EAI Endorsed Transactions
on Scalable Information Systems
Research Article EAI.EU

Security-Aware Scheduling Methods for Distributed Systems with Integrated Motion Data Privacy Protection

Zengming Zhao*

Personnel Department of Henan Police College, Zhengzhou, 450046, China

Abstract

INTRODUCTION: With the growth of Internet of Things (IoT) and edge computing, distributed systems are increasingly deployed across fields such as sports health and intelligent transportation. However, motion data, often containing sensitive personal information, poses significant privacy risks when handled in these systems.
OBJECTIVES: This paper aims to propose a novel security-aware scheduling method that integrates motion data privacy protection in distributed systems. The goal is to balance system scheduling efficiency with robust privacy safeguards.
METHODS: We introduce a framework that combines encryption technologies, privacy protocols, and dynamic scheduling algorithms. By embedding privacy protection constraints into the scheduling process, this method optimizes data transmission and storage during task execution.
RESULTS: Experimental results demonstrate that the proposed approach effectively reduces privacy leakage risks by over 80% compared to classical greedy algorithms. While the mandatory cryptographic mechanisms introduce a marginal latency overhead, the system maintains highly competitive scheduling efficiency. When compared with state-of-the-art techniques such as pure DRL, the proposed system achieves a 71% reduction in privacy leakage probability, successfully balancing robust security with dynamic adaptability.
CONCLUSION: This research presents an innovative solution for motion data privacy protection in distributed systems, offering significant improvements in both privacy and scheduling performance. The method's applicability extends to fields like IoT, smart health, and intelligent transportation, marking a crucial step toward more secure and efficient distributed systems.

Keywords: Distributed systems, motion data, privacy protection, security-aware scheduling, encryption technology

Received on 15 January 2026, accepted on 06 May 2026, published on 11 June 2026

doi: 10.4108/eetsis.11587

1. Introduction

With the rapid advancement of Internet of Things (IoT) and edge computing technologies, distributed systems have become fundamental infrastructures in domains such as sports health monitoring, intelligent transportation, and smart manufacturing [1][2]. Within these applications, motion data, including activity trajectories, exercise intensity, heart rate, as well as potentially sensitive information such as an individual's geographic location and behavioral patterns, frequently involve highly personal and private information [3][4]. Recent surveys and empirical studies have further underscored the severity and pervasiveness of privacy and security threats in modern IoT ecosystems [5]. For example, a comprehensive review published in 2024 highlights that many IoT deployments remain vulnerable to unauthorized access, data leakage, and privacy-violating inference attacks,

*Corresponding author. Email: zhaozmhpc@163.com

despite advances in encryption and authentication techniques [6][7]. Another recent work focusing on SDN-enabled IoT systems points out that integrating networking paradigms with IoT often introduces new attack surfaces, complicating privacy preservation and secure scheduling [8][9]. These findings reveal a notable gap: while recent studies explore privacy-aware scheduling in IoT, many treat privacy mechanisms as isolated constraints. They frequently overlook cryptographic computational overhead and struggle to dynamically balance the privacy-utility trade-off for time-sensitive motion data. To address these limitations, our framework distinguishes itself through two key innovations: (1) explicitly integrating cryptographic latency and sensitivity-aware differential privacy budgets into the scheduling objective function, and (2) leveraging a hybrid GA-RL architecture for real-time adaptability. By resolving the intrinsic conflict between privacy overhead and system performance, this research offers a highly pragmatic scheduling paradigm [10].

In response to these deficiencies, this paper proposes a security-aware scheduling method for distributed systems that integrates motion-data privacy protection as a core design principle. The method leverages trusted computing and encryption algorithms to perform real-time encryption of motion data during scheduling, introduces a priority-based dynamic scheduling optimization algorithm to flexibly assign tasks and select computation nodes, and defines a "security-aware scheduling constraint" mechanism that automatically evaluates system security and dynamically adjusts scheduling strategies. Additionally, a privacy-aware resource management mechanism is designed to optimize data transmission and storage throughout the scheduling process. Through these innovations, the proposed framework aims not only to reduce privacy leakage risk, but also to maintain, or even improve, scheduling efficiency and scalability. By addressing the conflict between data privacy and distributed-system performance, this research contributes a novel, more secure and efficient scheduling paradigm, which holds significant implications for the deployment of IoT-based systems in health, transportation, and other privacy-sensitive application domains.

