mathematics

MDPI

Article

An Adaptive Data Rate Algorithm for Power-Constrained End Devices in Long Range Networks

Honggang Wang [1], Baorui Zhao [1,*], Xiaolei Liu [1], Ruoyu Pan [1], Shengli Pang [1] and Jiwei Song [2,*]

[1] School of Communications and Information Engineering, Xi'an University of Posts and Telecommunications, Xi'an 710121, China; wanghonggang@xupt.edu.cn (H.W.); 230121241@stu.xupt.edu.cn (X.L.); panruoyu@xupt.edu.cn (R.P.); pangshengli@xupt.edu.cn (S.P.)
[2] China Electronics Standardization Institute, Beijing 100007, China
* Correspondence: zbr2320@stu.xupt.edu.cn (B.Z.); songjw@cesi.cn (J.S.)

Abstract: LoRa (long range) is a communication technology that employs chirp spread spectrum modulation. Among various low-power wide area network (LPWAN) technologies, LoRa offers unique advantages, including low power consumption, long transmission distance, strong anti-interference capability, and high network capacity. Addressing the issue of power-constrained end devices in IoT application scenarios, this paper proposes an adaptive data rate (ADR) algorithm for LoRa networks designed for power-constrained end devices (EDs). The algorithm evaluates the uplink communication link state between the EDs and the gateway (GW) by using a combined weighting method to comprehensively assess the signal-to-noise ratio (SNR), received signal strength indication (RSSI), and packet reception rate (PRR), and calculates a list of transmission power and data rates that ensure stable and reliable communication between the EDs and the GW. By using ED power consumption models, network throughput models, and ED latency models to evaluate network performance, the Zebra optimization algorithm is employed to find the optimal data rate for each ED under power-constrained conditions while maximizing network performance. Test results show that, in a single ED scenario, the average PRR achieved by the proposed ADR algorithm for power-constrained EDs in LoRa networks is 14% higher than that of the standard LoRaWAN ADR algorithm. In a multi-ED link scenario (50 end devices), the proposed method reduces the average power consumption of EDs by 10% compared to LoRaWAN ADR, achieves a network throughput of 6683 bps, and an average latency of 2.10 s, demonstrating superior performance overall. The proposed method shows unique advantages in LoRa networks with power-constrained EDs and a large number of EDs, as it not only reduces the average power consumption of the EDs but also optimizes network throughput and average latency.

Keywords: LoRa; ADR; power-constrained; Zebra optimization algorithm; network throughput

MSC: 68T05

check for updates

Citation: Wang, H.; Zhao, B.; Liu, X.; Pan, R.; Pang, S.; Song, J. An Adaptive Data Rate Algorithm for Power-Constrained End Devices in Long Range Networks. *Mathematics* **2024**, *12*, 3371. https://doi.org/10.3390/math12213371

Academic Editors: Ke-Lin Du and Jonathan Blackledge

Received: 20 September 2024
Revised: 11 October 2024
Accepted: 25 October 2024
Published: 28 October 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

LoRa (long range), proposed by Semtech, employs chirp spread spectrum technology. In low-power wide area networks (LPWAN), LoRa offers advantages such as low power consumption, long transmission distance, strong anti-interference capability, and high network capacity [1]. LoRaWAN, built on LoRa technology, is suitable for a variety of remote IoT applications. Since end devices are often deployed in environments that are difficult to maintain and rely on battery power, they need to operate in a low-power state for extended periods to prolong battery life.

The physical layer of LoRa technology employs specific modulation and coding schemes [2,3], allowing devices to select an appropriate data transmission rate based on the scenario. The data rate is adjusted by configuring the spreading factor (SF) and bandwidth

(BW), and signals with different SFs do not interfere with each other. Gateways in a LoRa network can simultaneously demodulate LoRa signals with different SFs and BWs. LoRaWAN (long range wide area network), an open standard maintained by the LoRa Alliance, operates as an MAC layer protocol on top of LoRa and is designed to handle traffic from large numbers of low-power devices. LoRa technology is widely used in applications such as smart homes, precision agriculture [4,5], smart metering, and remote healthcare [6,7], favored for its low cost, long transmission distance, ease of deployment, and excellent scalability. Its chirp spread spectrum modulation technology allows flexible adjustment of the data rate, achieving wide coverage and low power consumption.

In practical operation, the link of a LoRa network is affected by the low power consumption of EDs and environmental factors. The communication link is unstable and time-varying, requiring an evaluation and optimization of link stability. Since Class A EDs in LoRa networks use the ALOHA scheme for transmission, a fixed data rate cannot guarantee optimal network performance. The typical architecture of a LoRaWAN network includes end devices (EDs), gateways (GWs), network server (NS), and application server (AS), as shown in Figure 1. The GW and EDs use a star topology, where the ED transmits data to the GW using LoRa modulation. The GW forwards the data to the NS, which processes the data and sends them to the application server. The GW is responsible only for data forwarding and does not participate in data processing.

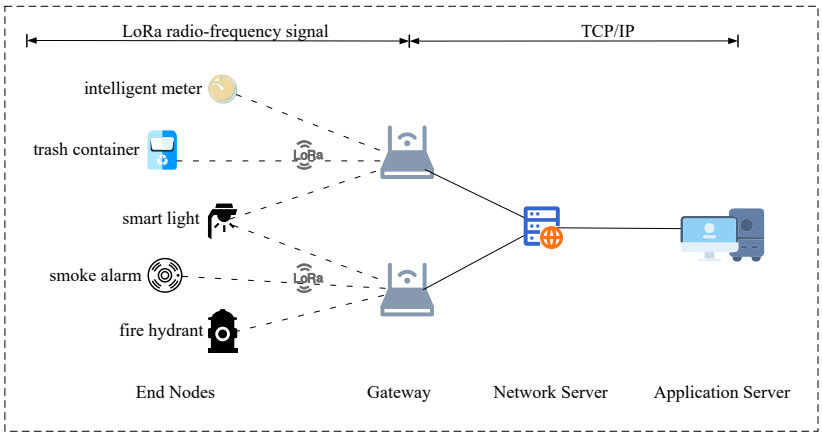


Figure 1. LoRaWAN network architecture.

Most existing adaptive data rate (ADR) algorithms are based on the LoRaWAN architecture, primarily operating on the NS side and typically optimizing only the uplink communication link of EDs or certain aspects of network performance [8]. To address this limitation, this paper first establishes an evaluation model for the stability and reliability of the communication link. Then, based on the characteristics of data transmission from EDs in a LoRa network, models for ED power consumption, network throughput, and ED latency are constructed. Based on these models, an ADR algorithm is proposed that optimizes overall network performance based on the communication link state. Through a two-level optimization—link state and network performance—the algorithm aims to reduce ED power consumption while improving network performance.

The remainder of this paper is organized as follows. Section 2 reviews the current research on ADR algorithms for LoRa networks. Section 3 constructs a link state evaluation model using the weighted combination method based on the characteristics of LoRa communication links. Section 4 determines the performance evaluation methods for LoRa networks by establishing models for network throughput, average power consumption of EDs, and average latency of EDs. Section 5 introduces the algorithm for obtaining selectable parameters for EDs based on link state and the ADR control method based on network performance optimization. Section 6 presents the verification and analysis of the algorithm. Section 7 summarizes the paper and describes future work to be carried out.

2. Related Work

The ADR algorithm in LoRaWAN [9,10] dynamically allocates the transmission rate of EDs based on network conditions to reduce power consumption and enhance network capacity and stability. ADR algorithm consists of two main mechanisms: on the ED side, the data rate is lowered to increase communication range and ensure connectivity with the GW; on the network server side, the data rate and transmission power are adjusted according to the maximum signal-to-noise ratio (SNR) of each packet, thereby reducing the power consumption of the EDs. This algorithm analyzes the link status of the EDs and flexibly adjusts communication parameters to ensure communication robustness and reliability.

In further optimizing the ADR mechanism, M. Slabicki et al. proposed the ADR+ algorithm [11], which uses the average SNR of the most recent 20 packets instead of the maximum SNR value, improving performance under unstable link conditions. This allows the ED to maintain better communication quality even in fluctuating channel conditions. Abdelfadeel proposed the FADR algorithm [12], which allocates the spreading factor and transmission power based on RSSI to achieve fairness between EDs and reduce collisions. However, FADR has limitations in network scalability, as the addition or removal of nodes affects fairness, limiting its application in large-scale networks.

In addition, Davide et al. proposed the I-SFA algorithm [13], focusing on addressing channel variations by adjusting fixed spreading factors. While effective initially, it lacks the ability to adaptively adjust the data rate, resulting in inadequate performance in dynamic environments. In contrast, Riccardo et al. proposed the CA-ADR algorithm [14], which reduces collision probability during data transmission by considering the distribution of all EDs, improving packet success rate and network throughput, particularly in large, dense networks. This algorithm reduces conflicts between nodes by assigning different spreading factors and further optimizes data rates through collision awareness.

Zhou et al. proposed the DRCC algorithm [15], which dynamically adjusts data rates and channel control to support large-scale ED deployments. The algorithm can evaluate channel conditions in real time and reduce collisions through multichannel load balancing, thereby improving network throughput and reducing latency, making it ideal for dense ED scenarios. Sandoval et al. [16] employed a reinforcement learning method to maximize ED throughput. Although highly scalable, the algorithm's complexity is high, and it does not adequately address the power consumption of EDs, limiting its effectiveness in practical applications.

Zhao et al. proposed the EF-LoRa algorithm [17], which enhances energy fairness among EDs (EDs) and extends network lifetime by optimizing spreading factors, transmission power, and channel allocation. The algorithm employs a max–min fairness model and uses a greedy algorithm for resource allocation, demonstrating significant advantages in large-scale network simulations. However, the algorithm struggles with dynamic network changes and may lead to performance fluctuations in large-scale networks. Additionally, the algorithm lacks granularity in ED resource allocation, preventing real-time adaptive adjustments, which could result in delays in resource distribution.

Pang et al. [18] introduced a channel quality rating and prediction method that incorporates the concept of channel quality scores, offering a more precise evaluation of link quality compared to packet reception rate (PRR). This method uses a gated recurrent unit (GRU) algorithm to build a channel quality rating model, classifying EDs based on their current channel quality scores and spatial channel parameters (SCPs). While this approach improves evaluation accuracy, its complexity can lead to prediction biases. Additionally, the method faces challenges in real-time performance and environmental adaptability.

Park et al. [19] proposed the EARN mechanism, an enhanced ADR algorithm that adapts coding rate (CR) based on the capture effect, optimizing data transmission success rate and energy consumption. Although EARN improves energy efficiency, its focus is primarily on uplink communication, lacking system-level optimization for the entire LoRa network. Gabriel et al. [20] introduced the ADRx adaptive data rate algorithm, which dynamically adjusts link margin values in real time to overcome the limitations of

traditional ADR in estimating channel quality, thus improving data extraction rate (DER) and energy efficiency. By dynamically adjusting ED data rates, ADRx optimizes ED power consumption while also enhancing overall network performance.

Jiang et al. [21] applied the ordinary kriging (OK) function to analyze the spatiotemporal correlation of signal-to-noise ratio (SNR) across different time points, proposing the K-ADR algorithm. This algorithm accurately detects changes in link conditions and predicts future link status in real time, improving ADR performance under challenging channel conditions. However, as overall channel conditions deteriorate, the packet delivery rate advantage of K-ADR diminishes, though improvements in ED energy consumption are still observed.

Lodhi et al. [22] proposed the ND-ADR algorithm, which dynamically adjusts link parameters by combining RSSI and SNR values. This algorithm provides a more accurate assessment of communication quality and dynamically adjusts network parameters, improving communication reliability in harsh environments and reducing power consumption.

In terms of physical layer optimization, Xia et al. proposed the XCopy algorithm [23], which improves SNR by repeatedly retransmitting weak signals and coherently accumulating them, providing more than 10 dB of gain in low-SNR environments. This makes communication more reliable, particularly for EDs in harsh conditions. Sun et al. [24] introduced the FLoRa algorithm, which improves the energy efficiency and reliability of firmware updates over-the-air (FUOTA) in LoRa networks through incremental updates, channel coding, and beamforming techniques. This effectively addresses the low energy efficiency and unreliable transmission issues in traditional FUOTA. Hou et al. [25] proposed the one-shot method, which aggregates data from multiple nodes, significantly improving transmission accuracy and efficiency, especially in large-scale LoRa networks.

Although these physical layer optimization algorithms perform well in improving communication reliability and energy efficiency, they also increase the hardware cost for EDs. In contrast, ADR algorithms flexibly adjust ED parameters, leveraging LoRa's features to optimize link connectivity and network performance. Table 1 compares the different algorithms, listing their optimization approaches and objectives.

Table 1. Comparison of adaptive data rate algorithms.

Ref.	Proposed Solution	Optimization Approach	Link Reliability	Link Stability	Power Consumption	Network Throughput	ED Latency
[9,10]	LoRaWAN ADR	Parameter Resource Optimization	✓		✓		
[13]	I-SFA	Parameter Resource Optimization	✓			✓	
[14]	CA-ADR	Parameter Resource Optimization	✓			✓	✓
[17]	EF-LoRa	Parameter Resource Optimization	✓		✓		
[21]	K-ADR	Parameter Resource Optimization	✓		✓		
[23]	XCopy	CommunicationProcess Optimization	✓	✓	✓	✓	
	Proposed Algorithm	Parameter Resource Optimization	✓	✓	✓	✓	✓

3. Link State Evaluation

3.1. Characteristics of LoRa Communication Link

A LoRa communication link refers to the connection between the ED and the GW. Due to the complex environment in which EDs often operate and their power consumption limitations, the link state is difficult to predict. In wireless communication, signal fading is divided into large-scale fading and small-scale fading. Large-scale fading is mainly caused by path loss, which is proportional to the propagation distance; the greater the distance, the higher the loss. Small-scale fading occurs due to multipath interference, where the signal rapidly fades over short periods or short distances, making large-scale fading negligible [26].

When the received signal experiences line-of-sight (LOS) propagation, the direct signal is the main component, and the multipath signals are superimposed on it. The envelope of

the received signal follows a Rician distribution. When there is no direct signal, i.e., non-line-of-sight (NLOS) propagation, the signal reaches the receiver through reflection, refraction, diffraction, and other paths, and the envelope follows a Rayleigh distribution [27,28].

3.2. Link State Parameters

In a LoRa network, ED collects sensor data and transmits them to the GW via LoRa modulation. The GW uploads the data packets to the NS, where the server gathers link quality information, including RSSI, SNR, and PRR. RSSI and SNR are physical layer parameters that can instantly reflect link conditions but are susceptible to fluctuations [29]. PRR, as a soft metric at the data link layer, more accurately reflects link quality but requires a longer time to be statistically meaningful and cannot be obtained directly [30]. In a LoRa network, link state evaluation should be accurate and comprehensive, and using PRR alone is insufficient to fully capture the link condition [31]. The link state is dynamic, with short-term fluctuations reflected by physical layer parameters and long-term trends indicated by PRR.

