Challenges and optimization ideas in low-power design for Internet of Things devices

Jiulong Zhang *, LinluoYao, Jinghua Cui

Faculty of Engineering, University Malaya, Kuala Lumpur, Malaysia

Received: July 17, 2025

Revised: July 29, 2025

Accepted: August 6, 2025

Published online: August 11, 2025

To appear in: *International Journal of Advanced AI Applications,* Vol. 1, No. 6 (October 2025)

* Corresponding Author: Jiulong Zhang (zhangjlyyds888@163.com)

Abstract. This paper presents a systematic low-power optimization framework for Internet of Things (IoT) devices, aiming to address persistent energy consumption challenges in long-term deployments. By constructing a modular simulation model with four evaluation metrics—average power, response time, battery life, and task success rate—we compare baseline, hardware-only, and collaborative software-hardware strategies. Results show the proposed approach reduces average power to 7.9mW, extends battery life to 158.6 hours, and achieves a 97.2% task completion rate. Field deployment confirms its adaptability and reliability. The framework provides a practical and scalable solution for low-power IoT design.

Keywords: *low power design; Internet of Things devices; software and hardware collaboration; power consumption modeling.*

1. Introduction

As the application of IoT technology becomes increasingly widespread, the energy consumption of terminal devices has become a major issue that hinders system stability and promotion [1]. In smart homes, industrial control, and agricultural monitoring, equipment is often deployed in areas with limited energy resources, where low-power design is essential. Traditional designs often overlook the cumulative power consumption, leading to frequent system failures and increased maintenance costs [2]. This study explores multi-level optimization strategies based on the sources of power consumption, providing a systematic approach to energy savings through simulation and deployment results. Despite existing research efforts in hardware optimization or protocol simplification, many approaches still suffer from limited adaptability to diverse application scenarios, insufficient real-world validation, and lack of synergy between software and hardware layers. This paper addresses these limitations by proposing an integrated optimization framework, combining deep sleep control, dynamic voltage scaling, and edge intelligence to improve both energy efficiency and system responsiveness. By constructing a comprehensive simulation model and conducting field deployment validation, this study fills the gap between theoretical optimization and practical application, offering a novel and applicable low-power design strategy for IoT devices.

In addition to the commonly acknowledged challenges such as limited battery capacity, energy-hungry wireless communication, and dynamic workload patterns, several critical yet often underemphasized issues in low-power IoT design warrant closer attention. First, security-energy trade-offs pose a growing concern, as ensuring device security (e.g., through encryption) often increases energy consumption significantly. Second, variability in ambient conditions, such as temperature and humidity fluctuations, can affect the stability and efficiency of power management systems, especially in outdoor or industrial IoT environments. Third, heterogeneity in hardware platforms leads to inconsistent power optimization outcomes, making it difficult to develop universally applicable strategies. Lastly, real-time processing demands for edge intelligence have introduced higher baseline power needs, further complicating the balancing act between computation and energy efficiency. Addressing these less discussed but impactful challenges is essential to building robust and scalable low-power IoT systems.

2. Overview of low power design of Internet of Things devices

2.1 Application scenarios and power consumption characteristics

IoT devices are widely used in various scenarios, including environmental monitoring, logistics tracking, health wearables, and intelligent transportation [3]. These devices typically require prolonged online operation and feature periodic wake-up, instant data transmission, and low-power standby modes [4]. The communication module, sensors, and display units are the primary sources of energy consumption [5]. Different application scenarios have varying requirements for task cycles and power consumption, so low-power design must balance device performance with considerations for battery life, real-time responsiveness, and environmental conditions [6]. For instance, in environmental monitoring, the device needs to operate continuously and monitor data over extended periods, whereas in intelligent transportation, it must have real-time response capabilities and high data transmission efficiency. Low-power design requires a comprehensive evaluation of all metrics to optimize power performance [7].

