [16]	Reinforcement learning	Improves convergence speed	Scalability issues
Rao and Sunder [17]	Coati algorithm	Less power transmission	Complex optimisation challenge
Minhaj et al. [18]	Machine learning + Reinforcement learning	Low energy usage	Takes more time for execution
Gava et al. [19]	Resource optimisation approach	Reduced execution time	Balancing energy efficiency is challenging
Mahesh et al. [20]	Q-reinforcement learning	Improves energy efficiency	High delay

Research Gap

Existing research faces various limitations, which are discussed as follows: Azizi et al. concentrated on allocating resources with an optimisation algorithm that had phases for exploration and exploitation, but they encountered difficulties with scalability and high processing costs. Some other existing models have enhanced data transfer rates and minimised power consumption using optimisation strategies but struggled with low data support and complex optimisation issues. Minhaj et al. combined centralised and decentralised learning for SF and power allocation, achieving high accuracy but suffering from a high packet loss ratio [26]. Gava et al. introduced a novel optimisation scheme using LoRa repeaters, yet balancing energy efficiency and data collection remained challenging. Mahesh et al. proposed a CO-QL model that improved packet delivery but suffered from high latency. Despite advancements, key research gaps include addressing computational complexity, scalability, adaptability, optimisation trade-offs, and ensuring reliable transmission in large-scale and dynamic LoRa networks.

III. PROPOSED METHODOLOGY

The proposed approach utilises the LoRa system to minimise transmission power, which is effectively identified by Hybrid reinforcement learning, namely, Double deep Q-Learning based actor critic mechanism (Hy-DeoQ-AC). Here, the Hy-DeoQ-AC mechanism is used to solve LoRa challenges, and it aims to optimise the allocation of network resources like transmission power, spreading factor, and channel. This approach helps to allocate the transmission power, spreading factor, and channel for IoT devices to enhance quality of service requirements. The appropriate parameters are selected from the Hy-DeoQ-AC model, which is properly tuned by the Hybrid Levy flight assisted rabbit optimisation algorithm (Hy-LevRBO). This approach helps to tune the network resources of the model. The tuning process helps to increase the energy efficiency, throughput, and transmission range and reduce latency [27-28]. The server in the LoRa is matched by the agents generated by the Hy-DeoQ-AC model. Then, transmission parameters are given to the network terminal hub after the agents in the Hy-DeoQ-AC are generated. Throughput, energy efficiency, latency, and transmission rate are analysed using this optimisation strategy. The proposed model optimises resource allocation in LoRa-based Non-Orthogonal Multiple Access (NOMA) networks by dynamically adjusting transmission power, spreading factor, and channel selection to enhance QoS. RL operates by training agents to interact with the environment, where the Hy-DeoQ-AC mechanism leverages a double deep Q learning based actor critic framework to balance the exploration and exploitation for optimal decision making. Hy-LevRBO mechanism fine-tunes the network parameters, improving energy efficiency, throughput, and transmission range while reducing latency. The trained RL agents match the server in LoRa, optimising transmission parameters for terminal hubs.

System Model

A single LoRa [21] model consists of a duplex gateway and fixed LoRa end devices. The LoRa system is divided into three classes D E F. The network model of LoRa is defined in Fig. 1.

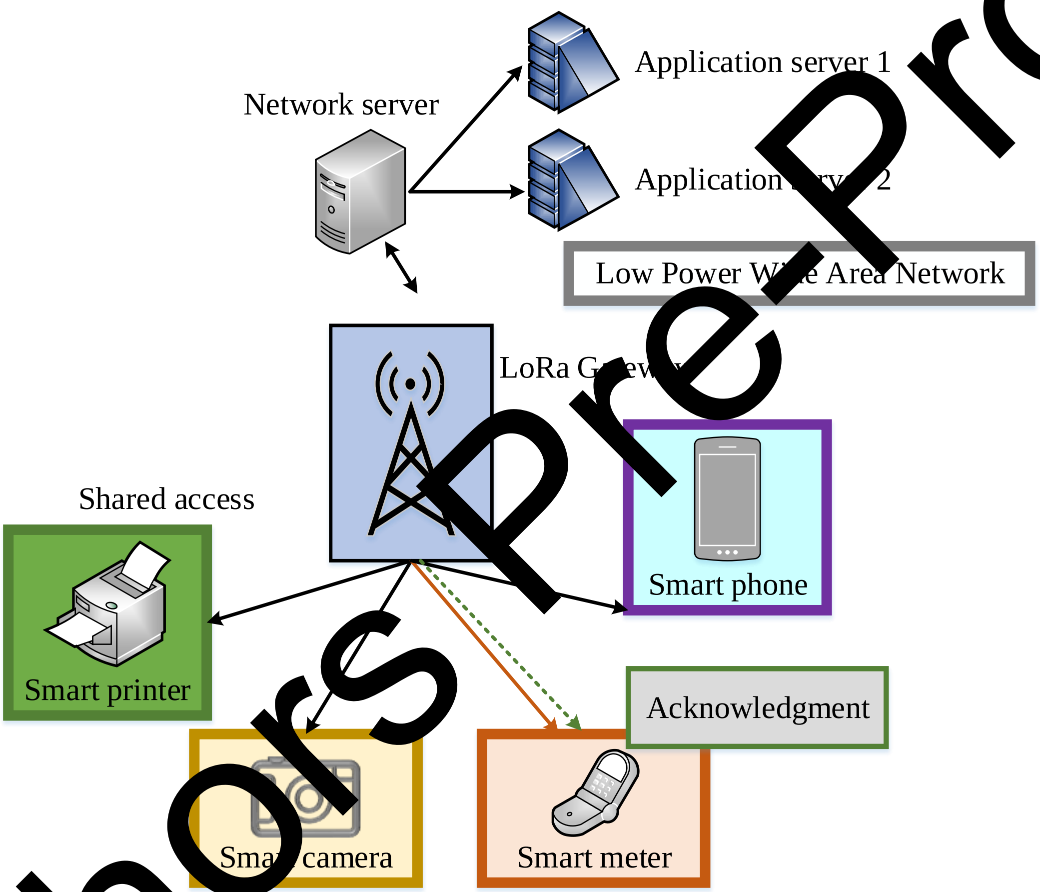


Fig 1. LoRa network model

The end devices are evenly distributed around the gateway and are classified as class D devices. Most of the time, these end devices remain in sleep mode to conserve battery life. They only wake up to perform uplink transmissions when a new packet is received. Fig. 2 defines the architecture of resource allocation in LoRa.