Assuming the statistical window size for packet reception is n, the SNR and RSSI of the received packets within the window are recorded, and the PRR is calculated. Let SNR_{avg_i} and $RSSI_{avg_i}$ represent the average SNR and average RSSI within the window, respectively, and let CV_{SNR_i} and CV_{RSSI_i} represent the coefficients of variation for SNR and RSSI within the window, respectively. The values SNR_{avg_i}, $RSSI_{avg_i}$ and PRR measure the reliability of the link, while CV_{SNR_i} and CV_{RSSI_i} measure the stability of the link. The specific calculation formulas are shown in Equations (1)–(4).

$$SNR_{avg_i} = \frac{\sum_{j=1}^{n} SNR_j}{n} \tag{1}$$

$$RSSI_{avg_i} = \frac{\sum_{j=1}^{n} RSSI_j}{n} \tag{2}$$

$$CV_{SNR_i} = \frac{\sqrt{\frac{1}{n}\sum_{j=1}^{n}\left(SNR_j - SNR_{avg_i}\right)^2}}{SNR_{avg_i}} \tag{3}$$

$$CV_{RSSI_i} = \frac{\sqrt{\frac{1}{n}\sum_{j=1}^{n}\left(RSSI_j - RSSI_{avg_i}\right)^2}}{RSSI_{avg_i}} \tag{4}$$

The evaluation of link state includes measuring various parameters that reflect the link state and calculating the combined weights of these parameters. The determination of weights is achieved using a combined weighting method. Subjective weights are calculated using the fuzzy analytic hierarchy process (FAHP), while objective weights are determined through the improved CRITIC method and further optimized using the VIKOR method. Finally, the subjective weights and the optimized objective weights are combined using the multiplicative comprehensive method to obtain the final combined weight. The calculation of the link state score is shown in Equation (5).

$$score = (\omega_1, \omega_2, \omega_3, \omega_4, \omega_5) \begin{pmatrix} \overline{\frac{SNR_{avg_i}}{RSSI_{avg_i}}} \\ \overline{PRR} \\ \overline{\frac{CV_{SNR_i}}{CV_{RSSI_i}}} \end{pmatrix} \tag{5}$$

Among them, $\left(\overline{SNR_{avg_i}}, \overline{RSSI_{avg_i}}, \overline{PRR}, \overline{CV_{SNR_i}}, \overline{CV_{RSSI_i}}\right)^T$ represents the normalized values of various link state parameters, while $(\omega_1, \omega_2, \omega_3, \omega_4, \omega_5)$ represents the combined weights corresponding to each parameter. The link state score ranges from 0 to 1.

3.3. Combined Weighting Theory

Combined weighting is used in multiattribute decision analysis to handle problems involving multiple evaluation criteria [32]. It determines the weight of each criterion by combining subjective and objective weighting methods to more accurately and comprehensively reflect the importance of each criterion. Since each individual method has its own advantages and disadvantages, using combined weighting can improve the rationality and credibility of decision making.

3.3.1. Fuzzy Analytic Hierarchy Process

The fuzzy analytic hierarchy process (FAHP) combines fuzzy mathematics and the analytic hierarchy process (AHP) to solve complex decision-making problems [33,34]. In FAHP, criteria and alternatives are represented using triangular fuzzy numbers. The specific representation of fuzzy sets is given by Equation (6):

$$A = (x, \mu_A(x)) | x \in X \tag{6}$$

Among them, A is the triangular fuzzy number, X is the domain of the input variable x, and $\mu_A(x)$ is the membership degree of x. The membership function can be expressed as

$$\mu_A(x) = \begin{cases} \frac{x-a}{b-a}, & a \le x \le b \\ \frac{c-x}{c-b}, & b \le x \le c \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

Equation (7), a, b, and c are the left, middle, and right vertices of the upper triangle, respectively. In a triangular fuzzy number, b represents the maximum degree of truth, while a and c reflect the experts' reservations. Assuming that the fuzzy numbers M_1 and M_2 are determined by the three-dimensional data (a_1, b_1, c_1) and (a_2, b_2, c_2), the basic operations for triangular fuzzy numbers are as per Equation (8).

$$\begin{cases} M_1 \oplus M_2 = (a_1, b_1, c_1) \oplus (a_2, b_2, c_2) = (a_1 + a_2, b_1 + b_2, c_1 + c_2) \\ M_1 \otimes M_2 = (a_1, b_1, c_1) \otimes (a_2, b_2, c_2) = (a_1 a_2, b_1 b_2, c_1 c_2) \\ (a_1, b_1, c_1)^{-1} = \left(\frac{1}{a_1}, \frac{1}{b_1}, \frac{1}{c_1}\right) \\ M_1^{M_2} \cong \left(a_1^{a_2}, b_1^{b_2}, c_1^{c_2}\right) \end{cases} \tag{8}$$

Equation (9) represents the geometric mean of the corresponding row parameters in the crisp matrix GM_i.

$$GM_i = \left(\prod_{j=1}^{M} b_{ij}\right)^{\frac{1}{M}} \tag{9}$$

Equation (9), b_{ij} represents the value in the crisp comparison matrix at the i-th row and j-th column, and M represents the number of parameters in the comparison matrix. The weight of the variable ω_i can be expressed using Equation (10).

$$\omega_i = \frac{GM_i}{\sum_{j=1}^{M} GM_j} \tag{10}$$

In the fuzzy analytic hierarchy process, the relative importance scale ranges from $\overline{1}$ to $\overline{9}$. Table 2 presents the fuzzy relative importance scale, where α represents the fuzzy factor.

Table 2. Definition of the fuzzy relative importance scale.

Scale	Fuzzy Score	Meaning
1	(1,1,1)	Indicates equal importance between two factors.
3	$(3-\alpha, 3, 3+\alpha)$	The former factor is slightly more important than the latter.
5	$(5-\alpha, 5, 5+\alpha)$	The former is significantly more important than the latter.
7	$(7-\alpha, 7, 7+\alpha)$	The former is strongly more important than the latter.
9	$(9-\alpha, 9, 9+\alpha)$	The former is extremely more important than the latter.
2, 4, 6, 8	$(x-\alpha, x, x+\alpha)$	Intermediate values between two adjacent judgments.

3.3.2. Improved CRITIC Method

The CRITIC method is used for objective weight assignment in multiattribute decision making, helping to avoid uncertainties caused by subjective weighting [35]. The traditional CRITIC method reflects data variability through standard deviation but can only capture linear relationships, potentially leading to weighting errors. The improved D-CRITIC method [36] introduces distance correlation, enabling a more accurate measurement of nonlinear relationships between criteria. This results in more stable weight generation, enhancing the accuracy and reliability of decision analysis, especially for complex or large-scale decision matrices.

Step 1: Data standardization. Let the original decision matrix be $X = (x_{ij})_{m\times n}$. By standardizing the data, the differences in units or magnitudes between the indicators are eliminated, resulting in the standardized matrix $X' = \left(x'_{ij}\right)_{m\times n}$. The standardization formula is given as Equation (11).

$$x'_{ij} = \frac{x_{ij} - x_j^{\min}}{x_j^{\max} - x_j^{\min}} \tag{11}$$

where $x_j^{\min}$ and $x_j^{\max}$ represent the minimum and maximum values of the j-th indicator, respectively.

Step 2: Calculate the standard deviation s_j for each indicator. The formula is given as Equation (12).

$$s_j = \sqrt{\frac{1}{m-1}\sum_{i=1}^{m}\left(x'_{ij} - \overline{x'_j}\right)^2} \tag{12}$$

In Equation (12), $\overline{x'_j}$ represents the mean value of the j-th indicator, and m is the number of samples.

Step 3: Calculate the distance correlation coefficient between indicators. Distance correlation (instead of Pearson correlation r_{jk}) is used, and $dCor(c_j, c_k)$; the calculation formula is given as Equation (13).

$$dCor(c_j, c_k) = \frac{dCov(c_j, c_k)}{\sqrt{dVar(c_j)\cdot dVar(c_k)}} \tag{13}$$

In Equation (13), $dCov(c_j, c_k)$ represents the distance covariance between c_j and c_k, and $dVar(c_j)$ and $dVar(c_k)$ represent the distance variances of c_j and c_k.

Step 4: Calculate the information content for each indicator. The information content I_j depends not only on the standard deviation s_j but also on the distance correlation coefficient $dCov(c_j, c_k)$. The information content can be calculated using the following formula, as shown in Equation (14).

$$I_j = s_j \cdot \sum_{k\neq j}(1 - dCov(c_j, c_k)) \tag{14}$$

Step 5: Determine the target weights. Based on the information content I_j, calculate the objective weight ω_j for each evaluation indicator. The formula is given as Equation (15).

$$\omega_j = \frac{I_j}{\sum_{q=1}^{n} I_q} \tag{15}$$

where I_j represents the information content of the j indicator.

3.3.3. VIKOR Method

The VIKOR method is a multiattribute decision-making approach that evaluates alternatives by calculating their overall performance (S value), worst performance (R value), and overall index value (Q value) to optimize objective weights [37]. The specific steps are as follows:

Step 1: Data standardization. Based on the data standardization in the improved CRITIC method, the standardized matrix $X' = \left(x'_{ij}\right)_{m\times n}$ is obtained.

Step 2: Calculate the S and R values. The comprehensive performance value S_i and the worst performance value R_i are calculated using the following formulas in Equations (16) and (17):

$$\begin{aligned} S_i &= \sum_{j=1}^{n} \omega_j \frac{(r_j^+ - r'_{ij})}{(r_j^+ - r_j^-)} \\ R_j &= \max_{1\le j\le n}\left(\omega_j \frac{(r_j^+ - r'_{ij})}{(r_j^+ - r_j^-)}\right) \end{aligned} \tag{16}$$

$$R_j = \max_{1\le j\le n}\left(\omega_j \frac{(r_j^+ - r'_{ij})}{(r_j^+ - r_j^-)}\right) \tag{17}$$

where ω_j represents the objective weight calculated using the improved CRITIC method; r'_{ij} represents the standardized value of the i-th alternative on the j-th indicator; and r_j^+ and r_j^- represent the positive ideal solution and negative ideal solution for the j-th indicator, respectively.

Step 3: Calculate the Q value. The Q value combines the comprehensive performance and the worst performance of each alternative. The calculation formula is as follows:

$$Q_i = v\left(\frac{S_i - S^-}{S^+ - S^-}\right) + (1-v)\left(\frac{R_i - R^-}{R^+ - R^-}\right) \tag{18}$$

In Equation (18), $S^+ = \max_{1\le i\le m} S_i$, $S^- = \min_{1\le i\le m} S_i$, $R^+ = \max_{1\le i\le m} R_i$, $R^- = \min_{1\le i\le m} R_i$; v is the decision-making coefficient, and $v \in [0,1]$.

Step 4: Optimize objective weights. Update the objective weights ω'_j based on the Q value. The calculation formula is given as Equation (19).

$$\omega'_j = 1 - \frac{Q_i}{\sum_{i=1}^{m}(1 - Q_i)} \tag{19}$$

Finally, the combined weight of each indicator is calculated using the multiplicative comprehensive method. The subjective weight ω_i is derived from the fuzzy analytic hierarchy process, and the objective weight ω'_j is obtained through the improved CRITIC method and the VIKOR method. The combined weight ω_k of the k-th indicator can be determined by the following formula, as shown in Equation (20).

$$\omega_k = \frac{\omega_{ik} \cdot \omega_{jk}}{\sum_{k=1}^{n} \left(\omega_{ik} \cdot \omega_{jk} \right)} \tag{20}$$

3.4. Link State Evaluation Model

The fuzzy analytic hierarchy process (FAHP) constructs the decision structure through different levels. The first level is the objective, and the third level consists of the criteria and indicators, as shown in Figure 2.

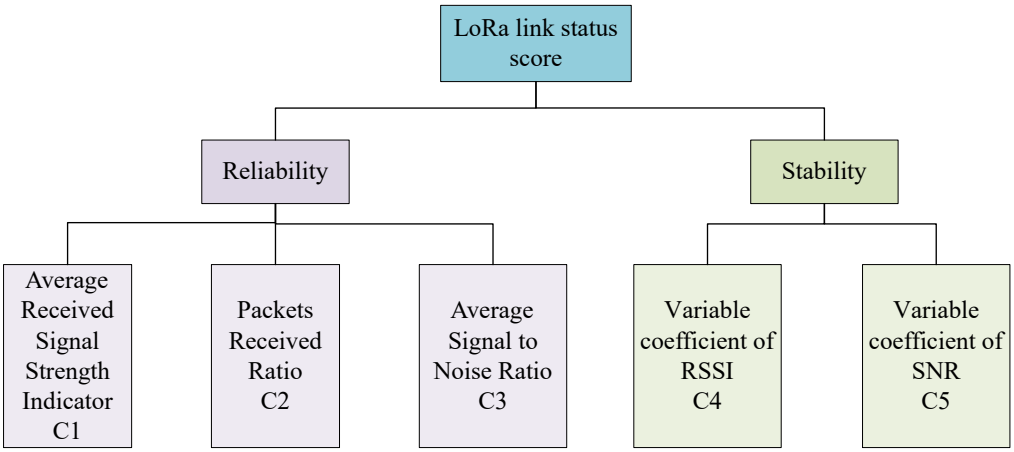


Figure 2. Link state scoring FAHP evaluation model.

The weight of each parameter is calculated based on the fuzzy numbers corresponding to the score of each indicator, and a sensitivity analysis is conducted using six sets of fuzzy factor values (0, 0.2, 0.4, 0.6, 0.8, 1) (see Table 3). The results show that although the weights vary with changes in the fuzzy factor, the ranking of the parameters remains unchanged, indicating that the weight calculation is reasonable.

The subjective weights are calculated using the judgment matrix and FAHP algorithm, while the objective weights are calculated using the improved CRITIC method. The objective weights are further optimized through the VIKOR method, resulting in the final combined weights, as shown in Table 4. Based on this, the link state score can be calculated using the five parameters.

Table 3. Sensitivity analysis of the decision model.

Category	$\alpha = 0$	$\alpha = 0.2$	$\alpha = 0.4$	$\alpha = 0.6$	$\alpha = 0.8$	$\alpha = 1$
C1	0.1325	0.1332	0.1337	0.1337	0.1335	0.1334
C2	0.2432	0.2417	0.2428	0.2426	0.2425	0.2423
C3	0.3215	0.3224	0.3220	0.3218	0.3216	0.3214
C4	0.1987	0.1998	0.1992	0.1993	0.1995	0.1997
C5	0.1025	0.1036	0.1023	0.1027	0.1030	0.1032

Table 4. Weight results for each parameter.

Category	C1	C2	C3	C4	C5
FAHP Weights	0.1337	0.2428	0.3220	0.1992	0.1023
Improved CRITIC Method Weights	0.3304	0.2881	0.1885	0.0549	0.1381
VIKOR Method Optimized Weights	0.3173	0.2793	0.1897	0.0694	0.1443
Combined Weights	0.1981	0.3166	0.2851	0.0968	0.1034

4. LoRa Network Performance Evaluation

4.1. Network Throughput Model

In a LoRa network, EDs use the pure ALOHA protocol for random access packet transmission. Assume that nodes generate packets according to a Poisson process, with an average generation rate of λ packets per second, and the packet length is L. EDs in the network use m different data rates for packet transmission, with N_i EDs using each rate R_{b_i}. Let G_i represent the number of packets generated by EDs transmitting at the rate R_{b_i}. Under the pure ALOHA protocol, the probability of a LoRa ED successfully transmitting a packet without collision at rate R_{b_i} is given by Equation (21).

$$P_{success} = e^{-2G_i} \tag{21}$$

The time required for ED i to transmit a packet T_i is given by Equation (22).

$$T_i = \frac{L}{R_{b_i}} \tag{22}$$

The data rate R_{b_i} can be calculated using Equation (23).

$$R_{b_i} = SF \cdot \frac{BW}{2^{SF}} \cdot \frac{4}{4 + CR} \tag{23}$$

where SF represents the SF used by the ED, BW is the modulation BW during the packet transmission, and CR represents the coding rate, with values ranging from 1 to 4.