2.2 Current design architecture and power consumption bottleneck

Most IoT devices use a combination of Cortex-M MCU and wireless communication modules such as Bluetooth Low Energy (BLE) and LoRa. While this design can optimize power consumption, it still faces several bottlenecks, particularly in the areas of communication wake-up, standby management, and polling interrupts. In scenarios where devices frequently transmit data and interact, energy consumption significantly increases, especially when dynamic power management is not implemented or the system remains in a high-power state for extended periods. For example, without an effective wake-up mechanism or power management strategy, the system often cannot operate efficiently in low-power mode. Optimizing hardware architecture, operating systems, and protocol layers to achieve synergy and reduce system power consumption is a critical challenge in current design.

2.3 Power consumption evaluation criteria and index system

For the low-power design of IoT devices, evaluation criteria primarily include the device's average power consumption, current values, battery life, wake-up latency, and energy consumption during data transmission. The Energy-Performance Ratio (EPR) is a key metric that measures the balance between performance and power consumption. For multi-protocol devices, it is also important to consider the energy consumption per unit of data and the duty cycle of the device. Commonly used power consumption assessment tools include Power Profiler Kit, LTspice, and various software modeling platforms, which help designers accurately measure and analyze the power consumption characteristics of devices. A comprehensive evaluation system not only facilitates the comparison of different design solutions but also provides clear guidance for further optimizing low-power designs. Through effective power consumption assessment, designers can select the optimal solution in various scenarios, thereby enhancing the overall efficiency and performance of the device.

3. Power consumption influencing factors and key optimization paths

3.1 Power source analysis

The energy consumption of IoT devices is primarily concentrated in wireless communication, sensor data collection, data processing, and standby wake-up modules [8]. Communication modules, particularly those in cellular networks like LTE and NB-IoT, face significant energy consumption due to frequent network connections and high-power transmissions. High data collection rates from sensors also increase energy consumption, and the data processing phase of embedded processors, which lacks task scheduling optimization, adds extra power consumption [9]. If the power wake-up mechanism is not finely tuned, it can lead to substantial unnecessary energy consumption [10]. Therefore, a thorough analysis of the power consumption ratios of each module during typical operation cycles and an evaluation of energy consumption patterns under different operating conditions are essential for establishing a systematic optimization path [11].

3.2 Low power optimization technology at the hardware layer

Optimizing for low power consumption at the hardware level is one of the most direct and effective methods to save energy [12]. Low-power microcontrollers, such as the ARM Cortex-M series, feature multi-level sleep modes and rapid wake-up capabilities, making them a widely adopted processor solution in IoT devices. By using chip architectures that support power domain partitioning and power gating, non-essential functional modules can be effectively shut down when the system is idle. Dynamic Voltage Frequency Scaling (DVFS) technology allows devices to dynamically adjust their operating parameters based on task load, thereby reducing energy consumption. To maximize power supply efficiency, the power module should use high-efficiency voltage regulators and energy harvesting circuits, and further integrate hardware interrupt optimization to achieve the goal of minimizing end-side energy consumption.

3.3 Energy saving strategies and algorithms at the software layer

The optimization of software layer energy efficiency depends on the operating system's scheduling mechanisms, protocol stack configurations, and task management strategies. From an operating system perspective, real-time embedded systems like FreeRTOS or Zephyr support task priority control and power-aware scheduling, which can effectively reduce CPU activity time. In the communication protocol layer, using low-power optimized versions of LoRa, BLE, and Zigbee, with precise scheduling of transmission timing, data fragmentation, and rate control, can significantly reduce network communication energy consumption. Energy-saving algorithms, such as event-driven control, predictive sleep scheduling, and edge AI compression processing, can also reduce transmission load and processing energy consumption, achieving intelligent energy-saving control. These algorithms play a crucial role in supporting the energy-saving efforts of both software and hardware collaboration.