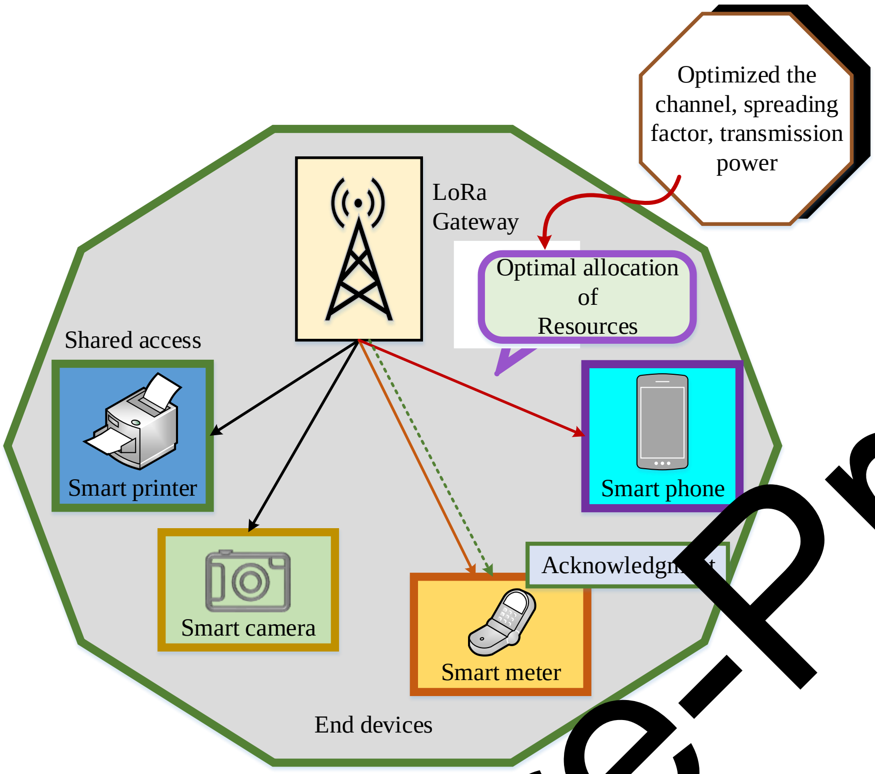


Fig 2. Architecture of Resource Allo

Each end device carries out an uplink transmission during the system training phase, and each one receives downlink acknowledgement from the gateway. They assume that the gateway sends acknowledgements separately from the uplink channel to avoid interference between downlink acknowledgements and uplink transmissions. The LoRa system specifies two receive windows $S1$ and $S2$. These windows help determine confirmed traffic within the network. In the second receive window, $S2$ end devices wait for an acknowledgement that saves channel resources and energy. The symbol duration of LoRa is based on bandwidth BW and the spreading factor SF. The LoRa symbol duration V_t is calculated using Equation (1).

$$V_t = \frac{2^{SF}}{BW} \tag{1}$$

The gateway sends out acknowledgements at a set spreading factor. The variable F^0 indicates the path loss exponent in LoRa communication range, which is influenced by path loss. The path loss P_{path} is calculated using Equation (2).

$$P_{path} = \left(\frac{4.\pi.h}{d}\right)^2 .f^0 \tag{2}$$

The LoRa frequency is noted by h and the link budget P_{bud} is measured using Equation (3).

$$P_{bud} = \frac{P_s}{U_s(tg, cx)} \tag{3}$$

Here, P_s refers to transmission power and $U_s(tg,cx)$ represents receiver sensitivity, which is influenced by bandwidth and spreading factor. The lowest power needed to detect the signal is the receiver sensitivity. Calculate SNR_0 using Equation (4).

$$SNR_0 = \frac{G_{bit}}{O_0} \tag{4}$$

Here, O_0 the noise power density. The parameter is taken to $G_{bit} = U_s.V_{bit}$. The received power is represented by U_s , and the bit duration is indicated by V_{bit}. The formula above is rephrased using Equation (5).

$$SNR_0 = \frac{U_s.2^{tg}}{og.m.v.cx} \tag{5}$$

where the term U_s is evaluated using Equation (6).

$$U_s = \frac{SNR_0.o.m.V.cx}{2^{tg}} \tag{6}$$

Here, receiver sensitivity $U_s(tg,cx)$ is analysed using Equation (7).

$$\begin{aligned} U_s(tg,cx) &= SNR(tg).o_0 \\ &= SNR(tg).og.m.V.cx \end{aligned} \tag{7}$$

Here, the Kelvin constant, noise, and temperature are represented by m, og and V. The path loss factor f is calculated using Equation (8).

$$f = \left(\frac{P_{path}}{\left(\frac{4.\pi.h}{d} \right)^2} \right)^{\frac{1}{0}} \tag{8}$$

Here, higher spreading factor values are utilised to achieve extended LoRa ranges. Therefore, the range of LoRa is enhanced according to the spreading factor. The specifics of LoRa modulation and the precise radio environment are recorded during uplink transmission. Every conceivable LoRa parameter is employed to minimise packet loss in uplink transmission.

Problem Formulation

Throughput

The throughput is referred to as the total number of tasks which are processed for a given period. It is the essential factor for system performance assessments and reliability, which is expressed as,

$$T_p = \sum_{x=1}^{N} C_t / T \tag{9}$$

Where, C_t is mentioned by the completed tasks are mentioned and T is represented as the starting time of the execution of the task

Energy efficiency

Energy efficiency is all about how much data gets transmitted successfully for each unit of energy used during the transmission. It is defined as,

$$EE = \frac{E_{succ}}{F_{tot}} \tag{10}$$

Here, E_{succ} defines total successfully delivered bits and F_{tot} denotes total energy consumed in transmission.

Latency

It is computed by the amount of time required to send a packet from the source to the destination. It encompasses processing, waiting, and transmission delays, which are calculated as

$$Delay = (Br)_t - (Hr)_t \tag{11}$$

Here, $(Br)_t$ denotes packet arrival time and $(Hr)_t$ is the generation time.

Transmission Rate

The transmission rate is measured by how much data is successfully received over a certain period.

$$TR = R_c.(1 - PLR) \tag{12}$$

Here, R_c denotes raw data rate, and PLR is the packet loss ratio.

Resource allocation by Hybrid reinforcement learning framework: Double deep Q-Learning based actor critic mechanism

A reinforcement learning [29][300] agent is divided into two main parts as actor (policy) and critic (value function). The main goal of using Hybrid Reinforcement Learning with double deep Q-Learning-based actor-critic (Hy-DeoQ-AC) for resource allocation is to enhance how network resources are used in LoRa-based IoT networks. LoRa is crucial to efficiently allocate key parameters like transmission power, spreading factor, and channel to satisfy increasing needs for energy efficiency, higher throughput, lower latency, and wider transmission range. Hy-DeoQ-AC mechanism addresses these issues by learning the best resource allocation strategies through ongoing interaction with the network environment. This smart and adaptive method results in more intelligent and sustainable IoT communication in resource-limited settings like LoRa networks. The double deep Q learning part helps reduce the overestimation bias by using two Q networks as one for action selection and another for evaluation. This results in steadier learning and more dependable decision-making. Through repeated interactions with the environment, the reinforcement learning agents figure out how to allocate network resources in a way that optimises energy use, boosts throughput, cuts down latency, and guarantees stable transmission rates. During the learning phase, the gradient of performance is estimated directly with respect to the actor parameters that are then adjusted to enhance performance. On the other hand, critic methods depend on approximating the value function by finding near near-optimal solution to the Bellman equation, which helps derive a near-optimal policy. Still, these methods often lack reliable assurances for achieving convergence and finding the optimal policy, even if a good approximation of the value function is created. The actor component provides advantages in convergence performance and the ability to compute continuous actions, while the critic uses an approximation framework to estimate a value function, which offers low variance insights into performance and is then used to update actor policy parameters. Actor-critic methods are recognised as policy gradient methods, boasting better convergence properties than critic-only methods. Moreover, due to variance reduction, actor-critic methods converge faster. In Q-learning, the Q-value is a key component of the learning algorithm. For smaller problems, a Q value pair $Q(U,b)$ can be stored in computer memory. But for larger problems, the number of Q-value pairs can become huge or even infinite. So, it is not practical to keep all possible pairs in memory; estimating Q value is crucial when using Q-learning. The aim is to learn behaviour and to allocate resources, which is why we suggest using actor-critic network to achieve both value and behaviour estimation. The Critic network handles value estimation.

When agents choose an action and find themselves in a new state, the critic network estimates potential action values based on the current state to assist the agent in deciding the next action. The Actor network is akin to the critic network, but it estimates behaviour distribution when the agent reaches a new state and takes an appropriate action based on the feedback from the critic network from the previous step. To solve issues faced in Q learning, actor critic network is used.

Actor-critic methods

The actor-critic framework employs two networks or function approximators to develop and finalise the training of the agent, which proves effective in continuous action control tasks. Fig. 3 defines the Actor-Critic approach.