The total traffic generated by all EDs transmitting at the rate R_{b_i} per unit time is given by Equation (24).

$$\lambda_i = \frac{N_i}{T_{\text{interval}}} \tag{24}$$

The traffic generated by EDs transmitting data packets at a rate R_{b_i} is $G_i = \lambda_i T_i$, and the collision probability is given by Equation (25).

$$P_{collision_i} = 1 - e^{-2\lambda_i T_i} = 1 - e^{-2 \cdot \frac{\lambda_i \cdot L}{R_{b_i}}} \tag{25}$$

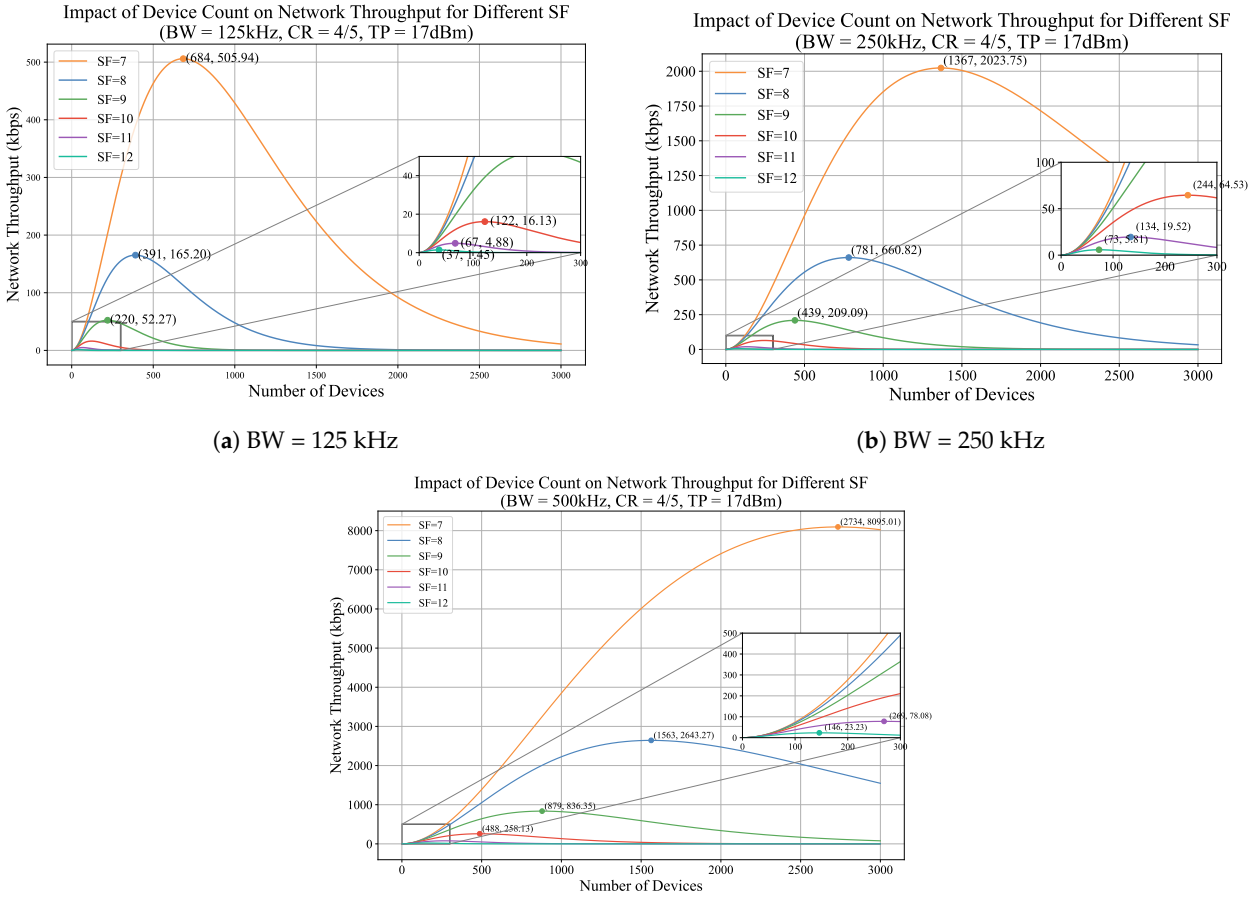
The average throughput of the ED is given by Equation (26).

$$\Gamma_i = \lambda \cdot L \cdot P_{success} = \lambda \cdot L \cdot e^{-2 \cdot \frac{\lambda_i \cdot L}{R_{b_i}}} \tag{26}$$

When a total of m different data rates are used for transmission in the network, and there are N_i EDs using data rate R_{b_i}, the total network throughput Γ is given by Equation (27).

$$\Gamma = \sum_{i=1}^{m} \Gamma_i \cdot N_i \tag{27}$$

When only one data rate is used in the LoRa network, the impact of the number of EDs on network throughput is investigated. From Figure 3a–c, it can be seen that network throughput first increases and then decreases as the number of EDs increases. Different SFs significantly affect the peak throughput and the number of devices. Increasing the SF reduces network capacity and throughput. Smaller SFs (e.g., SF = 7) and larger BWs (e.g., 250 kHz) significantly improve network throughput, while larger SFs (e.g., SF = 12) and smaller BWs reduce data rates and throughput. Reasonably adjusting the SF and BW can effectively enhance the performance and capacity of the LoRa network.



(**a**) BW = 125 kHz

(**b**) BW = 250 kHz

(**c**) BW = 500 kHz

Figure 3. Network throughput variation with different SFs when CR = 4/5 and TP = 17 dBm.

4.2. ED Power Consumption Model

4.2.1. ED Hardware Design

The LoRa ED is a fundamental component of the LoRa network, responsible for collecting sensor data and uploading them to the NS. Its hardware design must meet the requirements for multiple sensor interfaces, power supply options, and low power consumption. The ED uses the STM32F103RCT6 microcontroller (MCU) (manufactured by STMicroelectronics, Geneva, Switzerland) as the main controller and Semtech's SX1262 RF chip (manufactured by Semtech Corporation, Camarillo, CA, USA) as the communication module. The device is compatible with UART and RS-485 interfaces, making it suitable for a variety of sensors. A physical diagram of the ED hardware is shown in Figure 4.



Figure 4. Physical diagram of ED hardware.

4.2.2. ED Operating State Classification

After the ED is powered on for the first time, it enters the initialization phase, completing ED initialization, sending a network join request, receiving a join response, collecting data, sending data, receiving an ACK, and then entering sleep mode. Class A EDs use the pure ALOHA protocol, switching from sleep mode to active mode only when data need to be uploaded. The state classification in normal operation mode is shown in Figure 5. After the ED sends data to the GW, the GW will send an acknowledgment frame (ACK) within two receive windows. The ED should open the first receive window (RX1) within a specified time *RECEIVE_DELAY1* after sending the data, using the same communication parameters as when sending the data packet. If no data are received in the first window, the ED should open the second receive window (RX2) within a specified time *RECEIVE_DELAY2*, using the slowest data rate to receive the acknowledgment frame. If the data frame is not successfully transmitted due to a collision, the ED will not receive an ACK within the two receive windows and will then enter sleep mode. After a while, the ED will wake up again and repeat the normal operating state. A diagram of the ED's failed data transmission state is shown in Figure 6.

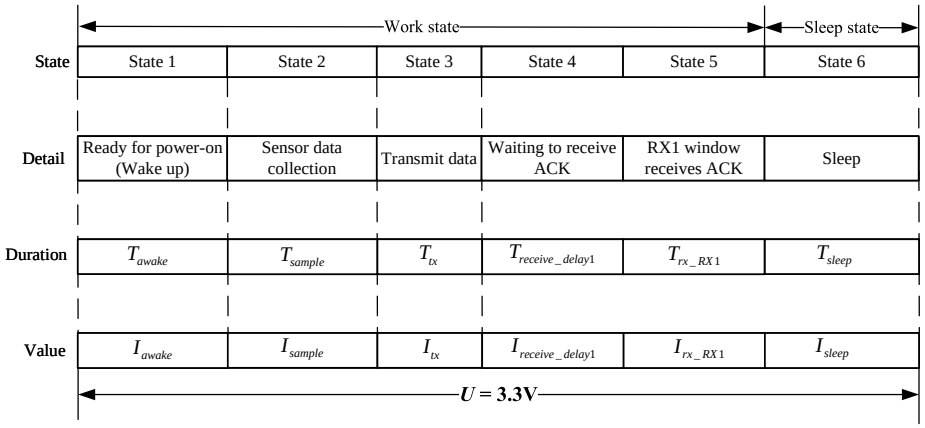


Figure 5. ED normal operating state classification.

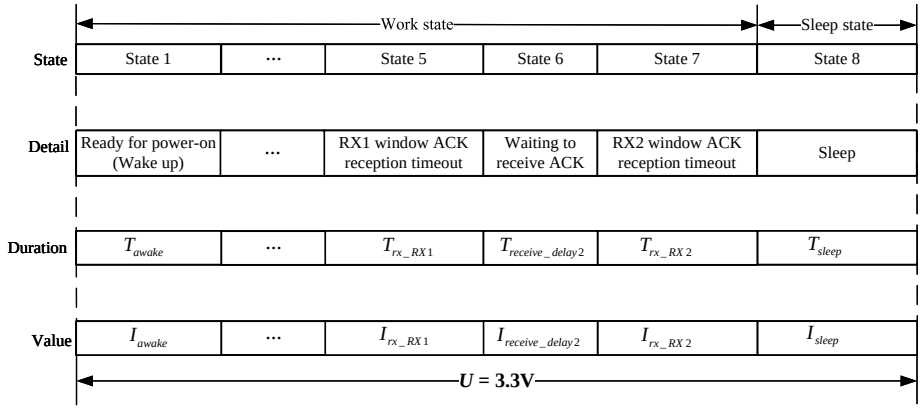


Figure 6. ED failed data frame transmission state classification.

4.2.3. Analysis of ED Power Consumption in Different Operating States

When a LoRa ED sends a data frame, it switches between various operating states until the transmission is successful. The energy consumption of the ED can be evaluated by calculating the average power consumption for successfully transmitting a data frame. The total power consumption E_{total} for successfully transmitting one frame equals the sum of the sleep power consumption E_{sleep} and the active power consumption E_{work}. However, in

actual testing, E_{sleep} is found to be negligible compared to E_{work}. Therefore, the total power consumption for successfully transmitting one frame E_{total} is given by Equation (28).

$$E_{total} = E_{sleep} + E_{work} \approx E_{work} \tag{28}$$

In the pure ALOHA model, a LoRa ED may encounter two scenarios: collision or successful transmission. Assuming the probability of successful transmission without collision is p, which follows a binomial distribution, successfully transmitting one data frame requires *k-1* failures and *1* success. Thus, the average power consumption for successfully transmitting one data frame is given by Equation (29).

$$E_{work} = \sum_{k=1}^{\infty} (1-p)^{k-1} p \cdot \left[(k-1)E_{fail} + E_{success} \right] = \frac{1-p}{p} \cdot E_{fail} + E_{success} \tag{29}$$

In Equation (29), the energy consumption for successful transmission $E_{success}$ is equal to the energy consumed by the ED when sending the data frame and successfully receiving the ACK. The energy consumption for a failed transmission due to collision E_{fail} is equal to the energy consumed by the ED when waking up, collecting data, sending the data frame, and experiencing ACK timeout in both receive windows, as shown in Equations (30) and (31).

$$E_{success} = U\left[I_{awake}T_{awake} + I_{sample}T_{sample} + I_{tx}T_{tx} + I_{receive_delay1}T_{receive_delay1} + I_{rx_RX1}T_{rx_RX1} \right] \tag{30}$$

$$\begin{aligned} E_{fail} = U\Big[& I_{awake}T_{awake} + I_{sample}T_{sample} + I_{tx}T_{tx} + I_{receive_delay1}T_{receive_delay1} + I_{rx_RX1}T_{rx_RX1} \\ & + I_{receive_delay2}T_{receive_delay2} + I_{rx_RX2}T_{rx_RX2} \Big] \end{aligned} \tag{31}$$

The power consumption of the ED when in the data transmission state is $E_{TX} = U \cdot I_{TX} \cdot T_{TX}$. The time required for a LoRa signal to travel from the transmitter to the receiver is defined as the time on air (ToA), that is, the time $T_{TX} = ToA$ during which the ED sends the data frame. ToA is calculated as in Equation (32).

$$T_{\mathrm{ToA}} = \frac{1}{R_{\mathrm{s}}} \times N_{\mathrm{symbol}}(s) \tag{32}$$

In the formula, R_s is the symbol rate, which can be calculated as $R_s = BW/2^{SF}$ $(symbols/s)$, where N_{symbol} represents the number of symbols in the data frame transmitted by the ED. A LoRa data frame [2] consists of a preamble and a payload, i.e., $N_{symbol} = N_{preamble} + N_{payload}$.

Assuming that in the LoRa network, an ED using data rate R_{b_i} and transmission power TP_i successfully sends a total of N_i data packets, the average power consumption for the ED to successfully send one data frame is given by Equation (33).

$$E_{avg_i} = \frac{1}{N_i} \sum_{k=1}^{N_i} E_{total_ik} \tag{33}$$

Equation (33), E_{total_ik} represents the power consumption of the ED when transmitting the k data frame using data rate R_{b_i} and transmission power TP_i, N_i represents the number of successfully transmitted data packets.

Assuming there are n EDs in the LoRa network, and a total of m different data rates and transmission power combinations (R_{b_i}, TP_i) are used, with n_i EDs for each combination, the average power consumption E_{avg} of the EDs in the entire LoRa network can be calculated using a weighted average as per Equation (34).

$$E_{avg} = \sum_{i=1}^{m} \frac{n_i}{n} \cdot E_{avg_i} \tag{34}$$

For a specific LoRa ED, the power consumption is mainly related to the working current and duration in each operating state. Using a LoRa ED equipped with an SX1262 chip and connected to an inclination sensor, the current and duration in each operating state are recorded, as shown in Table 5.

Table 5. Working current and duration of LoRa ED with SX1262 chip.

Type	Value
I_{awake}	150.69 mA
I_{sample}	182.55 mA
$I_{receive_delay1} \sim I_{receive_delay2}$	108.82 mA
$I_{rx_RX1} \sim I_{rx_RX2}$	119.72 mA
I_{sleep}	0.01 mA
T_{awake}	0.73 s
T_{sample}	1.19 s
$T_{receive_delay1} \sim T_{receive_delay2}$	1.00 s
$T_{rx_RX1} \sim T_{rx_RX2}$	1.50 s

The common transmission power range of LoRa EDs equipped with the SX1262 chip is from 2 dBm to 20 dBm. The working current in the LoRa transmission state varies with different transmission power levels. To determine the relationship between the two, the working current of the ED in the transmission state at different transmission power levels was measured, as shown in Table 6.

Table 6. Working current of ED in transmission state at different transmission power levels.

Transmission Power	Working Current
2 dBm	143.85 mA
5 dBm	153.51 mA
8 dBm	167.72 mA
11 dBm	183.39 mA
14 dBm	205.22 mA
17 dBm	222.89 mA
20 dBm	236.41 mA

Figure 7 shows the impact of different data rates and transmission power levels on the transmission power consumption of the ED. As shown in the figure, BW has a relatively large effect on the power consumption of the ED. When the SF, transmission power, and coding rate are fixed, EDs using larger BWs consume less energy to successfully send one data frame. Therefore, when the coding rate is fixed, optimizing the transmission power consumption of EDs requires paying attention to the allocation of transmission power and BW.

4.3. LoRa Network ED Latency Analysis

The latency of an ED can be defined as the time interval between the generation of a packet by the ED and the reception of the packet at the NS [14]. In a LoRaWAN network, compared to the LoRa communication between the ED and the GW, the connection between the GW and the NS via Ethernet means that T_{GW-NS} and T_{NS}^{proc} can be ignored. The ED latency mainly includes the time to generate a packet T_{ED}^{proc} and the time on air ToA_{UL}.