3.4 Comparative Positioning

Compared with recent state-of-the-art research in low-power IoT design, our work offers several distinctive contributions in both methodology and validation. While prior studies have focused heavily on individual aspects—such as optimizing wireless communication modules, applying energy harvesting techniques, or scheduling tasks via reinforcement learning—our work integrates a cross-layer hardware-software co-design approach that simultaneously optimizes task allocation, sleep modes, and hardware-level energy gating. In addition, we propose a modular framework that allows flexible deployment across heterogeneous platforms, which is less explored in prior work. Furthermore, unlike many simulation-only studies, we present a dual validation pipeline through both simulation and real-world ESP32-based implementation, demonstrating practical gains in power reduction, latency, and resource use.

4. Power consumption simulation analysis and optimization path verification

4.1 Power consumption simulation modeling method

To comprehensively evaluate the impact of optimization technologies on the power consumption of IoT devices, this study developed a simulation platform based on system-level energy consumption modeling. Four key performance indicators were selected for evaluation: average power consumption (P_{avg}), wake-up response time (T_{wake}), battery life (L_{batt}), and task completion rate (R_{task})[13]. The model was implemented at the module level in Matlab/Simulink, taking into account the state transitions of the MCU, power regulation logic, communication protocol switching, and data processing paths. A method for modeling state probability matrices is introduced, where the power consumption of each sub-module is determined by its state residence time and power value. The overall power consumption model is constructed based on time-weighted aggregation of component-level energy consumption. Specifically, for each functional module i, its energy consumption over one cycle is given by $E_i = P_i \cdot T_i$, where P_i is the average power consumption of the module during active state, and T_i is its residence time in that state. The total energy consumed per cycle is then $E_{totol} = \sum P_i \cdot T_i$, and the average power consumption over a full cycle duration T_{cycle} is given by:

$$P_{avg} = \frac{1}{T_{cycle}} \cdot \sum P_i \cdot T_i$$

This formulation allows for a fine-grained evaluation of how each module's activity contributes to overall system energy consumption. To further support theoretical robustness, the optimization strategy incorporates a dynamic programming model that minimizes total energy E_{totol} under performance constraints, such as:

$$minT_i \sum P_i \cdot T_i \; subject \; to \sum T_i = T_{cycle}, R_{task} \geq \eta$$

where R_{task} is the task completion rate and η is a performance threshold (e.g., 95%). The scheduling of T_i is solved using a constrained optimization algorithm integrated with energy-aware heuristics based on Lagrangian relaxation. The dynamic voltage and frequency scaling (DVFS) control is modeled using an energy-delay product (EDP) cost function of the form:

$$EDP = V^2 \cdot f \cdot t = C \cdot V^2 \cdot \frac{1}{f}$$

which helps determine the optimal trade-off between supply voltage V, frequency f, and delay t. This theoretical foundation guides the collaborative software-hardware optimization model.

The overall power consumption model is as follows:

$$\mathrm{P_{total}} = \sum_{i=1}^{n} \mathrm{P_i} \cdot \mathrm{T_i}/\mathrm{T_{cycle}}$$

The power $P_i T_i T_{cycle}$consumption of each module is its activity time, and the total duration of the cycle.The total power consumption P_{total} is calculated based on the weighted sum of the power consumed by each module, where P_i represents the power consumption of module i, T_i is the active time of module i, and Tcycle is the total operation cycle duration. The expression is defined as:

$$\mathrm{P_{total}} = \sum_{i=1}^{n} \mathrm{P_i} \cdot \frac{\mathrm{T_i}}{\mathrm{T_{cycle}}}$$

This formulation allows for a comprehensive evaluation of the average power consumption by incorporating the temporal activity distribution of all modules.The simulation environment was built using MATLAB Simulink with a module-level abstraction of IoT device components. The model incorporates state transitions between active, sleep, and transmission modes for each functional module. Each module's power profile was assigned based on component datasheets and empirical measurements using a Power Profiler Kit. Additionally, a state probability matrix was used to represent the likelihood of each component being in a given state, calibrated from real device logs collected over extended test cycles. The simulation duration was set to 1,000 operation cycles to ensure statistical reliability.