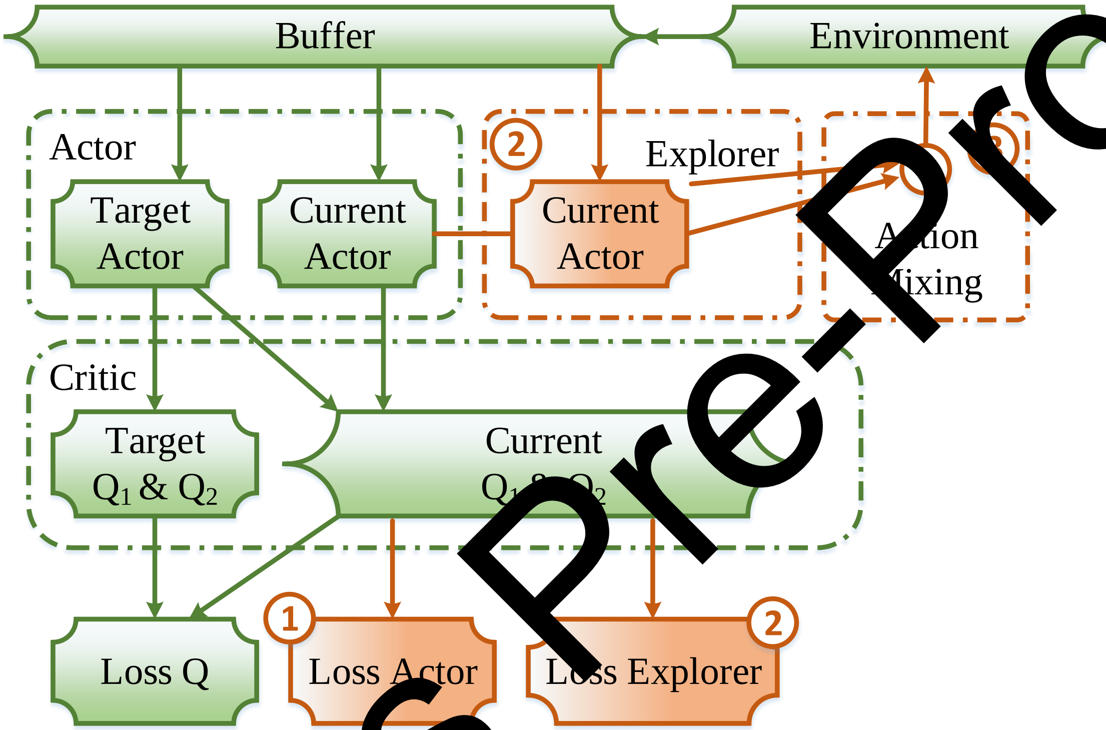


Fig 3. Actor-critic mechanism

Both the Q-based value function approach and the policy gradient approach can utilise actor-critic framework. The actor represents an action policy network $\pi(U)$, while the critic serves as an estimate of current action value $Q(U,b)$ or state value $V(U)$.

Value function method

In the value function method, the agent computes the current value function $Q_{\pi}(U,b)$ directly, where Q represents the action value for the current action and state. This is optimised using Bellman $\left[Q(U_t,b_t)-z\right]^2$, where $Z = r_{t+1}+\gamma \max_{b'\in B} Q^*(u_{t+1},b')$. Therefore, the current action value function $Q_{\pi}(U,b)$ is always moving closer to the optimal action value function $Q^*(u_{t+1},b')$. The critic objective is outlined in Equation (13).

$$J(Q_{\theta}) = \mathrm{E}_{(u,b,u')\sim E}[(r(u,b)+\gamma Q_{\theta}(u',\pi_{\phi}(u'))-Q_{\theta}(u,b))^2] \tag{13}$$

Here, θ denotes the parameter of the value function network Q, ϕ denotes the parameter of the action policy network π, E and describes the state-action domain of the environment. The action policy network aims to choose an action that maximises the Q value function in the current state. The actor's objective is defined in Equation (14)

$$J(\pi_\phi) = \mathrm{E}_{u\sim E}[Q_\theta(u, \pi_\phi(u))] \tag{14}$$

Deep deterministic policy gradient (DDPG) is a classic value function algorithm that uses a target network to enhance training stability and manages target network update rate through soft updates. The goal of its value function is presented in Equation (15)

$$J(Q_\theta) = \mathrm{E}_{(u,B,u')\sim c}[(r(u,b) + \gamma \hat{Q}_{\theta'}(u', \hat{\pi}_{\phi'}(u')) - Q_\theta(u,b))^2] \tag{15}$$

Here $\hat{Q}_{\theta'}, \hat{\pi}_{\phi'}$ corresponds to target networks, c denotes the experience relay buffer gotten by the agent interacting with the environment. Temporal difference error (TD) is suggested by minimising the value function, implementing policy delays, and adding target noise to reduce overestimation bias. Therefore, its value function objective is represented by Equation (16).

$$J(Q_\theta) = \mathrm{E}_{(u,B,u')\sim c}[(r(u,b) + \gamma \min_{j=1,2} \hat{Q}_{\theta_j}, (u', \hat{\pi}_{\phi'}(u') + \hat{\epsilon}) - Q_\theta(u,b))^2]] \tag{16}$$

Here $\hat{Q}_{\theta'}, \hat{\pi}_{\phi'}$ agree to target networks, the relay buffer is defined as c, $\hat{\epsilon}_t \in \sim clip(N(0,\sigma^2), -d, d)$ which denotes random action noise following a clipped Gaussian distribution.

Policy gradient method

The policy gradient approach involves utilising the current action policy network that engages with the environment and gathers trajectory data η. The agent computes policy gradient and tweaks parameters, which can boost the chances of picking a higher value action and keep optimising for a better policy. The critic represents the state value function $W(U)$, where W indicates the average state value that is linked to the current state. The goal of the state value is illustrated in Equation (17).

$$J(W_\theta) = \mathrm{E}_{u_T \in \eta, \eta\sim\pi_\phi}[(R(u_t) - w_\theta(u_t))^2] \tag{17}$$

Here, $R(u_t) = \sum_{k=t}^{n} \gamma^{k-t} r_k$ denotes the cumulative reward of states u_t, η and defines current policy π_ϕ sampling in the environment.

$$J(\pi_\phi) = \mathrm{E}_{u_T \in \eta, \eta\sim\pi_\phi}[(R(U_t) - W_\theta(U_t)) \log S(\pi_\phi(U))] \tag{18}$$

Here, the actor adjusts action selection probability in terms of state value and objective as defined in Equation (18)

Fine-tuning by the Hybrid Levy flight assisted rabbit optimisation algorithm

Rabbit optimisation algorithm [23][25] is mainly based on two laws of rabbit survival found in nature as detour foraging and random hiding. This hybrid approach improves energy efficiency, throughput, and minimises delay and transmission rate. The Hybrid Levy Flight Assisted Rabbit optimisation algorithm (Hy-LevRBO) is designed to refine network parameters chosen by a reinforcement learning model. However, the Hy-DeoQ-AC mechanism sets initial best resource allocation, such as transmission power, spreading factor, and channel. Also, fine-tuning is essential to enhance performance and adjust to changing network conditions. Hy-LevRBO merges two effective strategies. One strategy is the rabbit optimisation Algorithm (RBO), which draws inspiration from the clever foraging habits of rabbits. The second strategy is Levy flight, which is a random walk method based on the Levy distribution that enables occasional long jumps during the search process. This capability assists the algorithm in avoiding local optima and seeking out a wider range of solutions. The fitness function is defined as,

$$Fitness = \max imization(energy\ efficiency, throughput) \tag{19}$$

Foraging is an exploration tactic that helps rabbits avoid being spotted by predators by munching on grass close to their nests. On the other hand, random hiding involves rabbits relocating to different burrows to conceal themselves better. Every search algorithm kicks off with an initialisation process. If design variable size is considered as a dimension D, the artificial search agent colony size is considered n, and the upper and lower limits as UB and LB, initialisation proceeds as follows,

$$\vec{y}_{j,k} = r.(UB_k - LB_k) + LB_k,\ k = 1,2,...., D \tag{20}$$

Here, $\vec{y}_{j,k}$ represents the position of the dimension of i^{th} search agent and r is a random number provided alongside it. The metaheuristic algorithm primarily focuses on two processes as exploration and exploitation, whereas detour foraging mainly emphasises the exploration phase. Detour foraging refers to each search agent's inclination to wander around the parameter and randomly explore another search agent's location within the group to gather sufficient information. Below is the updated formula for detour foraging,

$$\vec{W}_j(t+1) = \vec{y}_j(t) + r.(\vec{y}_j(t) - \vec{y}_i(t)) + round(0.5.(0.05 + r_1)).N_1 \tag{21}$$