Assuming that the process of generating data packets in the network follows a Poisson process with an average generation rate of λ packets per second, and the packet length is L, the EDs in the LoRa network use m different data rates to transmit data packets. There are N_i EDs for each data rate R_{b_i}, and G_i represents the number of packets generated by the EDs transmitting at rate R_{b_i}. Under the pure ALOHA protocol, the probability $P_{success}$ of successful transmission without collision by EDs using rate R_{b_i} is calculated

using Equation (21). Assuming that failed transmissions can be retransmitted infinitely, the number of failures *k-1* plus *1* success is required to successfully transmit a data frame. For EDs using data rate R_{b_i}, the average latency T_{avg_i} can be calculated using Equation (35).

$$T_{avg_i} = \sum_{k=1}^{\infty} (1 - p_{success})^{k-1} p \left[(k-1) T_{fail} + T_{success} \right] = \frac{1 - p_{success}}{p_{success}} T_{fail} + T_{success} \quad (35)$$

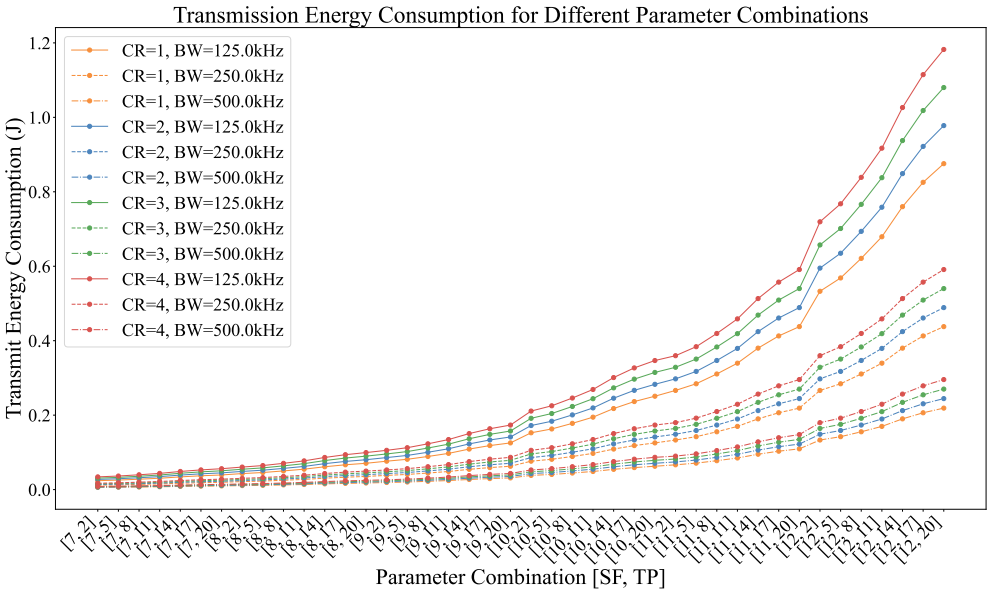


Figure 7. Comparison of ED transmission power consumption at different data rates and transmission power levels.

In Equation (35), the latency for a successful transmission $T_{success}$ is given as Equation (36).

$$T_{success} = T_{ED}^{\text{proc}} + \text{ToA}_{UL} \quad (36)$$

The latency for a failed transmission T_{fail} includes the time from when the ED generates a packet to when it waits for an ACK timeout, as well as the time interval T_{interval} before the next retransmission. Thus, the latency for a failed transmission T_{fail} is given by Equation (37).

$$T_{fail} = T_{ED}^{\text{proc}} + \text{ToA}_{UL} + 2 \cdot T_{wait} + 2 \cdot T_{RX} + T_{\text{interval}} \quad (37)$$

where T_{wait} is the time when the ED opens the receive window after sending the data packet; T_{RX} is the duration of the receive window (assuming the duration of both receive windows is the same); and T_{interval} is the time interval before the ED retransmits the data packet after a failed transmission. Since the process of generating data packets by the ED follows a Poisson process with an average generation rate of ρ packets per second, the time interval between two packet transmissions follows an exponential distribution with parameter ρ. Therefore, the average time interval between two packets is $T_{\text{interval}} = \frac{1}{\rho}$.

Assuming there are n EDs in the LoRa network, and a total of m different data rates are used, with n_i EDs for each data rate, the average latency T_{avg} of the EDs in the entire LoRa network can be calculated using a weighted average, as shown in Equation (38):

$$T_{avg} = \sum_{i=1}^{m} \frac{n_i}{n} \cdot T_{avg_i} \quad (38)$$

Figure 8 shows the relationship between the number of EDs and the average latency at different data rates. As the SF increases (resulting in a lower data rate), the time required for the ED to send a data packet increases, leading to a longer average latency. Additionally,

as the number of EDs using lower data rates increases, the probability of packet collisions rises, further increasing the average latency. Therefore, to optimize the average latency, it is necessary to reasonably control the number of EDs using the same data rate.

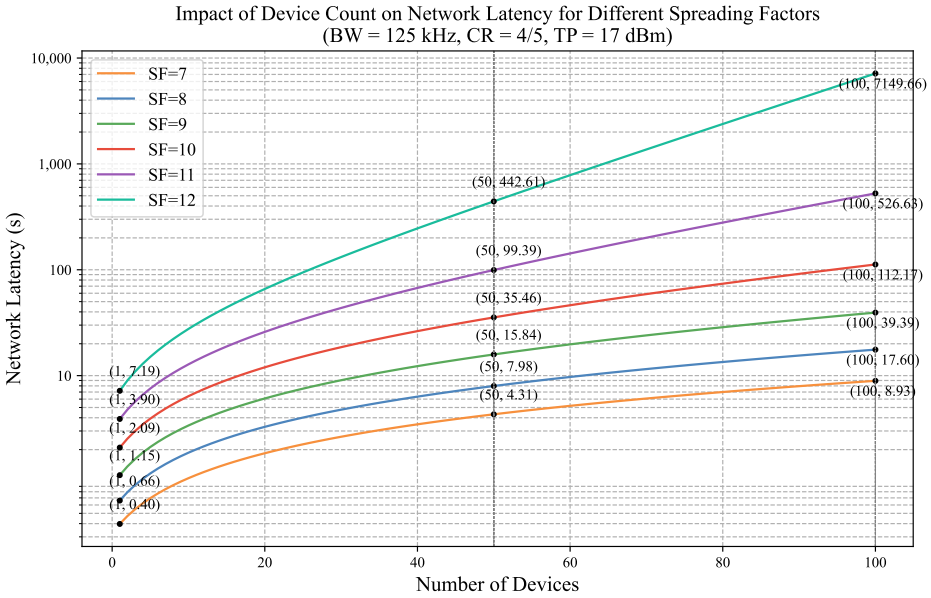


Figure 8. Relationship between the number of EDs and average latency at different SFs.

4.4. LoRa Network Performance Evaluation Model

The calculation of the network performance evaluation score is shown in Equation (39), where $(\overline{\Gamma}, \overline{E_{avg}}, \overline{T_{avg}})$ represents the normalized values of the three performance indicators, and (v_1, v_2, v_3) represents the weight of each parameter. The LoRa network performance score is calculated based on the weights and normalized parameters, with a value range of $[0, 100]$. The evaluation indicators include total network throughput Γ, average power consumption of the EDs E_{avg}, and average latency of the EDs T_{avg}. These indicators are affected by the number of EDs using different data rates in the network. The weight of each network performance parameter is obtained using the combined weighting method introduced in Section 3.4. The specific calculation formula is given as Equation (39).

$$S = (c_1, c_2, c_3) \cdot \left(\frac{\overline{\Gamma}}{\overline{E_{Avg}}} \right) \times 100 \tag{39}$$

5. Adaptive Data Rate Algorithm Design

LoRaWAN leverages the advantages of the LoRa physical layer by optimizing the communication performance of EDs through adjusting data rates (i.e., LoRaWAN ADR). In actual network deployments, adjusting the data rate requires balancing communication quality and network performance. When the environment where the ED operates is subject to interference, communication quality should be prioritized. Additionally, since LoRa networks use the pure ALOHA protocol, data rate allocation must also consider network performance optimization. A new ADR algorithm based on link state and network performance optimization is proposed. Based on the link state, the transmission power and a list of selectable data rates are first determined, followed by the optimal data rate allocation in combination with network optimization. Figure 9 shows the overall flowchart for designing the adaptive data rate algorithm.

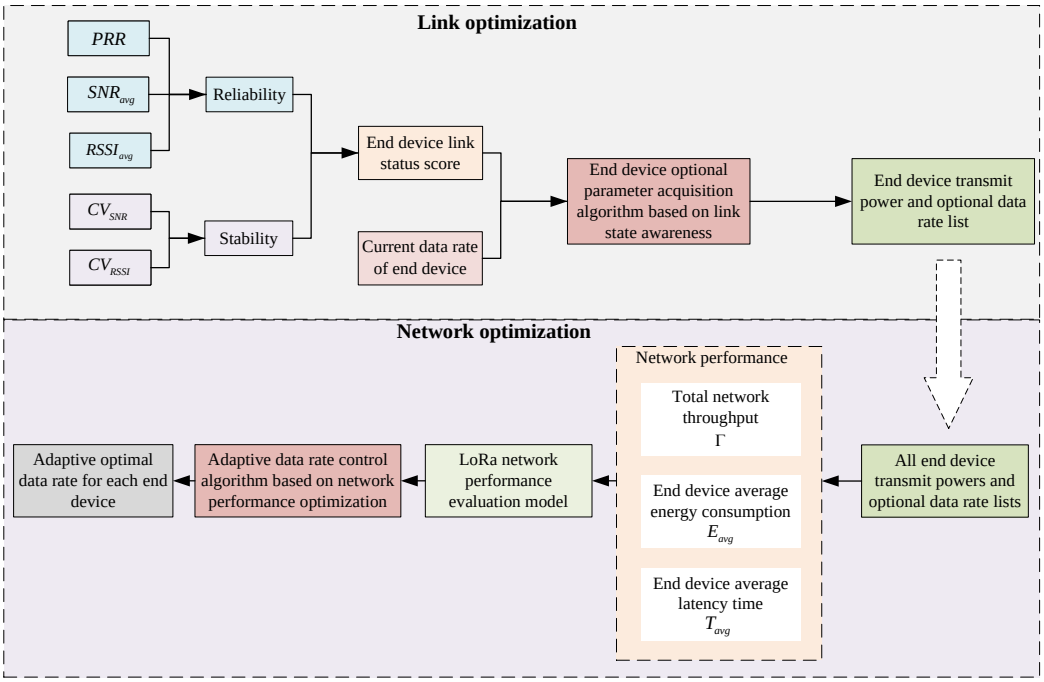


Figure 9. Overall flowchart of the ADR algorithm.

5.1. ED Optional Parameter Acquisition Algorithm Based on Link State

The ED optional parameter acquisition algorithm based on the link state determines the selectable data rate for the current ED based on the link state score. Every time the ED sends 20 data packets, it is considered a statistical window, during which the values of SNR, RSSI, and PRR are recorded. The link state score is calculated according to the link state evaluation method described in Section 3.

The inputs to the algorithm include the ED's current data rate, transmission power combination, and link state score. First, the calculation methods for SNR margin $SNR_{m\arg in}$ and link budget LM are clarified. The formulas are given as Equation (40).

$$\begin{cases} SNR_{margin} = SNR_{avg} - SNR_{\min} - M_1 \\ LM = RSSI_{avg} - \text{Sensitivity} - M_2 \end{cases} \tag{40}$$

where SNR_{avg} and $RSSI_{avg}$ represent the average SNR and RSSI of the received packets within the window, respectively; $SNR_{\min}$ represents the minimum demodulation SNR for the current ED's SF, with the corresponding relationship between SF and minimum demodulation SNR shown in Table 7; Sensitivity represents the receiver sensitivity under the ED's current communication parameters; and M_1 and M_2 represent the extra margins for two margins. The calculations for M_1 and M_2 are given as Equation (41).

$$\begin{cases} M_1 = 10(1 - \text{Score}) \\ M_2 = 30 - 20 \cdot \text{Score} \end{cases} \tag{41}$$

where $Score \in [0, 1]$ represents the current ED's link state score, which measures the reliability and stability of the communication. When the link state score is low, both M_1 and M_2 take larger values, as the link stability is poor and adjustments to the data rate should be more conservative. Conversely, when the link state score is high, both M_1 and M_2 take smaller values.

Table 8 provides the description of each parameter in Algorithm 1. The detailed process of the ED optional parameter acquisition algorithm based on link state awareness is shown in Algorithm 1.

Table 7. Correspondence between SF and SNR_{min}.

SF	SNR_{min} (dB)
7	−7.5
8	−10
9	−12.5
10	−15
11	−17.5
12	−20

Algorithm 1 End device optional parameter acquisition algorithm based on link state.

Input: SF_{now}, BW_{now}, TP_{now}, score, SNR_{avg}, $RSSI_{avg}$, $node_{num}$
Output: $node_{num}$, TP, list(SF), list(BW)
$M_1 = 10(1 - score)$, $M_2 = 30 - 20 \cdot score$
$SF_{temp} = SF_{now}$, $BW_{temp} = BW_{now}$, $TP_{temp} = TP_{now}$, list(SF) = [], list(BW) = []
$SNR_{margin} = SNR_{avg} - SNR_{\min} - M_1$, $LM = RSSI_{avg} - \text{Sensitivity} - M_2$
$N_{step} = \text{int}\left(\frac{SNR_{margin}}{2.5}\right)$
if $N_{step} > 0$ **then**
  while $N_{step} > 0$ **and** $SF_{temp} > SF_{min}$ **do**
    $SF_{temp} = SF_{now} - 1$, $LM = LM - 2.5$ dB, $N_{step} = N_{step} - 1$
  end while
  while $N_{step} > 0$ **and** $TP_{temp} > TP_{min}$ **do**
    $TP_{temp} = TP_{now} - 3$ dB, $LM = LM - 3$ dB, $N_{step} = N_{step} - 1$
  end while
else
  if $N_{step} < 0$ **then**
    while $N_{step} < 0$ **and** $TP_{temp} < TP_{max}$ **do**
      $TP_{temp} = TP_{now} + 3$ dB, $LM = LM + 3$ dB, $N_{step} = N_{step} + 1$
    end while
    while $N_{step} < 0$ **and** $SF_{temp} < SF_{max}$ **do**
      $SF_{temp} = SF_{now} + 1$, $LM = LM + 2.5$ dB, $N_{step} = N_{step} + 1$
    end while
  end if
end if
if $LM > 0$ **then**
  while $LM - 3$ dB > 0 **and** $BW_{temp} < BW_{max}$ **do**
    $BW_{temp} = BW \times 2$, $LM = LM - 3$ dB
  end while
else
  while $LM + 3$ dB < 0 **and** $BW_{temp} > BW_{min}$ **do**
    $BW_{temp} = BW/2$, $LM = LM + 3$ dB
  end while
end if
while $BW_{temp} > BW_{min}$ **do**
  append BW_{temp} to list(BW), $BW_{temp} = BW_{temp}/2$
end while
Return: $node_{num}$, TP, list(BW), list(SF)

Algorithm 1 outlines the basic steps of the ED optional parameter acquisition algorithm based on link state. The specific process is as follows: First, based on the obtained SNR margin, the sufficiency of link resources is determined. When there is redundancy in the link resources, the SF is reduced to increase the data rate or the transmission power is lowered to reduce power consumption. When the link resources are insufficient, the transmission power is increased to improve the signal quality or the SF is increased to reduce the data rate. Additionally, when the theoretical link budget is large, the BW is increased to improve the data rate; when the theoretical link budget is small, the BW is reduced to lower the data rate. Based on the values of the SNR margin SNR_{Margin} and

theoretical link budget LM, the algorithm outputs a selectable list of transmission power and data rate under the premise of maintaining the communication reachability of the ED.