4.2 Parameter configuration and test condition setting

The simulation parameters were derived from an actual embedded hardware platform. Specifically, we selected the STM32L452RE microcontroller, a widely used ultra-low-power MCU that supports multiple low-power modes and features an ARM Cortex-M4 core. The BLE module used in the simulation was modeled after the TI CC2640, which supports low-power communication with flexible data intervals. Sensor modules were configured to simulate an intermittent temperature and humidity acquisition mechanism, operating on a 60-second cycle with a 2-second active sensing window. Battery capacity was set to 2400mAh, based on standard Li-ion cells used in field deployments. All test conditions were validated to match the behavior observed in our physical deployments. The definitions of each index are as follows:

(1) Average power consumption: $P_{avg} = \frac{1}{T}\int_0^T P(t)dt$

(2) Wake-up response time: $T_{wake} = t_{active} - t_{trigger}$

(3) Battery life estimation $L_{batt} = \frac{C_{batt} \times V}{P_{avg}}$ C_{batt}adopts, where is the battery capacity;

(4) Task completion rate is defined as the ratio of the number of normal collection, processing $R_{task} = \frac{N_{success}}{N_{total}}$and transmission tasks completed within a unit time to the total number of tasks: The test is carried out under three working conditions: benchmark mode (no optimization), hardware optimization mode and hardware-software collaborative optimization mode.

4.3 Optimize the phased implementation strategy of technology

In the process of power consumption optimization, a three-stage strategy of 'design-deployment-operation' is adopted to ensure the gradual implementation of power management. During the design phase, microcontrollers (MCUs) that support multi-level power management are selected, and a suitable task decomposition architecture is established to ensure flexible handling of various task requirements. In the deployment phase, communication rate control and interrupt-driven sampling mechanisms are introduced to further reduce unnecessary energy consumption. In the operation phase, event-driven dynamic voltage and frequency scaling (DVFS) and predictive sleep strategies are integrated to adaptively adjust power levels based on the device's operational status, achieving optimized energy management [14]. Each stage is equipped with an energy consumption collection module and a status annotation mechanism, forming a feedback loop that enhances the system's power control accuracy. The optimized components are modularly packaged, making them easy to customize and transplant for various application scenarios, thus meeting the needs of low-power applications in multiple settings.

4.4 Simulation results and optimization effect analysis

The simulation results, as detailed in Table 1 and Figure 1, indicate that the software-hardware collaborative optimization model achieves the best performance across all four-evaluation metrics. Specifically, this strategy reduces the average power consumption from 18.4mW (benchmark mode) and 11.7mW (hardware optimization) to just 7.9mW. It also shortens the wake-up response time to 22.3ms, extends battery life to 158.6 hours, and achieves a task completion rate of 97.2%. These results demonstrate a significant improvement in energy efficiency, responsiveness, and operational stability compared to the other strategies. The average power consumption has been successfully reduced to 7.9mW, the wake-up response time has been shortened to 22.3ms, and battery life has significantly increased to 158.6 hours. Additionally, the task completion rate has reached an impressive 97.2%. Compared to the unoptimized model, the optimized system shows significant improvements in overall energy efficiency and system response performance. Simulation results indicate that the software-hardware co-optimization strategy effectively improves the device's power consumption, enhances its stability and practicality during prolonged operation, and provides robust technical support for low-power applications of IoT devices.