$$r = M.B \tag{22}$$

$$M = (f - f^{(\frac{t-1}{T})^2}).\sin(2\pi r_2) \tag{23}$$

$$B(k) = \begin{cases} 1 & if\ \ k == H(m) \\ 0 & else \end{cases} \quad k = 1,......, d\ \ and\ \ m = 1,....., [r_3.d] \tag{24}$$

$$H = randq(d) \tag{25}$$

$$n_1 \sim N(0,1) \tag{26}$$

Here, $\vec{W}_j(t+1)$ indicates the new position of the search agent, $\vec{y}_j$ signifies the location of j^{th} search agent and shows artificial rabbits at various other random locations. $t_{\max}$ is the highest number of iterations and $randq$ indicates a random arrangement from $1\ to\ d$ a random permutation of integers. r_1 , r_2 and r_3 are random numbers ranging from 0 to 1. M stands for running length, which is the speed of movement during detour foraging. n_1 follows a standard normal distribution. The perturbation is primarily shown by a normal distribution of random numbers of n_1. Random hiding is mainly modelled after the exploration phase of the algorithm, where the search agent typically digs several search spaces and randomly selects one to hide in to lower the chances of being preyed upon. Initially, the outline method by which rabbits search agents creates search space. The j^{th} search agent generates i^{th} search space by,

$$\vec{C}_{j,i}(t+1) = \vec{y}_j(t) + I.h.\vec{y}_j(t) \tag{27}$$

$$I = \frac{T_{\max} - t + 1}{T_{\max}} n_2 \tag{28}$$

$$n_2 \sim N(0,1) \tag{29}$$

$$h(k) = \begin{cases} 1 & if \quad k == i \\ 0 & else \end{cases} mk = 1,......, d \tag{30}$$

Here $j = 1,......, N \ and \ i = 1,....., d$, and n_2 adheres to the standard normal distribution. I represents a hidden parameter gradually decreasing from $1 \ to \ 1/T_{\max}$ with random perturbations. The formula for updating the random hiding method is displayed as,

$$\vec{W}_j(t+1) = \vec{y}_j(t) + R.(r_4.\vec{C}_{j,r}(t) - \vec{y}_i(t)) \tag{31}$$

$$h_r(k) = \begin{cases} 1 & if \quad k == [r_5.d] \\ 0 & else \end{cases} m \ k = 1,....., d \tag{32}$$

$$\vec{c}_{j,r}(t) = \vec{y}_j(t) + I.h_r.\vec{y}_j(t) \tag{33}$$

Here $\vec{W}_j(t+1)$ is the updated position of the search agent, $\vec{c}_{j,r}(t)$ which indicates a randomly chosen search space from d the search space created by the search agent for concealment. r_4 and r_5 are random numbers provided within range of 0 to 1. Once two update strategies are applied, refresh the position of j^{th} search agent using Equation (34).

$$\vec{y}_j(t+1) = \begin{cases} \vec{y}_j(t) & if \ g(\vec{y}_j(t)) \ \leq \ g(\vec{W}_j(t+1)) \\ \vec{W}_j(t+1) & else \ \ g(\vec{y}_j(t)) > g(\vec{y}_j(t+1)) \end{cases} \tag{34}$$

Here, this equation shows an adaptive update. The search agent instinctively decides whether to remain where it is or shift to a new spot based on the adaptation value. In an optimisation algorithm, populations tend to focus on the exploration phase at the beginning and switch to an exploitation phase in the middle and later stages. Artificial Rabbit Optimisation (ARO) uses the energy of the search agent to create a finding scheme as the search agent energy diminishes over time, effectively mimicking the transition from exploration to exploitation. The way to define the energy factor in the search agent algorithm is,

$$Z(t) = 4.\left(1 - \frac{t}{t_{\max}}\right).In \ \frac{1}{r} \tag{35}$$

Here, r represents a specific random number and r falls within the range of (0, 1). The hybrid of Levy flight with rabbit optimisation leverages strengths to enhance algorithm accuracy. Levy flights are commonly used in advanced optimisation algorithms to enhance exploration and avoid local optima. Levy flight operator is primarily used for producing a regular random number, which is usually a small number but can occasionally be a large random number. This random number generation rule can assist different update strategies in introducing dynamics and escaping local solutions. The Levy distribution is defined as,

$$levy(t) \sim v = t^{-1-\gamma}, 0 < \gamma \leq 2 \tag{36}$$

Here, t represents step length, which can be determined using Equation (37). The equations for calculating the step size of the Levy flight are provided in Equations (37)–(40).

$$t = \frac{v}{|w|^{1/\gamma}} \tag{37}$$

$$v \sim N(0, \sigma_V^2), W \sim N(0, \sigma_w^2) \tag{38}$$

$$\sigma_v = \left(\frac{\Gamma(1+\alpha'').\sin(\pi.\alpha/2)}{\Gamma((1+\alpha/2).\alpha.2^{(\alpha-1)/2}} \right)^{1/\alpha} \tag{39}$$

$$\sigma_w = 1 \tag{40}$$

Here, σ_v and σ_w are defined as stated in Equations (39) and (40). Both v and w follow Gaussian distributions with a mean of 0 and variances of σ_V^2 and σ_w^2 as indicated in Equation (39). Γ represents the standard gamma function, while α is the correlation parameter typically set to 1.5. In the random hiding phase, substitute r_4 random numbers with those generated by the Levy flight strategy. Since this stage is about exploitation, incorporate Levy flight into the strategy to prevent rabbit optimisation from getting stuck in local candidate solutions during the exploitation phase. Moreover, it enhances the adaptability of the random hiding stage.

$$\vec{W}_j(t+1) = \vec{y}_j(t) + R.(\beta.levy(\alpha).\vec{C}_j(t) - \vec{y}_j(t)), j = 1, ..., n \tag{41}$$

Here, Equation (41) outlines the random hidden phase based on Levy flight, where β is fixed parameter is set to 0.1. The Pseudocode is defined in Algorithm 1.

The structure of artificial rabbit optimisation
Start
Input: Channel, transmission power, spreading factor
The search agent parameters include the number of search agents n and $T_{\max}$.
Randomly initialise a group of search agents y_j and calculate g_j
Identify the best search agent.
While $t \le t_{\max}$ **do**
  For $j = 1$ to n do
    Calculate energy factor Z using Equation (35).
    If $Z > 1$ **then**
      Randomly select a search agent from the entire group.
      Update the search agent position using Equation (34).
    Else
      Generate d search space and randomly choose one based on Equation (33).
      Implement a random hiding strategy according to Equation (31).
      Calculate the fitness value of the search agent's position.
      Update the search agent position using Equation (34).
    End if
  End for
  Look for the best search agent.
  $t = t + 1$
  The Levy flight is included in Equation (41) to enhance accuracy.
End while
Stop
Output: Improve energy efficiency, minimise latency

IV. RESULTS AND DISCUSSION

The superiority of the proposed approach is demonstrated by evaluating the complete performance of the proposed approach with different existing models. The proposed model of LoRa is implemented using the NS3 tool. Performance metrics like delay, energy consumption, energy efficiency, execution time, remaining resources, SINR, throughput, and transmission rate [24] are evaluated and compared with recent research models. The proposed model's efficiency is analysed by different existing approaches. Table 2 denotes the simulation parameters of the proposed model.

Table 2. Simulation parameter

Parameter	Number of values used
Channel	1-20
Power transmission	0-1
IoT device	5
Gateway	[illegible]
Spread factor	[illegible]

Performance evaluation of the proposed model along with existing approaches

The proposed model performance is analysed with existing approaches like Powered by energy harvesting-quality of service (PEH-QoS), classic Q learning, and Modified Q learning.