Table 8. Description of parameters in Algorithm 1.

Parameter	Description
$node_{num}$	Unique identifier of the ED.
$SF_{now}, BW_{now}, TP_{now}$	The SF, BW, and transmission power currently used by the ED.
$SNR_{avg}, RSSI_{avg}$	The average SNR and RSSI of packets received during the statistical window.
TP	The transmission power of the ED after adjustment.
$list(SF), list(BW)$	The list of selectable SFs and BWs available to the ED.
SNR_{margin}, LM	The SNR margin and link budget for the ED.
M_1, M_2	The extra margin calculated from the link state score.
N_{step}	The number of iterations in the algorithm.
SF_{min}, SF_{max}	The minimum and maximum SF values, from 7 to 12.
BW_{min}, BW_{max}	The minimum and maximum BW values, from 62.5 kHz to 500 kHz.
TP_{min}, TP_{max}	The minimum and maximum transmission power, from 2 dBm to 20 dBm.

5.2. Adaptive Data Rate Control Algorithm Based on Network Performance Optimization

The goal of network performance optimization is to allocate a data rate to each ED to maximize the network performance score and achieve optimal network performance. During the optimization process, it is essential to ensure the communication link quality between each ED and the GW. The optimal data rate must fall within the range of selectable data rates and other communication parameters determined by Algorithm 1. Additionally, the characteristics of the receiver and other relevant constraints must also be considered.

5.2.1. Zebra Optimization Algorithm

The Zebra Optimization Algorithm (ZOA) is a metaheuristic algorithm that solves optimization problems by simulating the foraging and defensive behaviors of zebras, searching for the global optimal solution from an initial solution [38,39]. Zebras rely on keen perception and swift running while foraging and evading predators. The algorithm includes foraging behavior and defensive behavior, and the basic steps are as follows:

Step 1: Initialization phase. An initial population of zebras (candidate solutions) is generated within the search space, where each zebra represents a potential solution.

Step 2: Foraging phase. The positions of the population are updated through foraging behavior. In this phase, the best member of the population (the lead zebra) guides the other members toward its position. The position update is represented by Equations (42) and (43).

$$x_{i,j}^{\text{new},P1} = x_{i,j} + r \cdot \left(PZ_j - I \cdot x_{i,j}\right) \tag{42}$$

$$X_i = \begin{cases} X_i^{\text{new},P1} & \text{if} \quad F_i^{\text{new}} < F_i \\ X_i & \text{else} \end{cases} \tag{43}$$

Equation (42), $x_{i,j}^{\text{new},P1}$ represents the new position of the i-th zebra in the j-th dimension, and $x_{i,j}$ represents the current position of the i-th zebra in the j-th dimension. r is a random number within the range $[0, 1]$, introducing randomness. PZ_j represents the position of the lead zebra (current optimal solution) in the j-th dimension, while I is a binary number that indicates the update strength, typically 0 or 1. Equation (43), X_i represents the current position of the i-th zebra, and $X_i^{\text{new},P1}$ is the updated position of the i-th zebra in foraging mode. If the fitness value $F_i^{\text{new},P1}$ of the updated position is better than the current fitness value F_i, the position is updated to $X_i^{\text{new},P1}$; otherwise, the position remains unchanged.

Step 3: Defense phase. This phase simulates the defensive actions of zebras to update the position of each member. When a zebra is attacked by its natural predator, the lion, its position is updated using S_1 in Equation (44). When it is attacked by other predators

(non-natural enemies), its position is updated using S_2 in Equation (44). The final update of the position is achieved using Equation (45).

$$x_{i,j}^{\text{new},P2} = \begin{cases} S_1 : x_{i,j} + R \cdot (2r-1) \cdot \left(1 - \frac{t}{T}\right) \cdot x_{i,j}, & \text{if} \quad P_s \leq 0.5 \\ S_2 : x_{i,j} + r \cdot \left(AZ_j - I \cdot x_{i,j}\right), & \text{else} \end{cases} \tag{44}$$

$$X_i = \begin{cases} X_i^{\text{new},P2}, & \text{if} \quad F_i^{\text{new},P2} < F \\ X_i, & \text{else} \end{cases} \tag{45}$$

Equation (44), $x_{i,j}^{\text{new},P2}$ represents the updated value of the i-th zebra in the j-th dimension in defense mode. Here, $x_{i,j}$ denotes the current position of the i-th zebra in the j-th dimension; P_s is a randomly generated probability in the range $[0, 1]$; S_1 and S_2 are adaptive scaling factors used to adjust the search range; R is a constant typically set to 2.5; r is a random number in the range $[0, 1]$; t and T represent the current iteration count and the maximum number of iterations, respectively; AZ_j represents the position of the lead zebra in the j-th dimension; and I is a constant used for adjustment. Equation (45), X_i represents the updated position of the i-th zebra. If the fitness value $F_i^{\text{new},P2}$ of the updated position is better than the current fitness value F_i, the position X_i is updated to $X_i^{\text{new},P2}$; otherwise, the position remains unchanged.

5.2.2. Adaptive Data Rate Algorithm for Network Performance Optimization

The main factors affecting network performance are the number of different data rates and the number of EDs using the same data rate. As the receiver of data from LoRa EDs, the GW can simultaneously demodulate LoRa signals with different data rates without interference. However, most GWs can demodulate a maximum of eight different data rates at the same time, which serves as a constraint for network performance optimization. Using the optional communication parameters for each ED obtained in Algorithm 1 as constraints, the implementation of the network performance optimization algorithm becomes a solution to an engineering optimization problem.

Assuming there are n EDs in the network and different data rates, the selectable transmission power for each ED obtained through Algorithm 1 is TP_i, the selectable SF list is $listSF_i$, and the selectable BW list is $listBW_i$. The number of EDs using data rate DR_k is num_k. The optimization objective of this problem is to maximize the network performance score $\mathrm{S}\left(\Gamma, E_{avg}, T_{avg}\right)$, and this optimization problem is represented by Equation (46).

$$\begin{aligned} &\max \ \mathrm{S}\left(\Gamma, E_{avg}, T_{avg}\right) \\ &s.t. \begin{cases} \sum_{k=1}^{m} num_k = n \\ m \leq 8 \\ TP_i = TP_i \\ SF_i \in \mathrm{listSF}_i \\ BW_i \in \mathrm{listBW}_i \end{cases} \end{aligned} \tag{46}$$

Table 9 describes each parameter used in the algorithm, and the algorithm flow of the network performance optimization algorithm is shown in Algorithm 2.

Algorithm 2 is an ADR algorithm based on network performance optimization, with the goal of obtaining the optimal transmission parameter combination for each ED in the network. The specific steps of the algorithm are as follows:

First, the algorithm takes each ED's TP, $list(SF)$, and $list(BW)$ as inputs, and selects the minimum SF and maximum BW to generate the initial configuration. Based on these initial configurations, the algorithm calculates the network performance score, including the average power consumption of the EDs, network throughput, and average latency of the EDs. Next, the algorithm generates an initial set of candidate solutions through the initialize_population() function, where each solution contains different ED configuration combinations.

Table 9. Description of parameters in Algorithm 2.

Parameter	Description
$\{node_{num}, TP_i, list_i(BW), list_i(SF)\}$	The set of optional parameters for each ED in the network
$list(node_{num}, TP, SF, BW)$	The optimal parameter combination for each ED in the network
$Bestscore$	The current optimal network performance score
$calculate_score()$	Score based on the current ED parameter combination
$initialize_population()$	Initial zebra population to find feasible candidate solutions
$iter, max_iterations$	Iteration count and maximum number of iterations
P_s	Probability values for different states in the defense phase
r	Random values used to adjust ED parameters
$zebra$	A set of candidate solutions for multiple EDs in the network
$terminal$	The optional parameters of a particular ED in the candidate solution
$combination_count$	The number of optional data rates for EDs in the network
$max_combinations$	The maximum number of data rates for EDs
$total_terminal_count$	The number of EDs in the candidate solution
$network_count$	The total number of EDs in the network

Algorithm 2 Adaptive data rate algorithm for network performance optimization.

Input: $\{node_{num}, TP_i, list(BW), list(SF)\}$, where $i \in (1, 2, \ldots, N)$
Output: $list(node_{num}, TP, SF, BW)$
$Bestscore = 0$, $list(node_{num}, TP, SF, BW) = []$
for each $node_{num}$ **do**
  $TP = \min(list(SF))$, $BW = \max(list(BW))$
  append $(node_{num}, TP, SF, BW)$ to $list(node_{num}, TP, SF, BW)$
  $Score = calculate_score(list(node_{num}, TP, SF, BW))$
  if $Score > Bestscore$ **then**
    $Bestscore = Score$
  end if
end for
$Zebra_population = initialize_population(list(node_{num}, TP, SF, BW))$
for $iter = 1$ to $max_iterations$ **do**
  $P_s = random(0, 1), r = random(0, 1)$
  if $P_s \leq 0.5$ **then**
    for each terminal in Zebra_population **do**
      $node_{num}, TP, SF, BW = terminal$
      $SF_{new} = random(list(SF))$, $BW_{new} = random(list(BW))$
      $terminal = (node_{num}, TP, SF_{new}, BW_{new})$
    end for
  else
    for each terminal in Zebra_population **do**
      $node_{num}, TP, SF, BW = terminal$
      $AZ_{terminal} = best_solution(node_{num})$
      $SF_{new} = SF + r \cdot (AZ_{SF} - SF)$
      $BW_{new} = BW + r \cdot (AZ_{BW} - BW)$
      $terminal = (node_{num}, TP, SF_{new}, BW_{new})$
    end for
  end if
  $Score = calculate_score(list(node_{num}, TP, SF, BW))$
  if $Score > Bestscore$ **then**
    if $combination_count \leq max_combinations$ **and** $total_terminal_count = network_count$ **then**
      $Bestscore = Score$
      $list(node_{num}, TP, SF, BW) = Zebra_population$
    end if
  end if
end for
Return: $list(node_{num}, TP, SF, BW)$

During the iterative process, the zebra population is updated using random probabilities and adjustment values. The selection method involves either randomly generating new configurations or adjusting based on the current best solution. For each updated configuration, the algorithm recalculates the network performance score. If the new score is higher than the current best score and meets the constraints (such as the maximum number of ED data rate combinations and the total number of EDs), the best score and configuration are updated. When the maximum number of iterations is reached or the optimal configuration is found, the algorithm returns $list(node_num, TP, SF, BW)$. Through this process, the algorithm gradually finds the optimal ED configuration combination in the complex search space, thereby improving overall network performance.

6. Testing and Result Analysis

To test the proposed ADR algorithm, this section designs an experimental scenario to verify the effectiveness of the algorithm from two perspectives: link state and network performance, assessing its performance in real-world applications. The experiment aims to determine whether the algorithm can effectively improve both link state and network performance. To ensure the accuracy of the results, the experiment was conducted in a real-world environment, testing multiple network performance metrics, including PRR, network throughput, average power consumption of the EDs, and average latency of the EDs. The experiment also compared the performance of this algorithm with the I-SFA scheme mentioned in [13], the LoRaWAN ADR algorithm, and the ADR+ algorithm.

6.1. Construction of LoRa Network

6.1.1. GW Hardware Design

The LoRa GW plays a key role in the LoRa network by serving as a protocol converter, responsible for converting IP packets from the network server into LoRa signals to be sent to the EDs and encapsulating the LoRa packets from the EDs into IP packets for upload to the network server. It mainly handles the forwarding of uplink and downlink messages, connecting LoRa EDs and the network server. Due to varying deployment environments, the GW needs to possess multiple networking capabilities, such as using Ethernet or Wi-Fi in urban areas, while relying on 3G/4G network connections in rural or outdoor settings. Additionally, the GW must support remote debugging and OTA (over-the-air) upgrades, and provide a certain level of computing power to support edge computing, thereby alleviating the burden on the network server.

In terms of hardware, the GW uses the MediaTek MT7620A (manufactured by MediaTek Inc., Hsinchu, Taiwan) as the main control chip, which not only has strong computing capabilities but also supports the OpenWRT system for easier development. The RF section employs a four-channel MCU + SX1262 unit instead of Semtech's SX1301 chip, reducing costs while supporting bandwidth adjustment. To meet network demands, the system is designed to operate with two GWs working in coordination, with each GW equipped with four antennas, collectively achieving a configuration of eight antennas. Each RF unit is controlled by the main control chip through USB-extended serial ports and driven by the MCU to operate the SX1262. The GW also includes a 4G module for wireless access, LAN and WAN ports for Ethernet connections, and a Wi-Fi module for client access. Figure 10 shows the schematic diagram of the GW hardware architecture; Figure 11 displays the physical image of the GW.

6.1.2. ED Hardware Design

In the LoRa network, the EDs are critical components responsible for collecting sensor data and transmitting them to the gateway. The hardware design needs to support multiple sensor interfaces to accommodate different types of sensors while considering low power consumption requirements. To meet these demands, the STM32F103RCT6 was selected as the microcontroller (MCU), paired with Semtech's SX1262 chip as the LoRa communication

module. The SX1262 chip features higher transmission power, smaller packaging, and lower power consumption, making it highly suitable for low-power EDs.

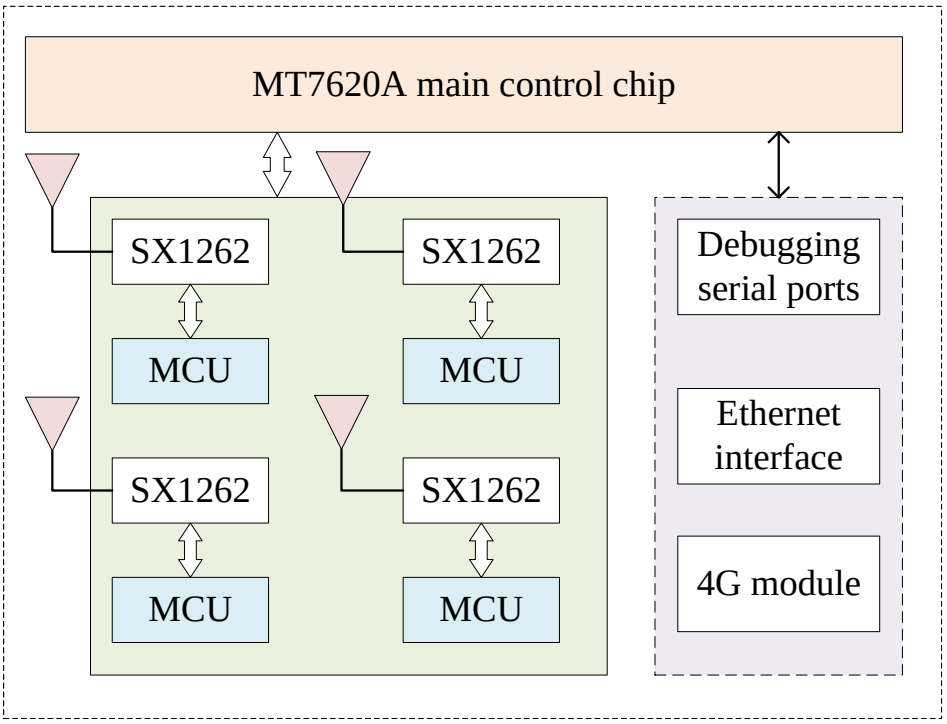


Figure 10. Schematic diagram of gateway hardware architecture.