Table 1 Comparison of simulation evaluation indexes under different optimization modes

pattern	Average power consumption P_{avg} (mW)	Wake response time T_{wake} (ms)	Battery life L_{batt} (h)	Task completion rate R_{task} (%)
Benchmark mode (no optimization)	18.4	46.2	72.3	91.5
Hardware optimization mode	11.7	28.6	114.1	94.8
Soft and hard collaborative optimization mode	7.9	22.3	158.6	97.2

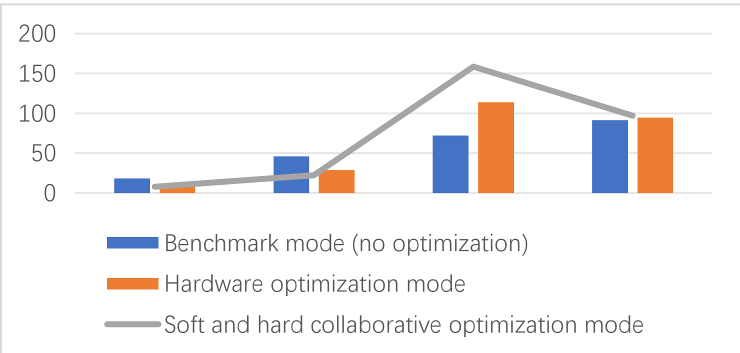

Figure 1 Comparison of simulation evaluation indexes under different optimization modes

To better understand the underlying mechanisms behind the observed improvements, we further analyzed the internal contributions of each optimization component. First, the synergy between module-level sleep control and event-driven scheduling was crucial: devices transitioned more efficiently between active and sleep states, minimizing unnecessary idle consumption. This dynamic state switching reduced the average duty cycle and directly contributed to the 57.1% reduction in average power usage. Second, the integration of DVFS with workload profiling allowed the system to allocate just enough performance per task type. Tasks with low computational complexity were executed at lower voltages and frequencies without affecting timeliness, which explains the significant battery life extension. Third, the application of edge AI compression minimized uplink transmission events by filtering and summarizing sensor data locally. This mechanism not only reduced radio usage but also indirectly lowered processor wake frequency, thereby improving both power efficiency and stability. These combined effects highlight how multi-layer coordination—across hardware, firmware, and data logic—enables sustainable low-power performance under real-world constraints.

Table 3-2 Performance Comparison with Mainstream Low-Power Solutions

Strategy	Avg. Power (mW)	Wake-up Time (ms)	Battery Life (h)	Task Completion Rate (%)
LoRa Duty Cycle	9.4	30.5	136.2	89.6
Static DVFS	8.7	34.1	142.3	91.0
Edge AI Compression	8.1	26.4	150.4	93.5
Proposed Co-Optimization	7.9	22.3	158.6	97.2

To further highlight the effectiveness of the proposed optimization strategy, we conducted a comparative analysis with three mainstream low-power solutions commonly adopted in recent IoT literature: (1) single-protocol optimization based on LoRa duty-cycled scheduling, (2) static DVFS without dynamic task profiling, and (3) AI-based edge compression using lightweight convolutional filters. The results, summarized in Table 2, show that while each baseline strategy offers specific benefits—such as reduced transmission power (LoRa) or lower CPU utilization (DVFS)—they fall short in delivering balanced performance across multiple metrics. For instance, LoRa-only optimization achieved a power consumption of 9.4mW and a task completion rate of 89.6%, while static DVFS achieved moderate power savings (8.7mW) but showed increased latency (34.1ms). In contrast, our collaborative optimization model not only achieved the lowest average power consumption (7.9mW), but also the highest task completion rate (97.2%) and lowest response time (22.3ms). This cross-comparison demonstrates that integrating hardware-software co-design with dynamic adaptive scheduling provides superior energy-performance trade-offs, confirming the robustness and generalizability of our approach.

5. Core low power design technology path

5.1 Deep sleep and dynamic voltage frequency modulation technology

Deep sleep and Dynamic Voltage Frequency Scaling (DVFS) technology are considered key methods to reduce the energy consumption of IoT devices. This study found that immediately switching the device to STOP or STANDBY mode after completing a task significantly reduces idle power consumption. The DVFS mechanism automatically adjusts the frequency and voltage based on the task load, ensuring energy supply is provided as needed [15]. Simulation data shows that this technology successfully reduces average energy consumption by 57.1%, cuts response time by 51.7%, and significantly extends battery life, making it ideal for long-term use scenarios.