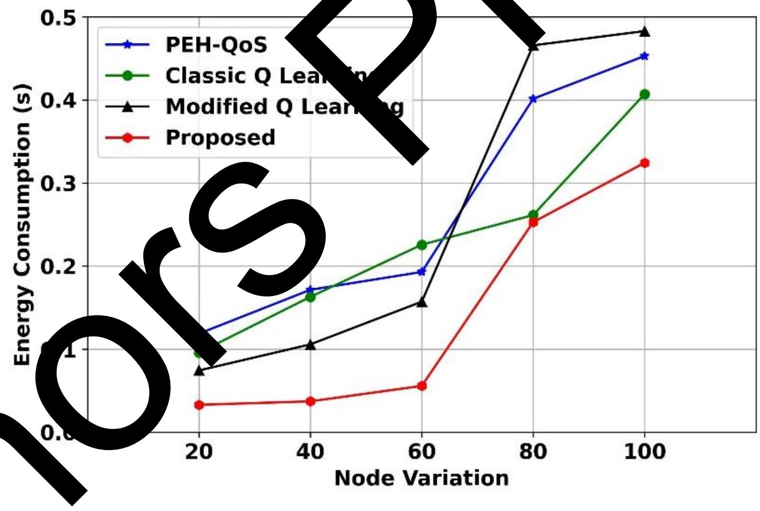


Fig 4. Energy consumption

Energy consumption evaluation is defined in Fig. 4. The graph shows how energy consumption compares across different node densities for four models. As the number of nodes increases from 20 to 100, all models see a significant rise in energy use. PEH-QoS struggles with resource allocation because its static design leads to higher energy consumption, especially at denser node setups. Classic Q learning is more dynamic but has issues with slow convergence and less effective exploration strategies that result in moderate energy savings but not enough adaptability in crowded networks. Modified Q learning does better than the classic version by improving update strategies, but it still faces problems with premature convergence and can get stuck in local optima, which causes a sharp rise in energy use as node numbers increase. The proposed model uses a hybrid double deep Q-learning based actor critic (Hy-DeoQ-AC) method along with hybrid Levy flight assisted rabbit optimisation (Hy-LevRBO). It shows improved performance by optimising key transmission parameters like transmission power, spreading factor, and channel selection in real time. This optimisation approach facilitates efficient exploration of the solution space and precise exploitation of the best configurations, greatly reducing energy consumption even as node density goes up. By striking a good balance between exploration and exploitation, the proposed model keeps energy use low across all node variations, which effectively addresses the main shortcomings of the existing models.

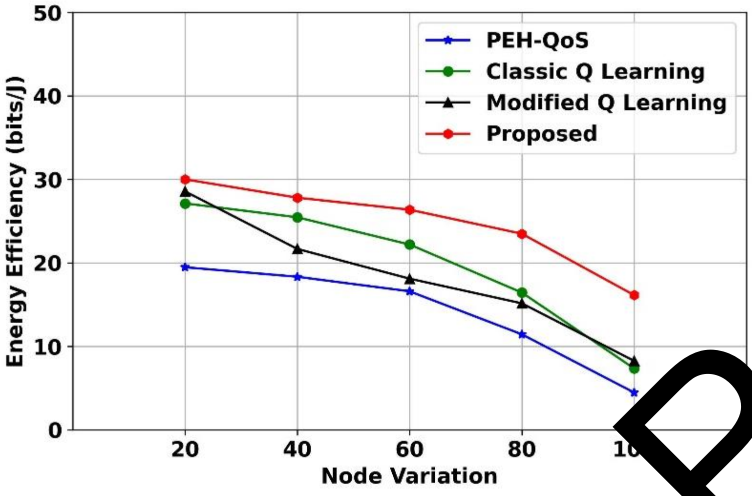


Fig 5. Energy Efficiency

Fig. 5 illustrates how energy efficiency (bits/Joule) changes with increasing node density for various approaches. As the number of nodes increases, all techniques experience energy efficiency drops due to more interference, congestion, and poor resource allocation in PEH-QoS consistently shows low energy efficiency, which makes it a poor fit for dense IoT settings. Classic Q learning does better than PEH-QoS by using a learning-based approach but faces issues with limited learning depth, which leads to inefficient resource allocation when network stress rises. Modified Q learning improves the Q value update process but still doesn't have enough global search capability, resulting in lower energy efficiency as node density increases. The proposed model achieves much higher energy efficiency across all node densities. This allows the model to effectively find the best parameter configurations through reinforced exploration and detailed local search. The hybrid Hy-LevRBO tuning mechanism ensures that even with rising network loads, the selected transmission parameters maintain higher data rates compared to energy consumption, which solves scalability issues faced by existing approaches.

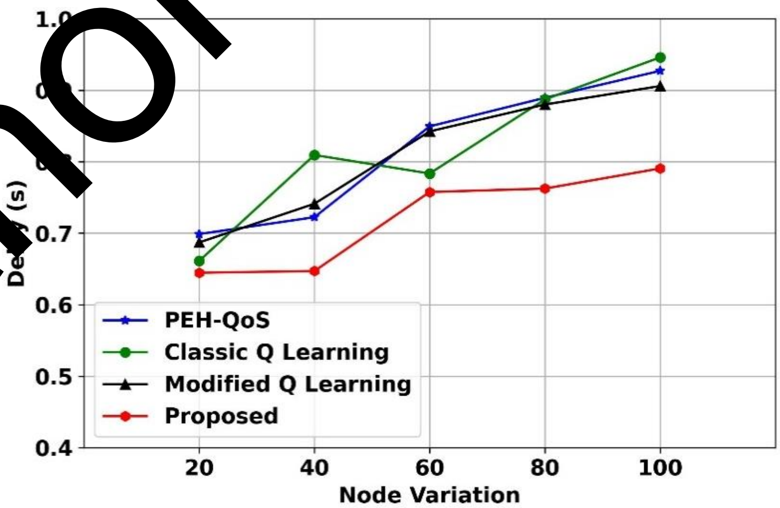


Fig. 6. Delay

Fig. 6 describes delay analysis. The graph shows how delay changes as the number of nodes increases across various models. As node count goes up, overall network delay rises for all methods due to increased contention and queuing delays. PEH-QoS consistently experiences high delays because of its fixed resource allocation, which causes congestion during peak traffic. Classic Q learning struggles to adapt in real time to quick changes in network topology. Modified Q learning faces issues with balancing exploration and exploitation. Moreover, use of Hy-LevRBO fine-tunes resource allocation in real time, cutting down on queuing and transmission delays even in high-density node situations. This results in consistently lower delays compared to other methods, which makes the proposed model very effective at reducing transmission latency in scalable environments.

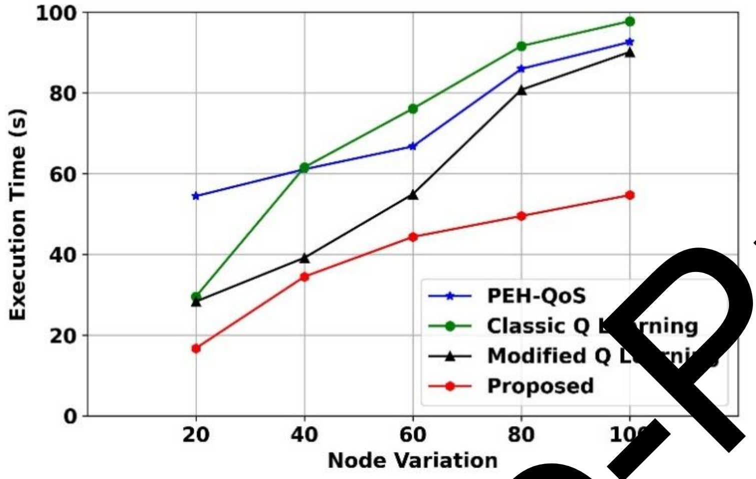


Fig 7. Execution time

Execution time evaluation is denoted in Fig. 7. As the node count goes up, execution time for PEH-QoS consistently rises because it depends on costly static algorithms that do not scale well. Classic Q learning takes a long time to execute, mainly due to its single agent learning setup, which needs several iterations to converge in large action spaces, making it less efficient in dense environments. Modified Q learning has scalability challenges since it adds extra computational load to adjust learning rates and balance exploration. The proposed approach uses Hy-DeoQ-AC with parallel reinforcement agents and Hy-LevRBO, which makes the optimisation process smoother. This hybrid approach reduces unnecessary computations by concentrating on both global searching and local fine-tuning at the same time, significantly reducing execution time across all node variations. By smartly directing learning and tuning processes, the proposed model effectively improves execution efficiency in high-density networks.