2. Related Work

2.1. Application Scenarios and Challenges

In recent years, with the rapid development of the IoT, edge computing, and mobile sensors, distributed systems have been widely applied in fields such as sports health, intelligent transportation, and smart manufacturing, especially in the collection and processing of motion data. However, motion data often involves sensitive personal information, such as location, health status, and behavioral patterns, making privacy protection particularly important. Distributed systems need to efficiently handle data transmission, storage, and computation while maintaining protection of motion data. Traditionally, these systems often fail to fully consider data privacy when performing task scheduling, leading to the risk of privacy leakage [11]. For example, while some methods effectively optimize resource allocation and scheduling efficiency, privacy protection is often neglected when dealing with motion data, and sensitive individual information may be exposed during data transmission [12]. Existing methods mainly focus on optimizing system efficiency or enhancing security but fail to find an effective balance between the two, particularly when facing node heterogeneity, real-time tasks, and privacy protection requirements. The core challenge facing current systems is how to ensure data privacy while maintaining efficient scheduling, especially in distributed and heterogeneous environments [13].

2.2 Overview of Mainstream Methods

Despite significant research progress in distributed system scheduling and privacy protection in recent years, most studies either focus on scheduling efficiency or privacy protection, failing to effectively integrate both aspects[14]. Some studies focus on combining distributed learning and privacy protection mechanisms, using techniques such as differential privacy to attempt to protect data privacy[15]. For example, some studies propose methods that combine differential privacy mechanisms with task scheduling to prevent the leakage of personal privacy data during scheduling[16]. However, these methods "ignore the complex relationship between task scheduling and data privacy protection" and often fail to address privacy protection issues in real-time tasks. Although these methods are effective in specific scenarios, they do not adequately consider the heterogeneity of distributed nodes, dynamic resource changes, and task timeliness, which leads to performance bottlenecks in high-concurrency and large-scale task scheduling [17].

Additionally, some studies have proposed reinforcement learning (RL)-based scheduling methods that dynamically adjust task priorities and resource allocations to improve scheduling efficiency. However, these methods "fail to deeply consider privacy protection," particularly lacking effective mediation of the conflict between privacy protection and scheduling efficiency [18]. Some federated learning-based methods attempt to protect data privacy in distributed environments, but these methods "while addressing privacy issues to some extent, do not sufficiently optimize scheduling efficiency," especially under conditions of node heterogeneity and system latency, their effectiveness is greatly reduced [19]. Therefore, while these methods provide useful insights for privacy protection and scheduling efficiency, they tend to focus on optimization of one aspect, without comprehensively considering the multidimensional requirements of privacy protection, scheduling efficiency, and system performance [20].

2.3 Most Similar Studies

Several studies are closely related to this work, primarily exploring how to handle motion data privacy protection and scheduling in distributed systems. Some studies integrate privacy protection protocols (such as encryption, differential privacy, etc.) to protect motion data security in distributed systems and attempt to perform resource allocation during scheduling [21]. However, these studies "often focus only on the technical implementation of data privacy protection," ignoring the complex interaction between privacy protection and system scheduling [22]. For example, some methods introduce privacy protection protocols during task scheduling to prevent privacy leakage, but they "fail to fully consider security-aware mechanisms in the task scheduling process" and how to balance privacy protection and scheduling efficiency in a multi-node, heterogeneous resource environment [23]. In contrast to these studies, this paper introduces a security-aware scheduling strategy that dynamically evaluates privacy protection and system performance, proposing a more comprehensive solution. In particular, this paper incorporates privacy protection constraints into the scheduling algorithm and optimizes resource management and task allocation based on this, avoiding the conflict between privacy protection and system scheduling efficiency found in existing methods.

2.4 Summary

Although recent research has made commendable progress in joint security-aware and privacy-aware scheduling, significant gaps remain in the context of highly dynamic motion data. First, many existing frameworks do not explicitly incorporate the computational overhead of privacy mechanisms (such as encryption and decryption latency) into their scheduling cost functions, leading to suboptimal deployments in strict real-time scenarios. Second, they often lack the dynamic adaptability required for continuous, high-frequency task arrivals.