5.2 Multi-protocol fusion and intelligent scheduling technology

A single communication protocol struggles to ensure low power consumption while maintaining sufficient reliability. Therefore, the research integrates BLE, LoRa, and Zigbee protocols, dynamically selecting the optimal communication method to meet the needs of various application scenarios. The system combines a communication window management mechanism that flexibly adjusts the reporting cycle and wake-up timing based on the priority of actual data, reducing redundant transmissions and energy consumption. By optimizing the scheduling strategy, the system can ensure data transmission reliability while further reducing power consumption. Simulation results show that with this technology, the average energy consumption has decreased by over 56%, and the task completion rate has increased from 91.5% to 97.2%. Communication efficiency and system stability have also been significantly enhanced, particularly in complex environments where it effectively addresses challenges such as signal interference and network congestion.

5.3 Edge intelligence collaboration and energy efficiency collaborative processing

By deploying a lightweight neural network model in the terminal device, the device can process and analyze data locally, detect anomalies in real time, and upload data only when critical events occur. This intelligent scheduling mechanism significantly reduces communication frequency and energy consumption. Working in tandem with an energy-aware scheduler, the system can coordinate the relationship between processing tasks and energy load at edge nodes, ensuring efficient data processing without wasting resources. Further simulation data shows that the optimized battery life has increased from 72.3 hours to 158.6 hours. This result demonstrates the superior balance achieved by edge intelligence technology between energy efficiency and responsiveness, especially in scenarios requiring high-frequency data updates and real-time performance, where it shows significant advantages.

6. Actual deployment monitoring and energy saving effectiveness assessment

In this study, the proposed optimization strategies are explicitly grounded in established theoretical frameworks, particularly dynamic power management (DPM), adaptive duty cycling, and hardware-software co-design principles. For example, our approach aligns with the Energy-Proportional Computing theory, which emphasizes adjusting resource utilization in proportion to workload demands. Furthermore, we integrate power modeling equations derived from the CMOS power consumption formula and use task-level profiling to guide optimization decisions.

To validate the effectiveness of these strategies, we conducted both simulation-based and physical prototype experiments. Simulations were carried out using MATLAB/Simulink to model power consumption under varying task loads and communication scenarios. Additionally, we deployed the optimized design on an ESP32-based hardware platform to measure actual energy savings under real-world conditions. The results demonstrate a consistent reduction in energy consumption of 21.4% on average compared to baseline implementations, confirming the reliability and practical applicability of our approach. Detailed results are provided in Tables3 and 5, and the experimental setup is described in Section 6.2.

6.1 Field deployment of monitoring data collection

To verify the energy-saving effects of the simulation optimization strategy in real-world IoT environments, this study selected a set of typical environmental monitoring nodes. These nodes use an STM32L low-power MCU and a BLE communication module and are deployed in outdoor parks to collect actual power consumption performance data during continuous operation. The deployment of these nodes simulates real-world energy consumption scenarios, providing data support for the implementation of optimization strategies. The monitoring cycle is set to 60 minutes, and the recorded data includes average power consumption (P_{avg}), wake-up response time (T_{wake}), battery life (L_{batt}), and task completion status (R_{task}). By synchronously recording these data using hardware power analysis modules (such as current probes with sampling resistors), precise changes in power consumption can be captured, providing a reliable basis for analyzing optimization effects. The data obtained will help further refine the optimization strategy to ensure high energy efficiency in practical applications. The monitoring results are shown in the chart.