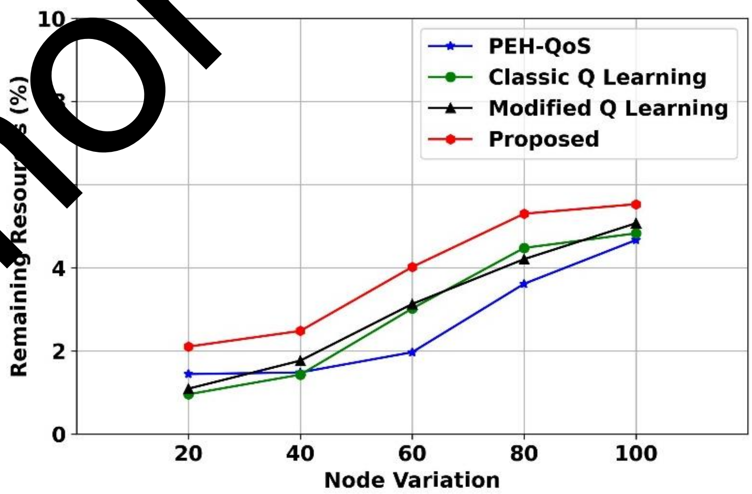


Fig 8. Remaining resources

The remaining resources analysis is illustrated in Fig. 8. The graph shows how the percentage of leftover resources in the network changes with increasing number of nodes. One ongoing issue with existing methods is poor use of network resources as node density rises. PEH-QoS results in fewer remaining resources because its static resource management strategies do not adapt dynamically, which leads to quick depletion of bandwidth, power, and channel availability. While classic Q learning is better at adapting, it does not have the fine control needed for resource distribution that causes unnecessary resource use in some areas of the network. Modified Q struggles with premature convergence, often overlooking globally optimal allocations and thus limiting resource conservation. The proposed model, featuring the Hy-DeoQ-AC framework enhanced by Hy-LevRBO, stands out by smartly distributing transmission power, spreading factor and channel assignments. This hybrid approach maximises resource retention by balancing the use of known configurations with the exploration of new options, ensuring efficient use of available resources. As a result, the proposed model consistently achieves higher percentages of remaining resources, which leads to more flexibility for future transmissions.

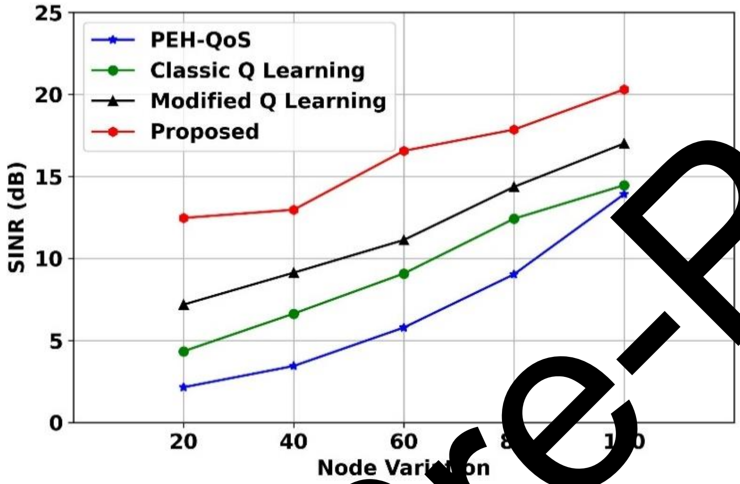


Fig 9. SINR evaluation

Fig. 9 shows the SINR evaluation. PEH-QoS consistently results in lower SINR values because its static optimisation method cannot adapt to growing interference in crowded node settings. This leads to poor signal quality and a higher chance of packet loss. Classic Q-learning has difficulty dealing with complex interference patterns, which results in less efficient channel use and ongoing interference problems. Modified Q learning faces some limitations in convergence and focusing on local optima still stops it from achieving consistently high SINR. The proposed model tackles these issues by dynamically adjusting transmission parameters like spreading factor, transmission power, and channel assignment based on changing interference levels. This hybrid approach boosts communication reliability and enhances overall network robustness and data integrity.

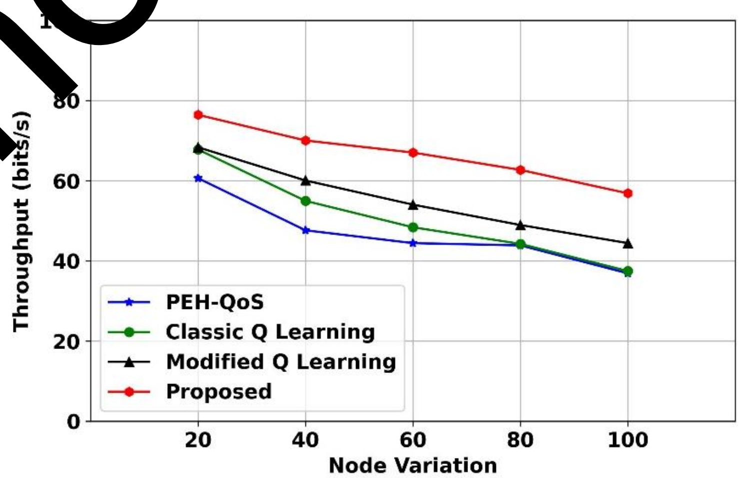


Fig 10. Throughput

Fig. 10 describes throughput changes as node density increases. PEH-QoS experiences a sharp drop in throughput with rising node variation, primarily because it cannot adjust dynamically to congestion and interference that lead to frequent retransmissions and data collisions. Classic Q learning has slower convergence in denser environments that holds it back. Modified Q learning faces issues in optimally managing transmission parameters in real time. The proposed Hy-DeoQ-AC with Hy-LevRBO mechanism makes better use of resources by optimising transmission power, spreading factor, and channel allocation all at once. Its hybrid design permits quicker adjustments to changing network loads, reducing packet collisions, and improving data delivery rates. As a result, the proposed model consistently maintains higher throughput, which ensures more stable and efficient data transmission even when node densities are high.

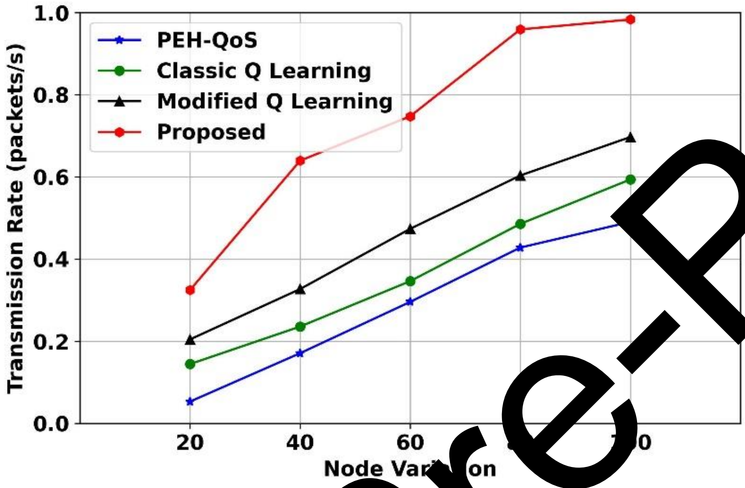


Fig 11. Transmission Rate

Fig. 11 illustrates how increasing node variation affects transmission rate. PEH-QoS results in poor spectrum utilisation and packet losses as node congestion increases. Classic Q learning has a limited learning range, making it slower to respond to changing transmission conditions, particularly in crowded environments. Modified Q learning improves the learning process but still facing problems with partial optimisation of resource allocation, which restricts its ability to achieve optimal packet dispatching under load. The proposed Hy-DeoQ-AC with Hy-LevRBO model outperforms all existing schemes by dynamically optimising essential transmission factors. This allows the network to adapt quickly, which maximises transmission rate even in high node density situations. As a result, the proposed method reaches near-optimal packet transmission rates and demonstrates adaptability and resource management capabilities. Table 3 defines values of existing and proposed models.