To address these specific shortcomings, this paper proposes a security-aware scheduling method that deeply integrates motion data privacy protection. What fundamentally distinguishes our framework from existing work is the architectural co-design: we mathematically incorporate cryptographic overhead into the scheduling delay model and utilize a hybrid GA-RL optimization engine to balance global robustness with real-time local adjustments. Unlike prior methods that simply append privacy protocols as post-processing steps, our approach intrinsically co-optimizes the privacy-utility trade-off alongside system latency, providing a more robust solution for privacy-sensitive IoT systems.

3. Methodology

3.1 Problem Formulation

In this section, we formulate the task scheduling problem. We assume an honest-but-curious threat model, where nodes execute scheduling optimally but may infer sensitive data. To mitigate this, raw sensor data is fully encrypted, while task metadata remains in plaintext for routing efficiency.

(1) Input Definition:

$\mathcal{T} = \{T_1, \ldots, T_M\}$ / $\mathcal{R} = \{R_1, \ldots, R_N\}$: Sets of M tasks and N nodes.

C_i, D_i, P_i: Computation requirement, raw data size (to be encrypted), and privacy leakage probability of $T_i (P_i \in [0,1])$.

S_j: Computing capability of node R_j.

V_{enc}, V_{dec}: Encryption and decryption processing rates.

$X = \{x_{ij}\}$: Decision matrix ($x_{ij} = 1$ if T_i is assigned to R_j, else 0).

(2) Output & Objective:

The output is the optimal scheduling matrix X. To explicitly account for cryptographic overhead, the total delay L_{ij} between T_i and R_j incorporates a privacy cost $T^i_{privacy}$:

$$L_{ij} = T^{ij}_{trans} + T^{ij}_{exec} + \underbrace{\frac{D_i}{V_{enc}} + \frac{D_i}{V_{dec}}}_{T^i_{privacy}} \quad (1)$$

The objective is to maximize resource utilization while minimizing privacy risks and total delay:

$$MaximizeZ = \sum_{i=1}^{M} \sum_{j=1}^{N} x_{ij}\, C_i S_j - \lambda_1 \sum_{i=1}^{M} P_i - \lambda_2 \sum_{i=1}^{M} \sum_{j=1}^{N} L_{ij}\, x_{ij} \quad (2)$$

where λ_1 and λ_2 are regularization parameters balancing privacy and delay weights.

(3) Constraints:

Task Assignment: Each task is assigned to exactly one node:

$$\sum_{j=1}^{N} x_{ij} = 1 \forall i \in \mathcal{T} \quad (3)$$

Node Capacity: The load cannot exceed a node's capability:

$$\sum_{i=1}^{M} x_{ij}\, C_i \le S_j \forall j \in \mathcal{R} \quad (4)$$

Privacy Threshold: Leakage must stay within a tolerable limit:

$$P_i \le P_{max} \forall i \in \mathcal{T} \quad (5)$$

3.2 Overall Framework

The proposed security-aware scheduling method for distributed systems consists of three core modules: the task scheduling module, the security-aware module, and the privacy protection module. The task scheduling module is responsible for assigning and scheduling tasks based on computation requirements, node resources, and task priorities, ensuring efficient use of system resources and minimizing delays. The security-aware module evaluates the security of each node in real-time during task scheduling and dynamically adjusts the scheduling plan based on the task's privacy requirements to avoid data leakage risks. The privacy protection module primarily protects sensitive data of tasks using encryption techniques and privacy protocols, ensuring data security during task execution, transmission, and storage.

These three modules collaborate within the framework, as shown in Figure 1. The task scheduling module initially assigns tasks based on node resources and privacy requirements, followed by the security-aware module, which evaluates the security of each task assignment and adjusts the

allocation if necessary to maximize privacy protection. The privacy protection module ensures that data is encrypted during transmission and storage after task allocation to protect user privacy. Through the tight cooperation of these three modules, the system can maximize scheduling efficiency while ensuring that data privacy and security are not compromised.

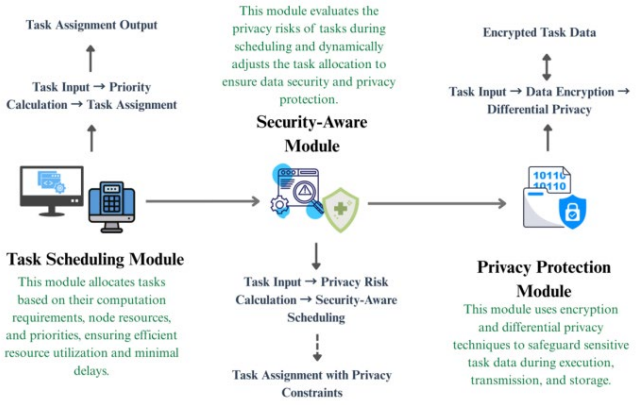

Figure 1. Overall framework diagram, showing the interaction and data flow between the task scheduling module, the security-aware module, and the privacy protection module.