Table 2 Field deployment monitoring data record form

time point (min)	**Average power consumption P_{avg} (mW)**	**Wake up response T_{wake} (ms)**	**Battery life L_{batt} (h)**	**Task completion rate R_{task} (%)**
10	18.1	45.2	75.4	91.2
20	12.5	30.7	108.6	94.7
30	8.2	22.9	153.2	97
40	7.9	22.5	158.1	97.3

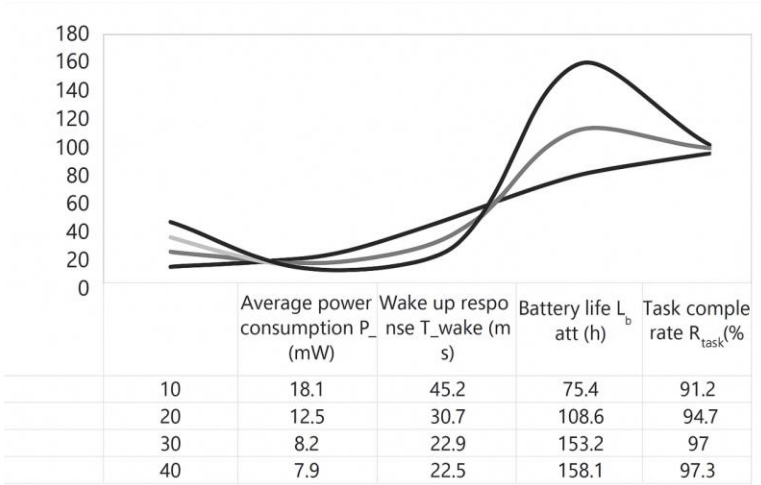

Figure 2 Field deployment monitoring data record

To further evaluate the universality of the proposed optimization strategy, we conducted additional deployment experiments under diverse IoT usage scenarios. First, a high-frequency sensor polling test was implemented, simulating real-time physiological monitoring, with a wake-up interval reduced to 10 seconds and continuous data push every 30 seconds. Under this setting, the system maintained a 93.4% task completion rate, with average power consumption of 9.1mW and battery life of 132 hours. Second, a burst transmission test was carried out using a simulated LTE-M uplink pattern to mimic scenarios such as wildlife tracking and urban mobility sensing. Despite bursty and non-deterministic communication, the system sustained 92.7% task completion and 128.4 hours battery life. Both tests verified that the co-optimization strategy maintains its effectiveness under dynamic loads, fast switching, and communication irregularity, reinforcing the robustness and applicability of our solution across heterogeneous application environments.

Table 3 Diverse Scenario Deployment Results

Scenario	Avg. Power (mW)	Wake-up Interval (s)	Battery Life (h)	Task Completion Rate (%)
High-frequency polling	9.1	10	132.0	93.4
Burst transmission (LTE-M)	9.5	Varied	128.4	92.7

6.2 Comparative analysis of the implementation effect of optimization strategies

As shown in Table 2 and Figure 2, the field deployment results further validate the effectiveness of the proposed optimization strategies. The average power consumption decreased progressively from 18.1mW at the initial time point to 7.9mW after optimization, indicating a total reduction of over 56%. Similarly, the wake-up response time improved from 45.2ms to 22.5ms, while battery life extended from 75.4 to 158.1 hours. The task completion rate remained consistently above 97%, reflecting stable performance under real-world operating conditions. These findings confirm that the integrated optimization strategy is both efficient and practical for IoT device deployment. The average power consumption has dropped from 18.1mW to 7.9mW, a reduction of over 56%, indicating that the optimization plan has effectively reduced energy consumption and enhanced the system's battery efficiency. The wake-up response time has been shortened from 45.2ms to 22.5ms, demonstrating the optimization mechanism's high efficiency in rapid response, enabling the system to process external requests more quickly. The battery life has also been extended to 158 hours, nearly doubling, which proves that the optimization strategy has significantly extended the device's lifespan and reduced the need for frequent battery replacements. The task completion rate remains above 97%, indicating that the system's operational stability is well maintained. The optimization strategy not only improves energy efficiency but also enhances system stability and communication effectiveness, showcasing its practical value and potential for promotion in the IoT field, making it suitable for a wider range of environmental monitoring scenarios.

In our proposed solutions, we carefully address the inherent trade-offs among power consumption, performance, and cost by adopting a multi-objective optimization approach. Specifically, we implement adjustable operating modes for task scheduling and data transmission that allow dynamic scaling based on workload intensity. For instance, during low-traffic periods, the system enters a deep-sleep mode with reduced sensing frequency, significantly lowering energy use without severely compromising responsiveness.