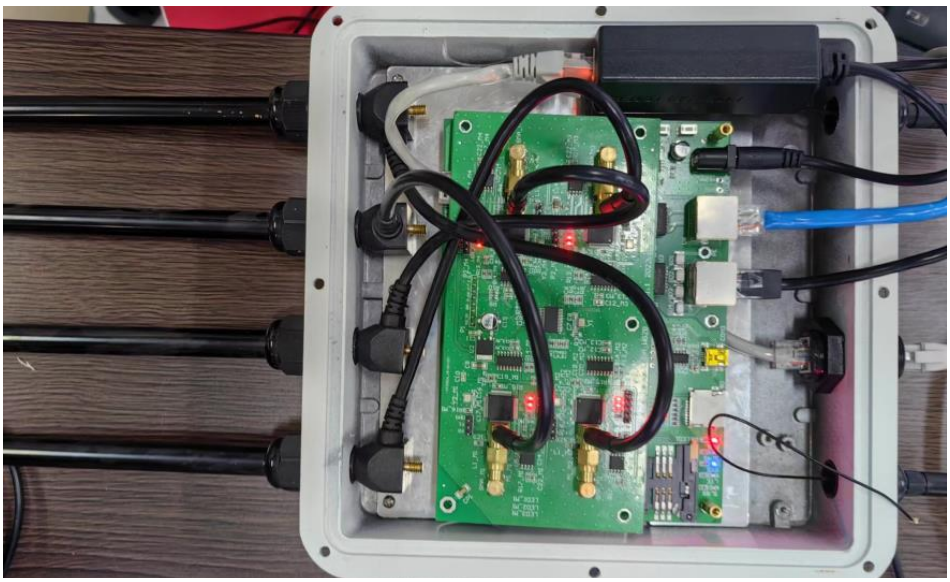


Figure 11. Physical image of the gateway.

For the power supply design, an AC–DC conversion chip was employed, supporting both 5 V or 3.3 V DC input and AC power supply options. Additionally, the ED is equipped with an SMA RF interface, connecting to the antenna via an SMA to IPEX RF cable for the transmission of LoRa signals. This design ensures reliable communication for EDs across various application scenarios. Figure 12 shows the structural diagram of the ED hardware module, while Figure 13 displays the physical image of the ED.

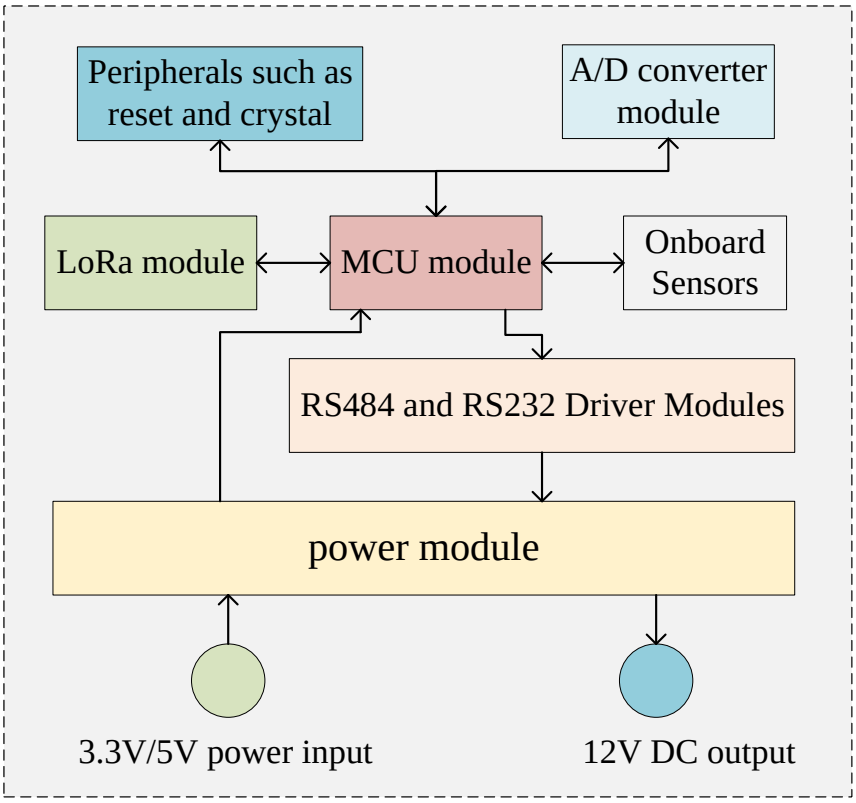


Figure 12. Schematic diagram of the ED hardware architecture.

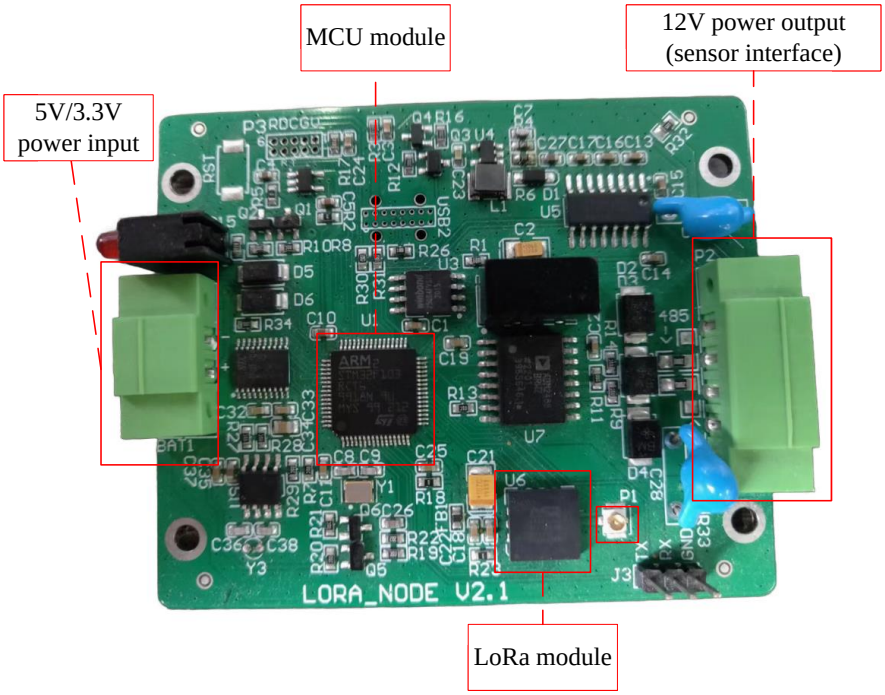


Figure 13. Physical image of the ED.

6.1.3. NS Design and Implementation

The NS is the core of the entire LoRa network, responsible for managing all devices and meeting various application service requirements. The NS designed for this study is developed using Java and deployed on an Alibaba Cloud Ubuntu 20.04 operating system.

It utilizes Docker technology to package the relevant environment and server program into a Docker image, simplifying the deployment process and enabling easy migration to different platforms.

The architecture of the NS is based on the Spring Boot, MyBatis, and Netty frameworks to enhance system scalability. The front end uses the React framework and adopts a separation of front-end and back-end design to ensure low coupling. The back-end program is quickly built using Spring Boot, which simplifies the coding, deployment, and configuration processes. Additionally, MyBatis is used as the persistence layer framework, significantly reducing the complexity of SQL connection queries and storage procedures.

Given that the NS needs to support concurrent connections from multiple gateways and devices, the Netty framework is employed to build TCP/UDP network applications, enabling it to handle large-scale, high-concurrency connections among network devices, thereby ensuring network stability and efficiency. Figure 14 shows the functional design of the NS.

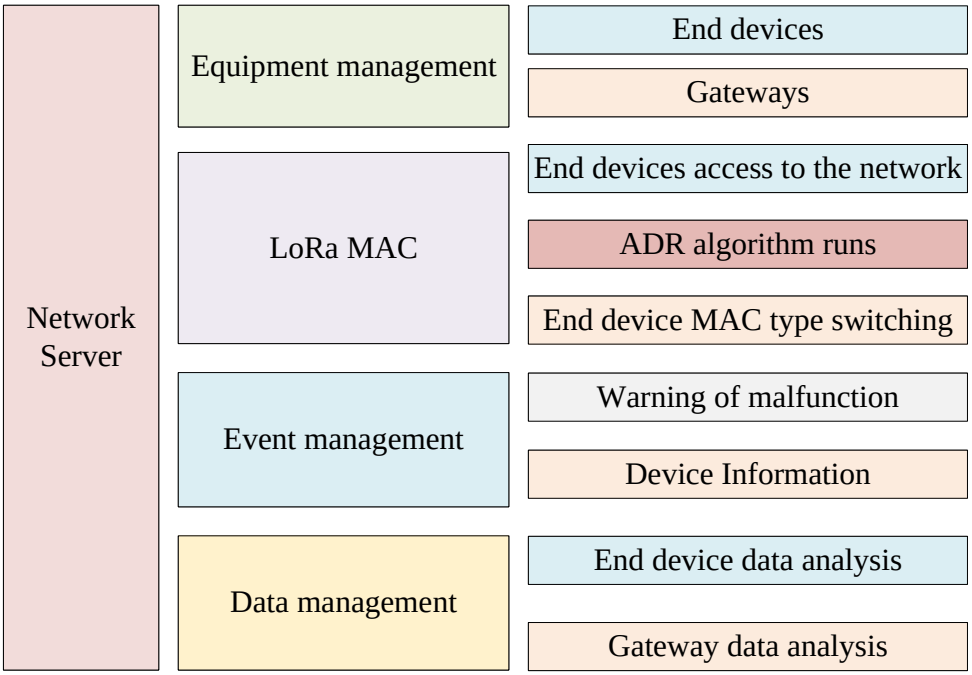
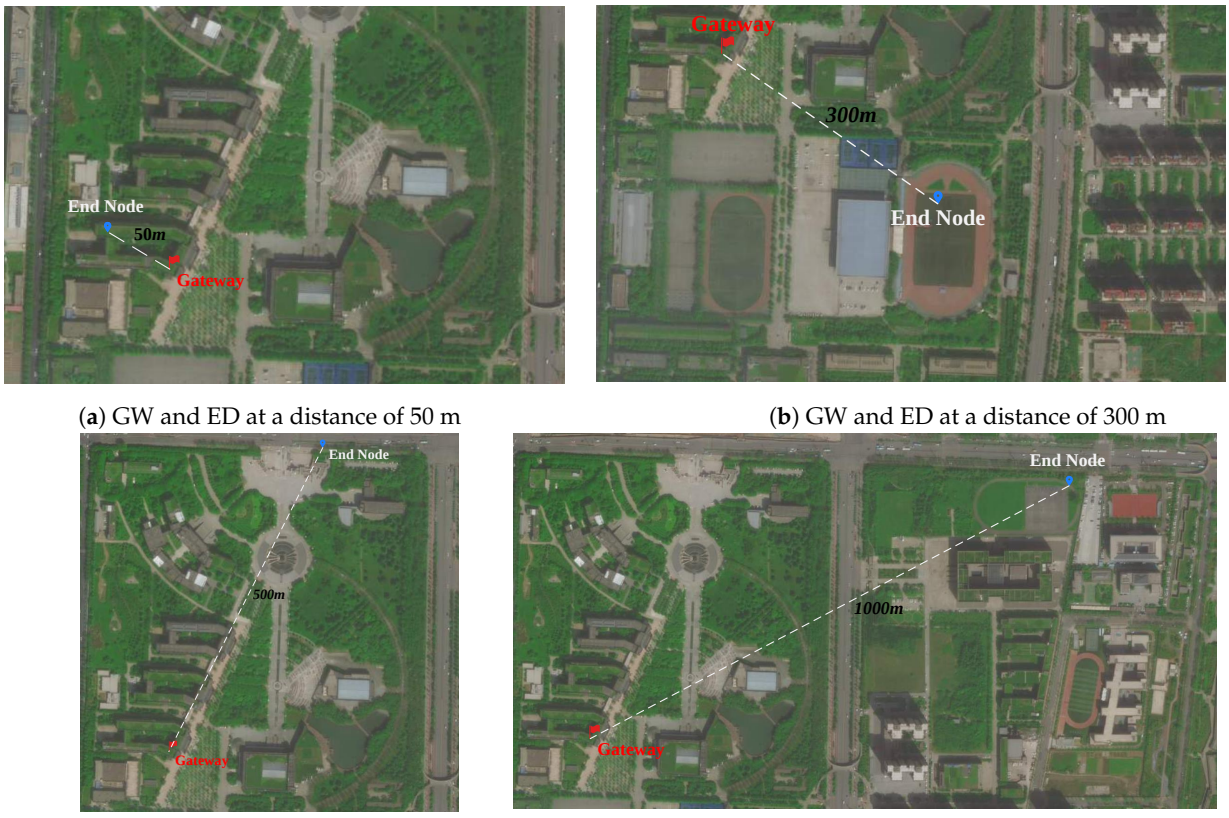


Figure 14. Functional design of the NS.

6.2. Single ED Communication Link Testing Scenario

In this test scenario, the GW was deployed on the roof of a 20 m high building, while the ED was located on a floor inside the same building, with a straight-line distance of 50 m between them. The deployment diagram of the ED and GW is shown in Figure 15. The initial communication parameters of the ED are shown in Table 10, configured with a 500 kHz modulation BW and SF 7. The node generates one 20-byte data packet per second.

The position of the GW is fixed, while the EDs select suitable locations within the given distances for testing in a campus environment, such as inside buildings (50 m), on the playground (300 m), at the north gate (500 m), and in the northeast corner of the campus (1000 m). This scenario aims to test the optimization effects on the link performance of a single ED and compare the impact on PRR among I-SFA, LoRaWAN ADR, ADR+, and the algorithm proposed in this paper over a period of 3600 s.



(**a**) GW and ED at a distance of 50 m

(**b**) GW and ED at a distance of 300 m

(**c**) GW and ED at a distance of 500 m

(**d**) GW and ED at a distance of 1000 m

Figure 15. Schematic diagram of GW and ED distribution at different distances.

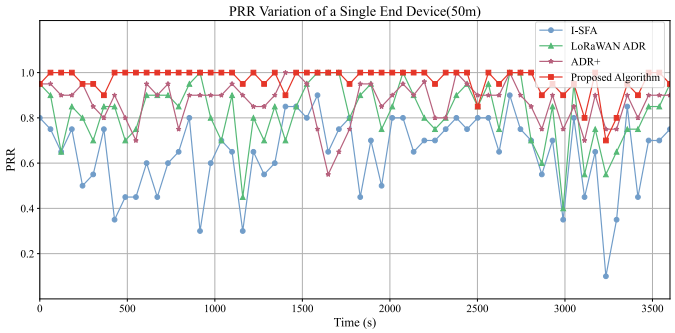
Table 10. Initial communication parameter configuration of the ED.