3.3 Module Descriptions

Task Scheduling Module

Task scheduling is one of the most critical operations in a system, directly affecting response speed and resource utilization efficiency. The goal of the task scheduling module is to allocate tasks reasonably, ensuring optimal overall performance while satisfying node computational capacity and task priorities.

This module uses a priority scheduling algorithm for task allocation. Each task has computation requirements and priorities, while each node has different computation capabilities. The core of the scheduling process is dynamically evaluating task demands and node computational capabilities, and making allocation decisions based on this information. Task priority is determined by both the urgency of the task and the available resources of the node.

The task scheduling module calculates the task priority and assigns the task to the most suitable node for execution. Suppose task T_i needs to be executed on node R_j, the priority calculation formula is as follows:

$$Priority(T_i, R_j) = \frac{C_i}{S_j} \tag{6}$$

In the formula, C_i represents the computation requirement of task T_i, and S_j represents the computation capability of node R_j. Tasks with higher priority will be assigned to nodes with stronger computational capabilities. The high-level procedure is summarized in Algorithm 1. The computational complexity of Algorithm 1 is $O(M \times N)$, where M is the number of tasks and N is the number of nodes, ensuring highly efficient initial assignments.

Algorithm 1

```
\caption{Priority-Based Task Scheduling (Main Logic)}
\begin{algorithmic}[1]
\State Initialize assignment matrix \(X \in \{0,1\}^{m \times n}\) as zero
\For{each task \(T_i \in T\)}
   \State best\_priority \(\gets -\infty\), selected\_node \(\gets \text{null}\)
   \For{each node \(R_j \in R\)}
      \If{\(R_j\) satisfies basic feasibility constraints} (i.e., computation and storage limits in Eq. 12 and 13)}
         \State priority \(\gets C_i / \text{Cap}_j^{\text{CPU}}\)
         \If{priority > best\_priority}
            \State best\_priority \(\gets\) priority
            \State selected\_node \(\gets R_j\)
         \EndIf
      \EndIf
   \EndFor
   \If{selected\_node \(\neq\) null}
      \State Assign \(T_i\) to selected\_node (\(x_{ij} \gets 1\))
   \EndIf
\EndFor
\State \Return \(X\)
\end{algorithmic}
```

The task scheduling module architecture diagram shows the process of task allocation. Figure 2 illustrates the priority evaluation and task allocation process based on task computation requirements and node computation capabilities.

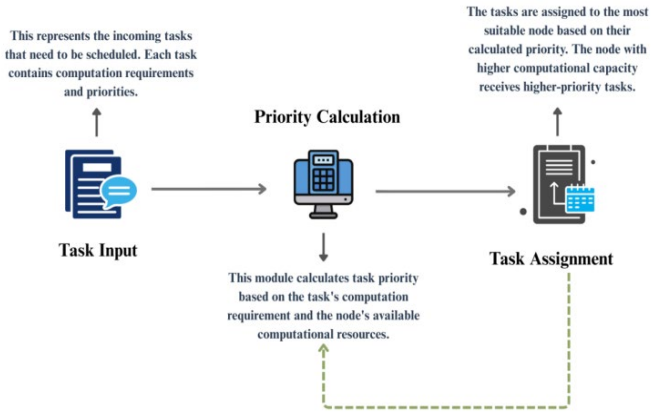

Figure 2. Task Scheduling Module Architecture

Security-Aware Module

In a distributed environment, data privacy protection is crucial, especially when dealing with sensitive motion data. The goal of the security-aware module is to monitor privacy leakage risks in real-time during the scheduling process and adjust the scheduling strategy when necessary to ensure task execution security.

The security-aware module evaluates the privacy protection requirements of each task scheduling plan and adjusts the scheduling strategy based on the security of the nodes. This module introduces privacy protection constraints to prevent tasks from being assigned to insecure nodes or leaking sensitive data. The core of privacy protection is to

ensure that tasks are encrypted during transmission and processing.