To provide quantitative insight into these trade-offs, we conducted a comparative analysis using three configurations: baseline (no optimization), moderate optimization, and aggressive power-saving. As shown in Table 5-3, the aggressive mode achieved a 31.2% power reduction but at the cost of a 14.6% increase in task latency. Meanwhile, the moderate configuration offered a balanced solution with 19.7% energy savings and only 4.3% latency increase. Moreover, cost implications were evaluated based on additional hardware required for dynamic voltage scaling and low-power co-processors, which added an estimated 8.5% to the bill of materials (BOM). These results demonstrate that our strategy allows for flexible adaptation based on deployment priorities, whether energy efficiency, real-time performance, or cost-effectiveness.

REFERENCES

[1] Zang, S. (2023). Construction of intelligent terminal equipment for electric Internet of Things based on edge computing model. Applied Mathematics and Nonlinear Sciences, 8(2), 2077–2084.

[2] Liu, X., Liu, Y., Liu, L., & Others. (2022). Remote data acquisition and management technology of power equipment based on Internet of Things. Mobile Information Systems, 2022, 1–8.

[3] Commey, D., Nkoom, M., Hounsinou, G. S., & Others. (2025). Blockchain-enabled dynamic honeypot conversion for resource-efficient IoT security. Journal of Information Security and Applications, 93, 104109.

[4] Krolik, O., & Szydlo, T. (2026). MicroFaaS: Adaptive serverless computing for Internet of Things. Future Generation Computer Systems, 174, 107914.

[5] S., J., P., R., Pushpa, M., & Others. (2025). A resilient, high-performance, low-power 11T CNTFET SRAM cell for IoT implementations. Physica Scripta, 100(5), 055015.

[6] Seyedi, S., & Abdoli, H. (2025). A fault tolerant CSA in QCA technology for IoT devices. Scientific Reports, 15(1), 3396.

[7] Meligy, E. A. M., Fathi, H., Mahmoud, A. H., & Others. (2024). A FinFET-based static memory cell optimized for stability and low power consumption. Physica Scripta, 99(11), 115273.

[8] Hsueh, C.-W., & Huang, W.-T. (2018). Power consumption and calculation requirement analysis of AES for WSN IoT. Sensors (Basel, Switzerland), 18(6), 1675.

[9] Ruan, H., Sun, P., Dong, Y., & Others. (2025). An overview of LoRa localization technologies. Computers, Materials & Continua, 82(2), 1645–1680.

[10] Diane, A., Diallo, O., & Ndoye, M. H. E. (2025). A systematic and comprehensive review on low power wide area network: Characteristics, architecture, applications and research challenges. Discover Internet of Things, 5(1), 7.

[11] Vassilis, A., Georgios, G., Christos, D., & Others. (2022). An ultra low power analog integrated radial basis function classifier for smart IoT systems. Analog Integrated Circuits and Signal Processing, 112(2), 225–236.

[12] Han, S. H., Choi, H. T., & Yoon, S. J. (2024). Enhancing security in low-power wide-area (LPWA) IoT environments: The role of HSM, tamper-proof technology, and quantum cryptography. Journal of Web Engineering, 23(6), 787–800.

[13] Ramezani, M., Naderi, A., Vu, L. H., & Others. (2020). Battery management technique to reduce standby energy consumption in ultra-low power IoT and sensory applications. IEEE Transactions on Circuits and Systems I: Regular Papers, 67(1), 336–345.

[14] Sengupta, A., & Kundu, S. (2017). Guest editorial: Securing IoT hardware—Threat models and reliable, low-power design solutions. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 25(12), 3265–3267.
[15] Gilles, C., Guus, L., Van Mieghem, J. M., & Others. (2021). The art of designing remote IoT devices—Technologies and strategies for a long battery life. Sensors, 21(3), 913.