Parameter	Value
Channel Frequency	500 MHz
BW	500 kHz
SF	7
Transmission Power	22 dBm
Coding Rate	4/5
Transmission Interval	20 s
Actual Data Rate	21.88 kbps

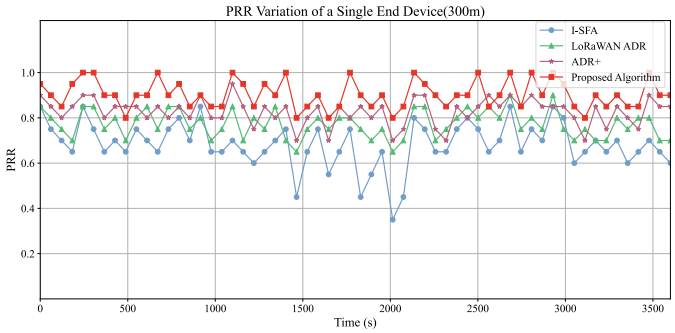
Figure 16 shows the variation of PRR over 3600 s for four adaptive data rate adjustment schemes under a single ED communication link at different testing distances, while Figure 17 compares the average PRR at four distances. The test results indicate that the proposed algorithm maintains a PRR of over 90% across all testing distances, with its advantages becoming increasingly apparent as the distance increases. Inside the building environment (50 m), the proposed algorithm is able to maintain a reliable link performance despite the link fluctuations caused by building shading and people walking around. At a distance of 1000 m, the PRR of the proposed algorithm improves by 20% compared to the standard LoRaWAN ADR algorithm.

Although the ADR+ algorithm shows performance improvements over the standard LoRaWAN ADR algorithm, it still has limitations compared to the proposed algorithm. The I-SFA scheme primarily allocates the spreading factor during the initial phase at the ED, configured according to the sensitivity of the gateway, but does not consider the reliability and stability of the link, nor does it possess the ability to adaptively adjust the spreading factor. The standard ADR scheme uses the maximum SNR within a fixed time window to assess link status, while the ADR+ scheme employs the average SNR. However, both

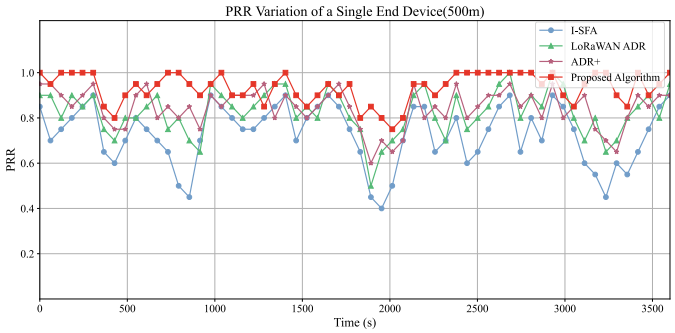
schemes inadequately assess the stability of the link, failing to effectively address short-term fluctuations in link status, resulting in limited improvements in packet reception rates.



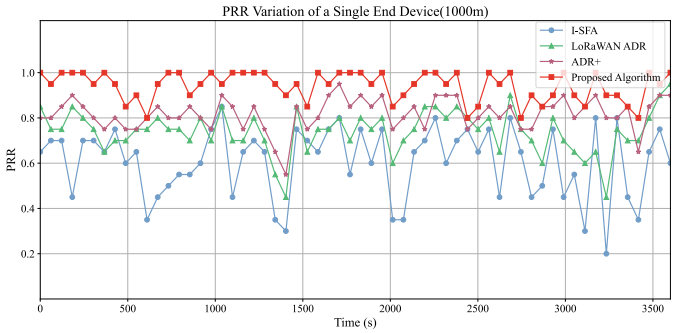
(**a**) GW and ED at a distance of 50 m



(**b**) GW and ED at a distance of 300 m

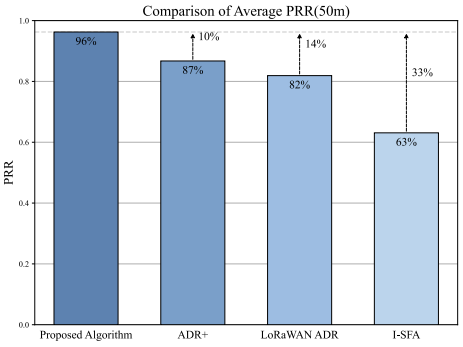


(**c**) GW and ED at a distance of 500 m

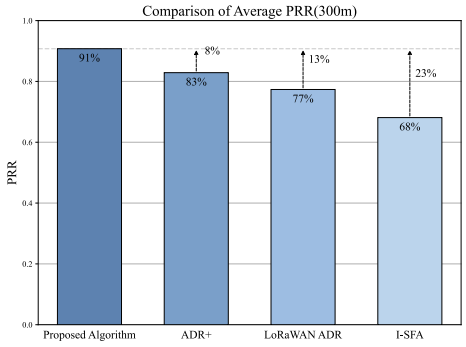


(**d**) GW and ED at a distance of 1000 m

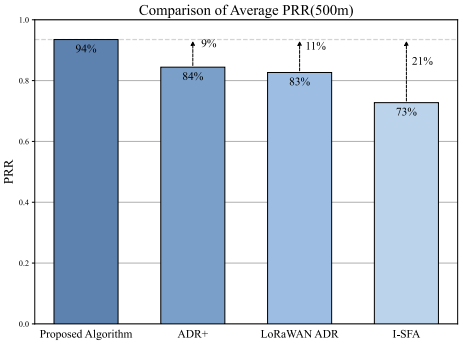
Figure 16. PRR variation curve at different distances for different ADR algorithms of a single ED.



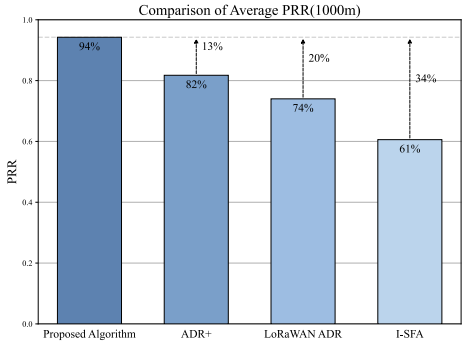
(**a**) GW and ED at a distance of 50 m



(**b**) GW and ED at a distance of 300 m



(**c**) GW and ED at a distance of 500 m



(**d**) GW and ED at a distance of 1000 m

Figure 17. Comparison of average PRR at different distances for different ADR algorithms of a single ED.

In contrast, the proposed algorithm evaluates the link state from both stability and reliability perspectives, taking into account factors such as average SNR, average signal strength, PRR, SNR variance, and signal strength variance. This multidimensional assessment method provides a more accurate judgment of link conditions, allowing the algorithm to demonstrate significant advantages in long-distance communication.

6.3. Multi-ED Network Performance Testing Scenario

The GW is deployed on the roof of a 20 m high building, with 50 EDs distributed around the GW in campus and urban environments. Table 11 shows the initial communication parameter configuration of the EDs in the network. Figure 18 illustrates the distribution of the GW and EDs in the LoRa network.

This experimental scenario is used to test the proposed algorithm's optimization of network throughput, average energy consumption of EDs, and average latency of EDs in the LoRa network, and compares the network performance with the I-SFA scheme, LoRaWAN ADR, and ADR+ algorithms. Figure 19 shows the actual deployment environment of the EDs and the GW.

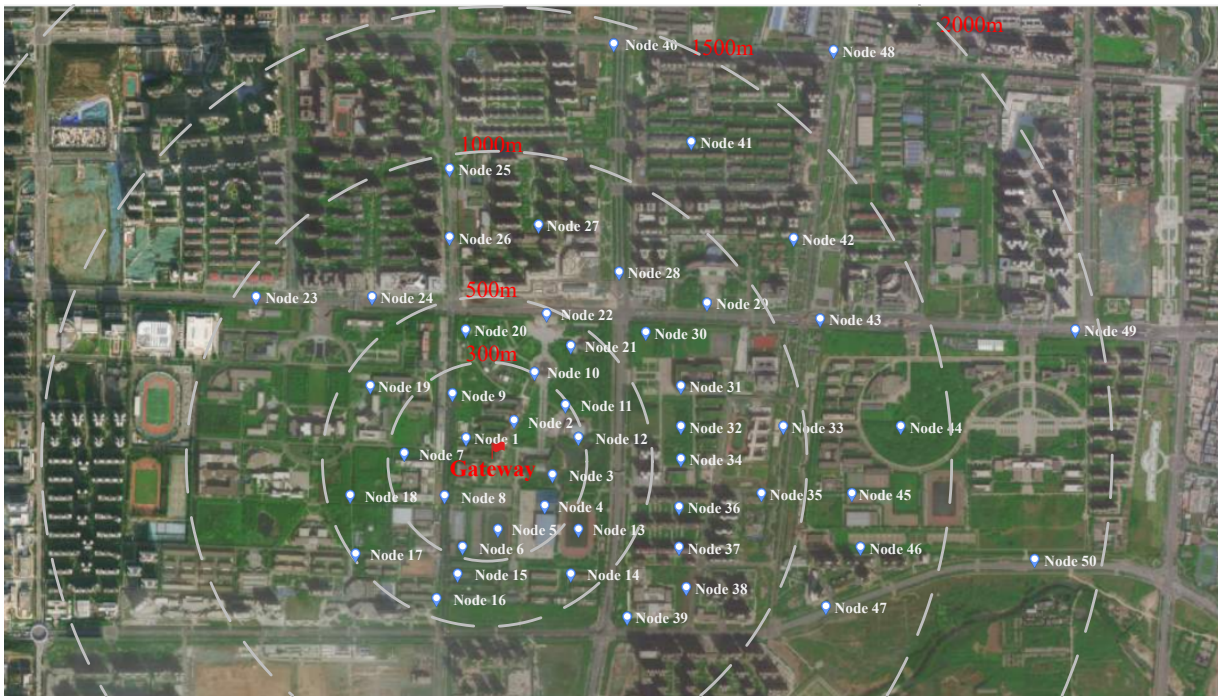
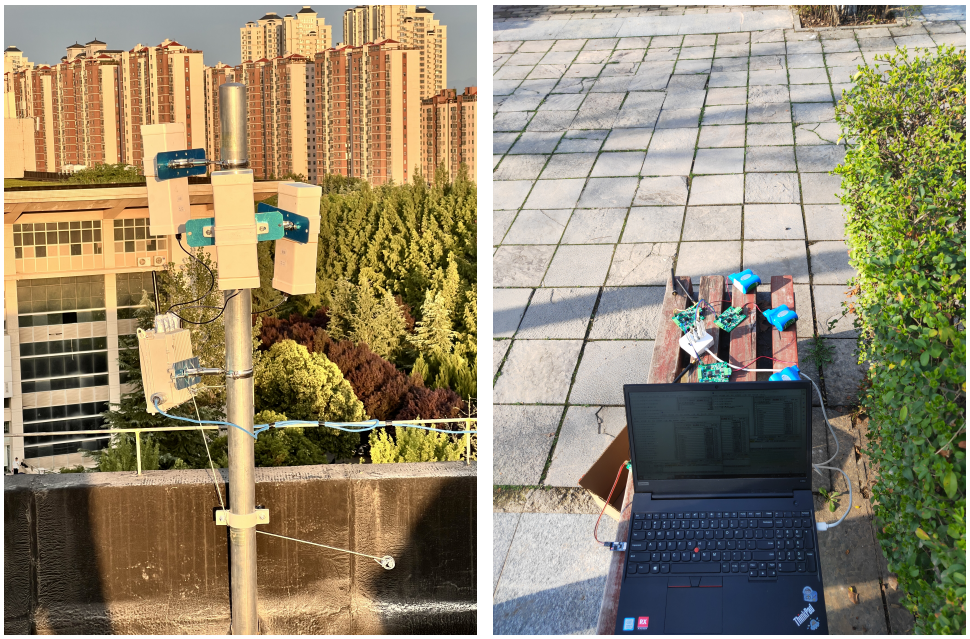


Figure 18. Deployment distribution of GW and ED in the LoRa network.

The results show the number of EDs corresponding to different data rates, as calculated by the NS after the algorithm has been completed. According to the algorithm's results, EDs that are closer to the GW and have better link quality are preferentially assigned higher data rates, thereby minimizing the use of lower data rates while ensuring communication reachability. There are 17 end devices (EDs) operating with the combination of spreading factor (SF) 7 and bandwidth (BW) 500, while 4 EDs use SF 7 and BW 250. For the combination of SF 8 and BW 500, 10 EDs are in operation, and 4 EDs use SF 9 and BW 500. Additionally, 5 EDs operate with SF 10 and BW 250, and 3 EDs use SF 11 and BW 500. Finally, 5 EDs correspond to the combination of SF 12 and BW 500, and 2 EDs use SF 12 and BW 125. To optimize network performance and reduce ED power consumption, a larger number of EDs are assigned higher data rates, while fewer devices are assigned lower data rates. This reduces the probability of packet collisions during transmission.



(**a**) GW actual location (**b**) EDs deployment example

Figure 19. Actual deployment of GW and EDs.

Table 11. Initial communication parameter configuration of EDs in the network.

Parameter	Value
BW	62.5 kHz
SF	12
Coding Rate	4/5
Preamble Length	10 Bytes
Packet Length	20 Bytes
Transmission Power	20 dBm
Channel Frequency	500 MHz
Packet Generation Rate	1 packet every 20 s

At the beginning of network operation, all EDs work at the lowest data rate, generating a 20-byte data packet every 20 s. Figure 20 shows a comparison of the average power consumption of EDs under four different ADR algorithms. The results indicate that the proposed algorithm reduces the average power consumption of EDs by 15.9% compared to the I-SFA scheme; in contrast, the LoRaWAN ADR algorithm and the ADR+ algorithm reduce power consumption by 6.2% and 11.3%, respectively. This advantage arises from the comprehensive evaluation of the link state in the proposed algorithm, which determines the optimal data rate for each ED, thereby effectively reducing power consumption. The I-SFA scheme, on the other hand, assigns SFs based solely on RSSI without optimizing power consumption. The LoRaWAN ADR and ADR+ algorithms rely on partial statistical information, causing EDs to tend to use faster data rates, which increases the probability of packet collisions and, consequently, the average power consumption of the EDs.

Figure 21 shows a comparison of network throughput and average latency of EDs for four different ADR algorithms. The results indicate that the proposed algorithm significantly increases the network throughput corresponding to the data rates of EDs, surpassing the I-SFA scheme and outperforming the LoRaWAN ADR algorithm and the ADR+ algorithm. At the same time, the average latency of the EDs under the proposed ADR algorithm is slightly lower than that of the other three algorithms. This is primarily due to the proposed algorithm's link state awareness, which obtains the transmission power and a list of selectable data rates for each ED, optimizing the stability and reliability of communication links. To further optimize network performance metrics, the algorithm

continuously adjusts the data rates of the EDs, ultimately finding a set of optimal data rate combinations that minimize packet collision probability, maximize network throughput, and minimize average latency. By comprehensively considering the link state of the EDs and network performance, the algorithm minimizes average power consumption while maximizing network throughput and minimizing average latency.

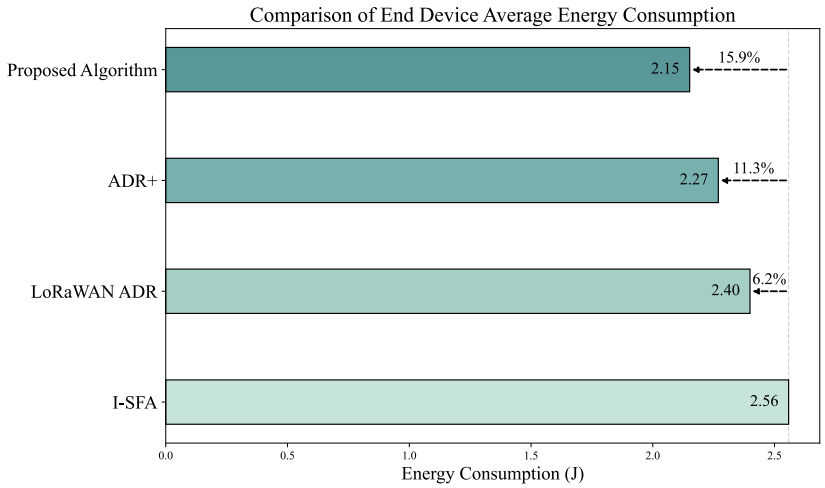


Figure 20. Comparison of average power consumption of EDs for different ADR algorithms.