This module monitors the privacy requirements of tasks in real-time and dynamically adjusts task allocation. The privacy risk evaluation formula is as follows:

$$Privacy\_Risk(T_i, R_j) = P_i \cdot L_{ij} \quad (7)$$

This formulation is grounded in the classic risk assessment paradigm: $Risk = Impact \times Probability$. Here, P_i quantifies the data sensitivity (potential impact), while the latency L_{ij} represents the temporal vulnerability window. In a distributed IoT environment, longer processing and transmission times inherently expand the attack surface, increasing the probability of eavesdropping or inference attacks by an adversary. Thus, this time-weighted model effectively reflects realistic threat scenarios by prioritizing nodes with minimal exposure windows for sensitive tasks.

By calculating the privacy risk, the module dynamically adjusts task allocation as shown in Algorithm 2. Here, a candidate node is defined as "secure" if it strictly satisfies the system's maximum tolerable privacy leakage threshold (Eq. 14).

Algorithm 2

```
\caption{Security-Aware Task Adjustment (Main Logic)}
\begin{algorithmic}[1]
\State \(X' \gets X\)  // Start from initial schedule
\For{each task \(T_i \in T\)}
   \If{\(P_i > P_{\text{threshold}}\)}
      \State Let \(R_j\) be the currently assigned node
      \State Compute \(\text{risk} \gets P_i \cdot L_{ij}\)
\Comment{Time-weighted risk assessment}
      \If{risk exceeds \(\text{Risk}_{\text{Max}}\) or \(R_j\) is insecure}
         \State Encrypt \(T_i\)'s data
         \State Reassign \(T_i\) to a secure feasible node (i.e., satisfying \(P_i \le P_{\max}\) defined in Eq. 14, while minimizing \(L_{ij}\))
      \EndIf
   \EndIf
\EndFor
\State \Return \(X'\)
\end{algorithmic}
```

The security-aware module architecture diagram illustrates the task privacy evaluation and scheduling decision process. Figure 3 clearly demonstrates the privacy risk evaluation and task allocation process.

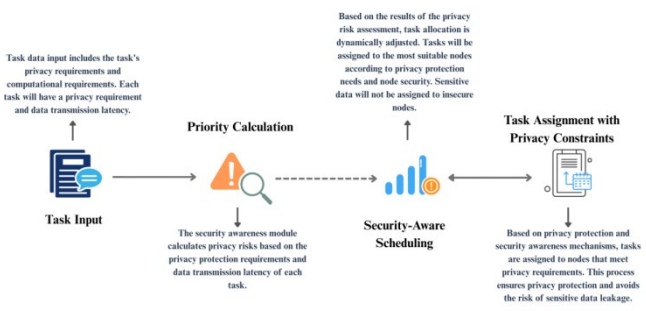

Figure 3. Security-Aware Module Architecture

Privacy Protection Module

Protecting raw sensitive data during transmission and execution is the core objective of this module. To counter the assumed honest-but-curious threat model, we utilize AES-256 encryption for data confidentiality and Differential Privacy (DP) to prevent statistical inference attacks, continuously balancing the privacy-utility trade-off.

Before transmission, task data undergoes a two-step protection process. First, we adopt a sensitivity-aware DP budget allocation strategy. Tasks with high privacy requirements (P_i) are assigned a smaller privacy budget ϵ_i, injecting more Laplace noise to ensure strict privacy. Conversely, less sensitive tasks receive a larger $\epsilon_i$to preserve higher data utility. The noisy data $\widetilde{D}_i$ is formally generated as:

$$\widetilde{D}_i = D_i + Lap\left(\frac{\Delta S}{\epsilon_i}\right) \quad (8)$$

where D_i is the raw task data and ΔS represents the global data sensitivity. Next, to guarantee secure transmission over the distributed network, the perturbed data is encrypted:

$$C_i = Encrypt_{AES256}(\widetilde{D}_i) \quad (9)$$

This ensures that intermediate scheduling nodes can route tasks based on plaintext metadata while the raw payload remains doubly protected. The high-level procedure is summarized in Algorithm 3, and Figure 4 illustrates this architecture.