Table 3. Performance analysis values of the proposed and existing models

			Energy consumption		
Models	20	40	60	80	100
PEH-QoS	0.1186	0.1714	0.1929	0.4014	0.4529
CQL	0.0957	0.1629	0.2257	0.2614	0.4071
MQL	0.0743	0.1057	0.1571	0.4657	0.4829
Proposed	**0.0329**	**0.0371**	**0.0557**	**0.2529**	**0.3243**
			Energy efficiency (bits/J)		
Models	20	40	60	80	100
PEH-QoS	19.4697	18.3333	16.5909	11.4394	4.4697
CQL	27.1212	25.4545	22.197	16.4394	7.3485
MQL	28.5606	21.6667	18.1061	15.1515	8.2576

Proposed	**30.0**	**27.803**	**26.3636**	**23.4848**	**16.1364**
			Delay		
Models	20	40	60	80	100
PEH-QoS	0.6988	0.7224	0.8494	0.8894	0.9271
CQL	0.6612	0.8094	0.7835	0.8871	0.4071
MQL	0.6871	0.7412	0.8424	0.88	0.9058
Proposed	**0.6447**	**0.6471**	**0.7576**	**0.7624**	**0.7096**
			Execution time (S)		
Models	20	40	60	80	100
PEH-QoS	54.4335	61.0837	66.7488	85.9606	92.[illegible]10[illegible]
CQL	29.5567	61.5764	76.1084	91.62[illegible]	97.[illegible]33
MQL	28.3251	39.1626	54.9261	80.[illegible]82	90.147[illegible]
Proposed	**16.7488**	**34.4828**	**44.335**	**49.507[illegible]**	**54.6798**
			Remaining resources (%)		
Models	20	40	60	80	100
PEH-QoS	1.4427	1.4831	1.9685	[illegible].6135	4.6652
CQL	0.9573	1.4292	3.0[illegible]2	[illegible].4764	4.827
MQL	1.0921	1.7663	[illegible].1281	4.2067	5.0697
Proposed	**2.1034**	**2.480[illegible]**	**[illegible].018**	**5.2989**	**5.5281**
			[illegible] (dB)		
Models	20	40	60	80	100
PEH-QoS	2.1446	[illegible]414	5.7855	9.0274	13.9152
CQL	4.3392	6.6334	9.0773	12.419	14.4638
MQL	7.182	9.1272	11.1222	14.3641	17.0075
Proposed	**12.[illegible]**	**12.9676**	**16.5586**	**17.8554**	**20.2993**
			Throughput (bits/sec)		
Models	20	40	60	80	100
PEH-QoS	[illegible].6118	47.6235	44.4235	43.8588	36.8941
CQL	67.7647	54.9647	48.3765	44.2353	37.4588
MQL	68.3294	60.0471	54.0235	48.9412	44.4235
Proposed	**76.4235**	**70.0235**	**67.0118**	**62.6824**	**56.8471**
			Transmission rate (packet/sec)		
Models	20	40	60	80	100
PEH-QoS	0.0529	0.1707	0.2957	0.4279	0.4904
CQL	0.1442	0.2356	0.3462	0.4856	0.5938
MQL	0.2043	0.3269	0.4736	0.6034	0.6971
Proposed	**0.3245**	**0.6394**	**0.7476**	**0.9591**	**0.9832**

Discussion

The presented approach improves efficiency based on a hybrid reinforcement learning approach. The existing PEH-QoS is associated with high latency and energy consumption during changing network conditions, limiting its use in real-time IoT applications. The classic Q learning method has trouble with convergence speed and scalability in large networks. Modified Q Learning does enhance parameter selection but adds complexity to optimisation and can lead to inconsistent performance with different interference levels. The proposed model solves this issue by incorporating Hybrid Levy Flight Assisted Rabbit Optimisation, which not only streamlines optimisation search but also guarantees stable and adaptable performance, even when interference and load conditions vary. By merging advanced reinforcement learning with metaheuristic optimisation, the proposed method strikes an excellent balance between energy efficiency, throughput, and reduced execution time, clearly surpassing both traditional and modified reinforcement learning techniques in thorough simulation tests.

V. CONCLUSION

This research introduces a promising solution to tackle the main issues of resource allocation in LoRa-based networks by presenting a hybrid optimisation model that merges Hy-DeoQ-AC with Hy-LevRBO. The suggested model effectively deals with typical drawbacks of traditional resource allocation techniques, including higher packet loss, interference, excessive energy use, limited coverage, and lower transmission speeds. By smartly optimising transmission parameters such as transmission power, spreading factor, and channel selection, the proposed model guarantees efficient data transmission while boosting QoS for IoT devices. The combination of Hy-DeoQ-AC for intelligent learning, and Hy-LevRBO for fine-tuning allows the system to adapt dynamically to changing network conditions. The proposed model attains a throughput of 96.8471 (bits/s), energy efficiency of 16.1364 (bits/J), which leads to improved use of network resources. Looking forward, there are some meaningful ways to expand on this work. Future studies could explore the implementation of distributed reinforcement learning to decentralise the optimisation process across various nodes to enhance scalability and fault tolerance in large IoT networks. Moreover, integrating this hybrid optimisation model with mobility-aware algorithms could enhance adaptability in settings with mobile nodes or changing network topologies. The proposed method might also be broadened to include energy harvesting mechanisms that enable IoT devices to operate for extended periods without needing manual battery changes.

Acknowledgement
The author expresses sincere gratitude to the Department of Electronics & Communication Engineering, Canara Engineering College, Bantwal, for their valuable support of this study.

Credit Author Statement
The authors confirm their contribution to the paper as follows:
Conceptualisation: Suchitra N Shenoy, Ganesh V Bhat, H. Manoj T. Gadiyar; **Writing- Original Draft:** Suchitra N Shenoy; **Validation**: Ganesh V Bhat, H. Manoj T. Gadiyar; **Supervision**: H. Manoj T. Gadiyar. All authors reviewed the results and approved the final version of the manuscript.

Data Availability
The data used in this study were generated through simulations conducted as part of the proposed model's performance evaluation.

Conflicts of Interest
The author(s) declare(s) that they have no conflicts of interest.

Funding
No funding agency is associated with this research.

Competing Interests
There are no competing interests.

References

[1]. Jouhari, M., Saeed, N., Alouini, M.-S., & Amhoud, E. M. (2023). A survey on scalable LoRaWAN for massive IoT: Recent advances, potentials, and challenges. *IEEE Communications Surveys & Tutorials, 25*(3), 1841–1876.

[2]. Cheikh, I., Aouami, R., Sabir, E., Sadik, M., & Roy, S. (2022). Multi-layered energy efficiency in LoRa-WAN networks: A tutorial. *IEEE Access, 10*, 9198–9231.

[3]. Farhad, A., & Pyun, J.-Y. (2023). Lorawan meets ML: A survey on enhancing performance with machine learning. *Sensors, 23*(15), 6851.

[4]. Acosta-Garcia, L., Aznar-Poveda, J., Garcia-Sanchez, A. J., Garcia-Haro, J., & Fahringer, T. (2023). Dynamic transmission policy for enhancing LoRa network performance: A deep reinforcement learning approach. *Internet of Things, 24*, 100974.

[5]. Alkhayyal, M., & Mostafa, A. (2024). Recent developments in AI and ML for IoT: A systematic literature review on LoRaWAN energy efficiency and performance optimization. *Sensors, 24*(14), 4482.

[6]. Pandey, O. J., Yuvaraj, T., Paul, J. K., Nguyen, H. H., Gundepudi, K., & Shukla, M. K. (2021). Improving energy efficiency and QoS of LPWANs for IoT using Q-learning based data routing. *IEEE Transactions on Cognitive Communications and Networking, 8*(1), 365–379.