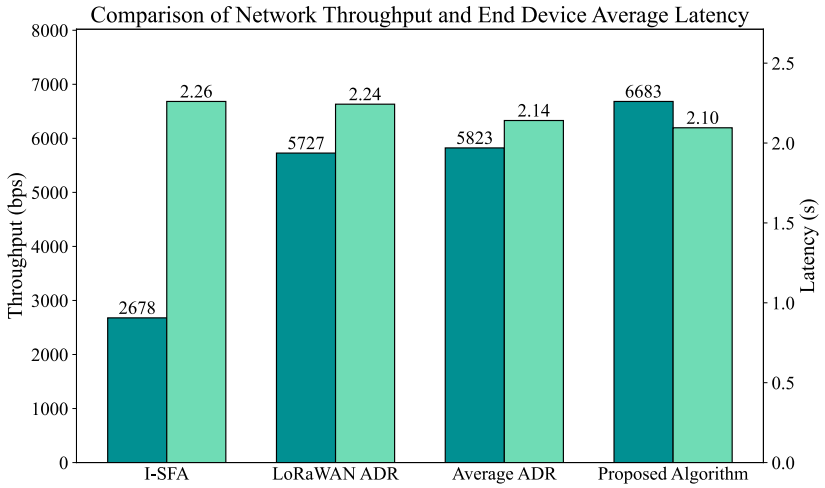


Figure 21. Comparison of network throughput and average latency of EDs for different ADR algorithms.

6.4. Algorithm Complexity Analysis

On a standard server equipped with a 2.4 GHz CPU and 8 GB of memory, the execution time of the algorithm can be divided into two parts. First, the transmission power and selectable data rate list for each ED are extracted based on the link state, with the algorithm's complexity being. The runtime for a single ED is 8.012 ms, which includes saving the relevant information. Second, based on the selectable communication parameters of the EDs in the network, the data rate required to achieve optimal network performance for each ED is calculated, with the complexity increasing as the number of EDs in the network grows. In a scenario with 50 EDs and 500 iterations, the total execution time of the algorithm is 0.95 s. This demonstrates that the algorithm can meet application requirements.

7. Conclusions and Future Work

This paper proposes an ADR algorithm specifically designed for LoRa networks with EDs constrained by limited power in actual deployments. First, based on link state

awareness, the algorithm assesses the transmission power and optional data rate list for each ED in the network. Second, considering network throughput, average ED power consumption, and average ED delay, an optimization model is established to determine the objective function for network performance enhancement, leading to the design of an algorithm capable of adaptively adjusting the data rate. Finally, the proposed algorithm is validated in a deployed LoRa network.

The proposed ADR algorithm achieves adaptability in static ED scenarios through link state awareness and dynamic adjustments. Specifically, the algorithm dynamically selects appropriate transmission power and data rates by real-time evaluation of link states such as SNR and RSSI. This effectively addresses fluctuations in channel quality, interference, and changes in network topology, ensuring optimal communication performance across various scenarios. The proposed algorithm, based on real-time monitoring and parameter adjustment, demonstrates adaptability in multiple scenarios, including static and dynamic node deployments and variations in network density. Additionally, by integrating ED power consumption, network throughput, and delay models, the algorithm can flexibly optimize communication parameters to enhance overall network performance.

In future work, the LoRa communication characteristics inside the building environment can be explored, while by combining with techniques such as transfer learning or few-shot learning, the flexibility and adaptive capability of the system can be further enhanced, especially when facing brand new scenarios or environments with sparse business data; the system will be able to adapt faster to different deployment environments or new network conditions, and utilize a small amount of data for rapid learning and adjustment,. This further enhances the system's adaptive capability.

Author Contributions: Conceptualization, H.W. and B.Z.; methodology, B.Z.; software, B.Z.; validation, H.W. and S.P.; formal analysis, R.P. and H.W.; investigation, X.L. and J.S.; resources, B.Z.; data curation, R.P.; writing—original draft preparation, B.Z.; writing—review and editing, H.W. and R.P.; visualization, H.W.; supervision, B.Z.; project administration, B.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the Key Industry Innovation Chain Project of Shaanxi Province (No. 2021ZDLGY07-10, No. 2021ZDLNY03-08), the Science and Technology Plan Project of Shaanxi Province (No. 2022GY-045), the Key Research and Development plan of Shaanxi Province (No. 2024GX-ZDCYL-01-33, No. 2018ZDXM-GY-041), Scientific Research Program Funded by Shaanxi Provincial Education Department (Program No. 21JC030), and the Science and Technology Plan Project of Xi'an (No. 22GXFW0124, No. 2019GXYD17.3).

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Noreen, U.; Bounceur, A.; Clavier, L. A study of LoRa low power and wide area network technology. In Proceedings of the 2017 International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), Fez, Morocco, 22–24 May 2017; pp. 1–6. [CrossRef]
2. Semtech Corporation. LoRa Modulation Basics. 2015. Available online: https://www.frugalprototype.com/wp-content/uploads/2016/08/an1200.22.pdf (accessed on 19 September 2024).
3. Semtech Corporation. LoRa and LoRaWAN: A Technical Overview. Available online: https://www.katykoenen.com/wp-content/uploads/2024/07/LoRa_and_LoRaWAN-A_Tech_Overview-Downloadable.pdf (accessed on 19 September 2024).
4. Souifi, J.; Bouslimani, Y.; Ghribi, M.; Kaddouri, A.; Boutot, T.; Abdallah, H.H. Smart Home Architecture Based on LoRa Wireless Connectivity and LoRaWAN® Networking Protocol. In Proceedings of the 2020 1st International Conference on Communications, Control Systems and Signal Processing (CCSSP), El Oued, Algeria, 16–17 May 2020; pp. 95–99. [CrossRef]
5. Yang, Y. Design and Application of Intelligent Agriculture Service System with LoRa-Based on Wireless Sensor Network. In Proceedings of the 2020 International Conference on Computer Engineering and Application (ICCEA), Guangzhou, China, 18–20 March 2020; pp. 712–716. [CrossRef]
6. Tomar, R.; Gemein, O.-G. LoRa Network for Cities: Private and Complete Secured by Design. In Proceedings of the 2018 Global Internet of Things Summit (GIoTS), Bilbao, Spain, 4–7 June 2018; pp. 1–5. [CrossRef]

7. Olatinwo, D.D.; Abu-Mahfouz, A.; Hancke, G. A Survey on LPWAN Technologies in WBAN for Remote Health-Care Monitoring. *Sensors* **2019**, *19*, 5268. [CrossRef] [PubMed]
8. Kufakunesu, R.; Hancke, G.P.; Abu-Mahfouz, A.M. A Survey on Adaptive Data Rate Optimization in LoRaWAN: Recent Solutions and Major Challenges. *Sensors* **2020**, *20*, 5044. [CrossRef]
9. Semtech Corporation. LoRaWAN—Simple Rate Adaptation Recommended Algorithm. Semtech. 2016. Available online: https://www.thethingsnetwork.org/forum/uploads/default/original/2X/7/7480e044aa93a54a910dab8ef0adfb5f515d14a1.pdf (accessed on 19 September 2024).
10. Semtech Corporation. Understanding the LoRa® Adaptive Data Rate. Semtech. December 2019. Available online: https://www.binefa.cat/training/20221020_viladecans/Understanding_LoRa_Adaptive_Data_Rate_Downloadable.pdf (accessed on 27 October 2024).
11. Slabicki, M.; Premsankar, G.; Di Francesco, M. Adaptive Configuration of LoRa Networks for Dense IoT Deployments. In Proceedings of the 2018 IEEE/IFIP Network Operations and Management Symposium (NOMS), Taipei, Taiwan, 23–27 April 2018; pp. 1–5. [CrossRef]
12. Abdelfadeel, K.Q.; Cionca, V.; Pesch, D. Fair adaptive data rate allocation and power control in LoRaWAN. In Proceedings of the 2018 IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM), Chania, Greece, 14–15 June 2018. [CrossRef]
13. Magrin, D.; Centenaro, M.; Vangelista, L. Performance evaluation of LoRa networks in a smart city scenario. In Proceedings of the 2017 IEEE International Conference on Communications (ICC), Paris, France, 21–25 May 2017; pp. 1–7. [CrossRef]
14. Marini, R.; Cerroni, W.; Buratti, C. A novel collision-aware adaptive data rate algorithm for LoRaWAN networks. *IEEE Internet Things J.* **2020**, *8*, 2670–2680. [CrossRef]
15. Zhou, Q.; Xing, J.; Hou, L.; Xu, R.; Zheng, K. A novel rate and channel control scheme based on data extraction rate for LoRa networks. In Proceedings of the 2019 IEEE Wireless Communications and Networking Conference (WCNC), Marrakech, Morocco, 15–18 April 2019; pp. 1–6. [CrossRef]
16. Sandoval, R.M.; Garcia-Sanchez, A.-J.; Garcia-Haro, J. Optimizing and updating LoRa communication parameters: A machine learning approach. *IEEE Trans. Netw. Serv. Manag.* **2019**, *16*, 884–895. [CrossRef]
17. Zhao, Z.; Gao, W.; Du, W.; Min, G.; Mao, W.; Singhal, M. Towards Energy-Fairness in LoRa Networks. *IEEE Trans. Mob. Comput.* **2023**, *22*, 5597–5610. [CrossRef]
18. Pang, S.; Li, Z.; Yao, Z.; Wang, H.; Long, W.; Pan, R. Research on the Evaluation and Prediction of V2I Channel Quality Levels in Urban Environments. *Electronics* **2024**, *13*, 911. [CrossRef]
19. Park, J.; Park, K.; Bae, H.; Kim, C.-K. EARN: Enhanced ADR With Coding Rate Adaptation in LoRaWAN. *IEEE Internet Things J.* **2020**, *7*, 11873–11883. [CrossRef]
20. de Jesus, G.G.M.; Souza, R.D.; Montez, C.; Hoeller, A. LoRaWAN Adaptive Data Rate with Flexible Link Margin. *IEEE Internet Things J.* **2021**, *8*, 6053–6061. [CrossRef]
21. Jiang, Y.; Wang, M.; Wang, X. A Efficient Adaptive Data Rate Algorithm in LoRaWAN Networks: K-ADR. In Proceedings of the 2023 24th Asia-Pacific Network Operations and Management Symposium (APNOMS), Sejong, Republic of Korea, 6–8 September 2023; pp. 183–188.
22. Lodhi, M.A.; Wang, L.; Farhad, A. ND-ADR: Nondestructive adaptive data rate for LoRaWAN Internet of Things. *Int. J. Commun. Syst.* **2022**, *35*, e5136. [CrossRef]
23. Xia, X.; Chen, Q.; Hou, N.; Zheng, Y.; Li, M. XCopy: Boosting Weak Links for Reliable LoRa Communication. In Proceedings of the 29th Annual International Conference on Mobile Computing and Networking, Madrid, Spain, 2–6 October 2023; pp. 1–15. [CrossRef]
24. Sun, Z.; Ni, T.; Yang, H.; Liu, K.; Zhang, Y.; Gu, T.; Xu, W. FLoRa: Energy-efficient, Reliable, and Beamforming-assisted Over-the-air Firmware Update in LoRa Networks. In Proceedings of the 22nd International Conference on Information Processing in Sensor Networks, San Antonio, TX, USA, 9–12 May 2023; pp. 14–26. [CrossRef]
25. Hou, N.; Xia, X.; Wang, Y.; Zheng, Y. One Shot for All: Quick and Accurate Data Aggregation for LPWANs. In Proceedings of the IEEE INFOCOM 2023—IEEE Conference on Computer Communications, New York, NY, USA, 17–20 May 2023. [CrossRef]
26. Ampoma, A.E.; Zhang, H.; Huang, Y.; Wen, G.; Kwame, O.G. Three-Dimensional Spatial Fading Correlation of Uniform Rectangular Array Using Total Power of Angular Distribution. *IEEE Antennas Wirel. Propag. Lett.* **2017**, *16*, 2134–2137. [CrossRef]
27. Youssef, N.; Wang, C.-X.; Patzold, M.; Jaafar, I.; Tabbane, S. On the Statistical Properties of Generalized Rice Multipath Fading Channels. In Proceedings of the 2004 IEEE 59th Vehicular Technology Conference (VTC 2004-Spring), Milan, Italy, 17–19 May 2004; Volime 1, pp. 162–165. [CrossRef]
28. Lu, J.-H.; Han, Y. Application of Multipath Shape Factors in Nakagami-m Fading Channel. In Proceedings of the 2009 International Conference on Wireless Communications & Signal Processing, Nanjing, China, 13–15 November 2009; pp. 1–4. [CrossRef]
29. Zacharias, S.; Newe, T.; O'Keeffe, S.; Lewis, E. 2.4 GHz IEEE 802.15.4 Channel Interference Classification Algorithm Running Live on a Sensor Node. In Proceedings of the SENSORS, 2012 IEEE, Taipei, Taiwan, 28–31 October 2012; pp. 1–4. [CrossRef]
30. Wang, H.; Jia, J.; Liu, M.; Pan, R. Research on Data-Driven LoRa Link Quality Estimation. In Proceedings of the 2022 4th International Conference on Natural Language Processing (ICNLP), Xi'an, China, 18–21 December 2022; pp. 580–586. [CrossRef]
31. Luo, X.; Liu, L.; Shu, J.; Al-Kali, M. Link Quality Estimation Method for Wireless Sensor Networks Based on Stacked Autoencoder. *IEEE Access* **2019**, *7*, 21572–21583. [CrossRef]

32. Li, M.; Yang, M.; Yu, Y.; Shahidehpour, M.; Wen, F. Adaptive Weighted Combination Approach for Wind Power Forecast Based on Deep Deterministic Policy Gradient Method. *IEEE Trans. Power Syst.* **2024**, *39*, 3075–3087. [CrossRef]
33. Balusa, B.C.; Gorai, A.K. Sensitivity analysis of fuzzy-analytic hierarchical process (FAHP) decision-making model in selection of underground metal mining method. *J. Sustain. Min.* **2019**, *18*, 8–17. [CrossRef]
34. Ahmed, F.; Kilic, K. Fuzzy Analytic Hierarchy Process: A Performance Analysis of Various Algorithms. *Fuzzy Sets Syst.* **2019**, *362*, 110–128. [CrossRef]
35. Diakoulaki, D.; Mavrotas, G.; Papayannakis, L. A Multicriteria Approach for Evaluating the Performance of Industrial Firms. *Omega* **1992**, *20*, 467–474. [CrossRef]
36. Krishnan, A.R.; Kasim, M.M.; Hamid, R.; Ghazali, M.F. A Modified CRITIC Method to Estimate the Objective Weights of Decision Criteria. *Symmetry* **2021**, *13*, 973. [CrossRef]
37. Opricovic, S.; Tzeng, G.-H. Multicriteria Planning of Post-Earthquake Sustainable Reconstruction. *Comput.-Aided Civ. Infrastruct. Eng.* **2002**, *17*. [CrossRef]
38. Trojovská, E.; Dehghani, M.; Trojovský, P. Zebra Optimization Algorithm: A New Bio-Inspired Optimization Algorithm for Solving Optimization Algorithm. *IEEE Access* **2022**, *10*, 49445–49473. [CrossRef]
39. Vijayaraj, V.; Balamurugan, M.; Oberoi, M. Balancing the Load and Scheduling the Tasks Using Zebra Optimizer in IoT-Based Cloud Computing for Big-Data Applications. *Métod. Numér. Para Cálc. Diseño Ing. Rev. Int.* **2024**, *40*, 1–17. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.