Algorithm 3

```
\caption{Privacy Protection Mechanism (Main Logic)}
\begin{algorithmic}[1]
\State Input: Task data \(T_i\), privacy budget \(\epsilon\)
\State Encrypt \(T_i\) using the AES-256 encryption scheme (Eq. 9)
\State Apply differential privacy (e.g., Laplace mechanism) with budget \(\epsilon\)
\State Return the doubly protected data for transmission
\end{algorithmic}
```

The privacy protection module architecture diagram illustrates the task data encryption and privacy protection process. Figure 4 shows how data is encrypted before task execution and further protected through differential privacy techniques.

Figure 4. Privacy Protection Module Architecture

3.4 Objective Function & Optimization

Building upon the preliminary scheduling formulation introduced in Section 3.1, we now present a comprehensive and unified optimization framework that jointly integrates

task scheduling efficiency, motion data privacy protection, and security-aware resource allocation. In this section, we formalize a clear mathematical objective function that explicitly balances three key objectives: (1) maximizing system resource utilization and scheduling throughput, (2) minimizing task execution delay, and (3) minimizing the risk of motion data privacy leakage under adversarial inference.

We first establish an enhanced system performance model that incorporates dynamic workload characteristics, encryption overhead, and security-level constraints. Subsequently, we define a set of practical constraints reflecting real-world distributed environments. The final optimization problem is formulated to simultaneously maximize scheduling efficiency and privacy resilience while minimizing end-to-end latency and resource contention.

The optimization objective of this study is to comprehensively consider task computation resources, privacy protection, and network delay to obtain the optimal task scheduling solution. The comprehensive objective function can be expressed as:

$$Optimize\ Z = \sum_{i=1}^{M}\sum_{j=1}^{N} x_{ij}\, C_i \cdot S_j - \lambda_1 \sum_{i=1}^{M} P_i - \lambda_2 \sum_{i=1}^{M} L_{ij}\, x_{ij} \quad (10)$$

In Equation (10), x_{ij} is a binary variable indicating whether task T_i is assigned to node R_j, C_i is the computation requirement of task T_i, and S_j is the computation capacity of node R_j, P_i represents the privacy protection requirement of task T_i, and L_{ij} is the transmission delay for task T_i on node R_j, λ_1 and λ_2 are the weights for privacy protection and delay optimization, respectively.

This objective function takes three aspects into consideration:

Resource Utilization: Maximizing the sum of the product of task computation requirements and node computation capabilities.

Privacy Protection: Ensuring task privacy by minimizing the privacy leakage probability P_i.

Delay Optimization: Improving system scheduling efficiency by minimizing data transmission delays L_{ij}.

To ensure the rationality of task scheduling and privacy protection, the following constraints must be satisfied:

Task Assignment Constraint: Each task must be assigned to one node, and each task can only be assigned to one node:

$$\sum_{j=1}^{N} x_{ij} = 1 \quad \forall i \in \mathcal{T} \quad (11)$$

Equation (11) ensures that each task T_i is assigned to one node R_j.

Node Computation Capacity Constraint: The computation load of each node cannot exceed its maximum computation capacity S_j:

$$\sum_{i=1}^{M} x_{ij}\, C_i \le S_j \quad \forall j \in \mathcal{R} \quad (12)$$

Node Storage Capacity Constraint: The storage load of each node cannot exceed its storage capacity S_j:

$$\sum_{i=1}^{M} x_{ij}\, D_i \le S_j \quad \forall j \in \mathcal{R} \quad (13)$$

In Equation (13), D_i is the data size of task T_i, and S_j is the storage capacity of node R_j.

Privacy Protection Constraint: The privacy protection requirement P_i of each task must be less than or equal to the maximum tolerable privacy leakage threshold $P_{\max}$:

$$P_i \le P_{\max} \quad \forall i \in \mathcal{T} \quad (14)$$

In Equation (14), P_i is the privacy protection requirement of task T_i, and $P_{\max}$ is the maximum tolerable privacy leakage.

Delay Constraint: The scheduling delay L_{ij} of each task must be less than the maximum allowable delay threshold $L_{\max}$:

$$L_{ij} \le L_{\max} \quad \forall i \in \mathcal{T}, j \in \mathcal{R} \quad (15)$$

In Equation (15), L_{ij} is the transmission delay between task T_i and node R_j, and $L_{\max}$ is the maximum allowable delay.