[7]. Chen, M., Mokdad, L., Ben-Othman, J., & Fourneau, J.-M. (2023). Dynamic parameter allocation with reinforcement learning for LoRaWAN. *IEEE Internet of Things Journal, 10*(12), 10250–10265.

[8]. Sisinni, E., Depari, A., Bellagente, P., Flammini, A., Silva, I., Flores, T., & Ferrari, P. (2023). Can adaptive strategies sustain bidirectional LoRaWAN traffic? In *2023 IEEE International Workshop on Metrology for Industry 4.0 & IoT (MetroInd4.0 & IoT)* (pp. 66–71). IEEE.

[9]. Farhad, A., Kim, D.-H., Kim, B.-H., Mohammed, A. F. Y., & Pyun, J.-Y. (2020). Mobility-aware resource assignment to IoT applications in long-range wide area networks. *IEEE Access, 8*, 186111–186124.

[10]. Dai, B., Niu, J., Ren, T., & Atiquzzaman, M. (2022). Toward mobility-aware computation offloading and resource allocation in end–edge–cloud orchestrated computing. *IEEE Internet of Things Journal, 9*(19), 19450–19462.

[11]. Lasri, K., Ben Maissa, Y., Echabbi, L., Iova, O., & Valois, F. (2023). Probabilistic and distributed traffic control in LPWANs. *Ad Hoc Networks, 143*, 103121.

[12]. S. K. Suman, S. Pandey, V. K. Kumar, S. Balamurugan, M. Pingh[illegible] and P. Tanna, "Optimizing Point of Care Interaction with Speech Recognition to Preserve Learnability and Intelligibility," 2025 4th International Conference on Distributed Computing and Electrical Circuits and Electronics (ICDCECE), Ballari, India, 2025, pp. 1-5, doi: 10.1109/ICDCECE65353.2025.11034855.

[13]. Yao, F., Ding, Y., Hong, S., & Yang, S.-H. (2022). A survey on evolved LoRa-based communication technologies for emerging Internet of Things applications. *International Journal of Network Dynamics and Intelligence*, 4–19.

[14]. Anitha, C., Priscila, S. S., & Balakrishnan, B. (2023, March). An Efficient Secure Routing and Hierarchical Approach to Elliptic Curve Cryptography Combined with Distributed Group Key Management. In 2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS) (Vol. 1, pp. 2208-2212). IEEE.

[15]. Wu, W., Wang, H., & Chen, L. (2023). ReLoRaWAN: Reliable data delivery in LoRaWAN networks with multiple gateways. *Ad Hoc Networks, 1[illegible]7*, 103203.

[16]. Azizi, F., Teymuri, B., Aslani, R., Rasti, M., Tolvanen, J., & Nardelli, P. H. J. (2022, June). MIX-MAB: Reinforcement learning-based resource allocation algorithm for LoRaWAN. In *Proceedings of the IEEE 95th Vehicular Technology Conference* (pp. 1–[illegible]).

[17]. Rao, M. R., & Sundar, S. (2023). An efficient method for optimal allocation of resources in LPWAN using hybrid coati-energy valley optimization algorithm based on reinforcement learning. *IEEE Access, 11*, 116169–116182.

[18]. Minhaj, S. U., Mahmood, A., Abedin, S. F., Hassan, S. A., Bhatti, M. T., Ali, S. H., & Gidlund, M. (2023). Intelligent resource allocation in LoRaWAN using machine learning techniques. *IEEE Access, 11*, 10092–10106.

[19]. Gava, M. A., Rocha, H. R. O., Faber, M. J., Segatto, M. E. V., Wörtche, H., & Silva, J. A. L. (2023). Optimizing resources and increasing the coverage of Internet-of-Things (IoT) networks: An approach based on LoRaWAN. *Sensors, 23*(3), 1239.

[20]. H. P[illegible]dar, V. Roy, M. Bhoyar, A. Gangrade, K. N. Venkata Kowshik and P. Rathod, "Leveraging Context-Aware Fogging for Advanced Remote Patient Monitoring and Healthcare in Modern Life," *2025 4th OPJU International Technology Conference (OTCON) on Smart Computing for Innovation and Advancement in Industry 5.0*, Raigarh, India, 2025, pp. 1-6, doi: 10.1109/OTCON65728.2025.11070467.

[21]. Devasenapathy, Deepa. Bhimaavarapu, Krishna. Kumar, Prem. Sarupriya, S.. Real-Time Classroom Emotion Analysis Using Machine and Deep Learning for Enhanced Student Learning. Journal of Intelligent Systems and Internet of Things , no. (2025): 82-101. DOI: https://doi.org/10.54216/JISIoT.160207

[22]. S. Jain, P. K. Sholapurapu, B. Sharma, M. Nagar, N. Bhatt and N. Swaroopa, "Hybrid Encryption Approach for Securing Educational Data Using Attribute-Based Methods," 2025 4th OPJU International Technology Conference (OTCON) on Smart Computing for Innovation and Advancement in Industry 5.0, Raigarh, India, 2025, pp. 1-6, doi: 10.1109/OTCON65728.2025.11070667

[23]. Chaturvedi, A., Pathak, N., Sharma, N., & Mahaveerakannan, R. (2024, February). Deep Learning in the Context of Artificial Intelligence: Advancements and Applications. In International Conference On Innovative Computing And Communication (pp. 23-44). Singapore: Springer Nature Singapore.

[24]. Xu, W., Cai, G., Fang, Y., Mumtaz, S., & Chen, G. (2022). Performance analysis and resource allocation for a relaying LoRa system considering random nodal distances. *IEEE Transactions on Communications, 70*(3), 1638–1652.

[25]. Hamdi, R., Baccour, E., Erbad, A., Qaraqe, M., & Hamdi, M. (2021). LoRa-RL: Deep reinforcement learning for resource management in hybrid energy LoRa wireless networks. *IEEE Internet of Things Journal, 9*(9), 6458–6476.

[26]. Ji, X., Lu, C., Xie, B., Guo, H., & Zheng, B. (2024). Combination of a Rabbit Optimization Algorithm and a deep-learning-based convolutional neural network–long short-term memory–attention model for arc sag prediction of transmission lines. *Electronics, 13*(23), 4593.

[27]. Bhanurangarao, M., & Mahaveerakannan, R. (2023, September). Urban object detection in UAV imagery for healthcare applications: leveraging the Internet of Things (IoT) in smart cities for enhanced urban healthcare monitoring and management. In 2023 International Conference on Sustainable Emerging Innovations in Engineering and Technology (ICSEIET) (pp. 607-612). IEEE.

[28]. Thomas, A.K., Shekhar, H., Bhagyalakshmi, L., Suman, S.K., Sreelakshmy, R., Bhasha, A. (2025). IoT Enabled Smart Street Light and Air Quality Control. In: Kumar, A., Gunjan, V.K., Senatore, S., Hu, YC. (eds) Proceedings of the 5th International Conference on Data Science, Machine Learning and Applications; Volume 1. ICDSMLA 2023. Lecture Notes in Electrical Engineering, vol 1273. Springer, Singapore. https://doi.org/10.1007/978-981-97-8031-0_90

[29]. H. P. Bhupathi, S. Chinta, S. K. Yeditha, V. Biradar, L. Bhagyalakshmi and S. K. Suman, "Enhanced SOC Estimation Performance Through Noise Covariance Analysis in Nonlinear Kalman Filters," 2025 4th OPJU International Technology Conference (OTCON) on Smart Computing for Innovation and Advancement in Industry 5.0, Raigarh, India, 2025, pp. 1-6, doi: 10.1109/OTCON65728.2025.11070919.

[30]. S. Chinta, H. P. Bhupathi, S. K. Yeditha, V. Biradar, S. K. Suman and L. Bhagyalakshmi, "Optimized Battery Modeling Using Adaptive Forgetting Factor in Recursive Least Squares Algorithm," 2025 4th OPJU International Technology Conference (OTCON) on Smart Computing for Innovation and Advancement in Industry 5.0, Raigarh, India, 2025, pp. 1-6, doi: 10.1109/OTCON65728.2025.11070505.