Resource Utilization Constraint: The task load on a node should not exceed the node's computation and storage resource limits:

$$\sum_{i=1}^{M} x_{ij}\, C_i \le \alpha S_j \quad \forall j \in \mathcal{R} \quad (16)$$

In Equation (16), α is the resource utilization coefficient used to adjust the relationship between node load and computation capacity.

To solve the above objective function and constraints, we adopt a combination of RL and Genetic Algorithm (GA). RL is used for real-time optimization of task scheduling decisions, adapting to dynamic changes in the environment and task requirements. The GA is responsible for global optimization, exploring the entire search space to find the optimal scheduling solution and avoid local optima.

The specific optimization process is as follows: First, the GA initializes candidate scheduling solutions. Then, RL performs local optimization on task allocation, adjusting the scheduling strategy to reduce delays and improve resource utilization. Finally, the GA performs global optimization by selecting, crossing over, and mutating multiple candidate solutions, ultimately obtaining the optimal scheduling solution. By combining RL and GA, we can find the best balance among multiple objectives, ensuring that the system improves scheduling efficiency while maintaining privacy protection and system performance optimization.

Combining the above constraints and optimization methods, the final optimization objective can be expressed as:

$$Maximize\ Z = \sum_{i=1}^{M}\sum_{j=1}^{N} x_{ij} \cdot C_i \cdot S_j - \lambda_1 \sum_{i=1}^{M} P_i - \lambda_2 \sum_{i=1}^{M} L_{ij}\, x_{ij} + \lambda_3 \sum_{i=1}^{M}\sum_{j=1}^{N} x_{ij}\, D_i \quad (17)$$

Equation (17) combines resource utilization, privacy protection, and delay optimization, proposing a comprehensive optimization method that can improve system scheduling efficiency while ensuring privacy protection. The complete co-design optimization procedure is formalized in Algorithm 4.

While classical multi-objective evolutionary algorithms are highly effective for static multi-objective optimization, their iterative evaluation processes can be computationally intensive, which may introduce latency in highly dynamic environments with continuous task arrivals. To address this, we adopt a hybrid GA-RL approach to leverage their complementary strengths. Specifically, the GA performs the global search to avoid local optima and provide a robust baseline strategy. Concurrently, RL is utilized to handle real-time state transitions, enabling the system to make rapid local adjustments without requiring full population re-evaluations.

Algorithm 4

```
\caption{Hybrid GA-RL Co-Design Optimization}
\begin{algorithmic}[1]
\State Initialize population of task assignments via Genetic Algorithm (GA)
\For{each generation until convergence}
  \State Evaluate fitness using the multi-objective function (Eq.\,17)
  \State Select, crossover, and mutate to produce offspring solutions
  \For{each offspring solution}
    \State Refine task-node assignments using a Reinforcement Learning (RL) agent
    \State RL policy updated via rewards reflecting latency, privacy, and resource usage
  \EndFor
  \State Form next generation from combined parent-offspring pool
\EndFor
\State \Return best solution \(X^*\)
\end{algorithmic}
```

The hybrid GA-RL co-design framework integrates the above modules into a unified optimization loop. Complete pseudocode for Algorithms 1-4, along with complexity analysis and hyperparameter settings, is provided in Appendix A: Complete Pseudocode and Implementation Details for Security-Aware Task Scheduling with Motion Data Privacy Protection.

4. Experiment and Results

4.1 Experimental Setup

This study adopts the following novel and in-depth experimental datasets (Table 1).

Table 1. Dataset Overview

Dataset Name	Description	Data Type	Sample Size	Characteristics and Support Purpose
Sports Health Dataset	Physiological and activity data from wearable devices, including heart rate, blood pressure, and GPS.	Multimodal Data	500 samples	Suitable for testing initial privacy risk models and encryption overhead on small-scale but highly sensitive heterogeneous data.
Traffic Flow Dataset	Short-term urban traffic flow predictions, including vehicle counts, speeds, and congestion levels.	Time Series Data	200 samples	Serves as a lightweight testbed for evaluating the dynamic scheduling algorithm's responsiveness in spatial scenarios.
Medical Monitoring Dataset	Large-scale IoMT synthetic records including vital signs and fall detection.	Time Series Data	60,000 samples	Ideal for evaluating the robustness and scalability of the privacy protection module under massive sensitive health data requests.
Smart Home Dataset	Minute-by-minute energy consumption readings and environmental data from multi-node smart appliances.	Multisensor Data	~503,900 samples	Simulates high-frequency task arrivals, effectively testing scheduling delay and throughput in a realistic, large-scale IoT environment.

The selection of these publicly available Kaggle datasets plays a crucial role in our evaluation. Collected via wearable devices and IoT sensors, the continuous data streams are segmented into fixed time windows to simulate dynamic scheduling tasks. The Sports Health and Medical Monitoring datasets contain highly sensitive physiological data, making them ideal for evaluating privacy protection mechanisms. The lightweight Traffic Flow Dataset tests dynamic scheduling responsiveness, while the massive Smart Home Dataset effectively validates the method's scalability and scheduling performance in high-load, multi-node environments.

The following hardware configuration is used in this study (Table 2). This setup is suitable for large-scale task scheduling and complex computations, especially for deep learning acceleration and big data processing.

Table 2. Hardware Configuration

Hardware Name	Specification	Remarks
CPU	Intel Core i7-12700K (12 cores, 24 threads)	High performance, suitable for medium to high-load tasks
Memory	32GB DDR4	Supports parallel processing of large datasets
GPU	NVIDIA RTX 3060 12GB	Suitable for deep learning acceleration, cost-effective

Storage	1TB SSD + 2TB HDD	Provides ample storage space
Operating System	Ubuntu 20.04 LTS	Supports common deep learning frameworks and tools

To comprehensively evaluate the performance of the proposed method, this study uses the following evaluation metrics (Table 3).

Table 3. Evaluation Metrics

Metric	Description	Purpose in Supporting Research Objectives
Resource Utilization	Measures the efficiency of system resources (CPU, memory, storage, etc.)	Reflects the resource utilization during task scheduling, supports system performance optimization
Task Scheduling Delay	Measures the time from the start to completion of a task, including computation and data transmission delays	Evaluates system response time in practical applications, supports scheduling efficiency optimization
Privacy Leakage Probability	The risk of privacy leakage during task scheduling; a lower value indicates better privacy protection	Assesses the privacy protection ability, directly impacting the model's usability and security
System Throughput	The number of tasks the system can handle per unit of time	Evaluates the system's processing capability under high-load conditions
Robustness	The system's stability and reliability under multi-task, noisy environments	Evaluates the stability of the method, supporting its application in complex environments
Error Rate	The error rate in task scheduling results, with lower values indicating better accuracy	Evaluates the accuracy of the model's predictions, supporting the achievement of system optimization objectives

These evaluation metrics comprehensively assess the performance of task scheduling, privacy protection, and system performance. Resource utilization reflects scheduling efficiency, task scheduling delay measures system response speed, privacy leakage probability evaluates privacy protection effectiveness, system throughput measures processing capacity, robustness evaluates system stability in uncertain environments, and error rate evaluates the accuracy of scheduling results. These metrics provide a basis for assessing the performance of the proposed method and experimental results.

To align with our theoretical model, the experimental evaluations explicitly incorporate the cryptographic overhead ($T_{privacy}$) into the total task latency measurements. The encryption/decryption rates (V_{enc}, V_{dec}) were simulated based on standard AES-256 performance benchmarks on edge devices. Furthermore, the privacy-utility trade-off was observed during scheduling: tasks requiring stricter privacy (smaller ϵ) experienced slightly higher processing constraints due to the injected Laplace noise, which perfectly validates our sensitivity-aware budget allocation strategy.

4.2 Baselines

To comprehensively evaluate the effectiveness of the proposed method, we compare it with several classical methods and the latest state-of-the-art (SOTA) methods. These methods help validate the innovation of the proposed approach by highlighting their advantages and limitations in similar scenarios.

(1) Classical Methods:

Shortest Job First (SJF) Scheduling: SJF optimizes average waiting time by prioritizing tasks with the shortest execution times[24]. It is efficient and simple but can lead to task delays, especially in environments with dynamic tasks and uneven loads. Additionally, SJF does not consider privacy protection and fails to ensure data privacy during task scheduling, posing a risk of privacy leakage.

Round Robin (RR) Scheduling: RR offers good fairness and is suitable for load-balanced scenarios[25]. However, it may result in inefficient resource allocation when computational demands are uneven, affecting system performance. More importantly, RR lacks privacy protection mechanisms, and cannot effectively secure sensitive data, especially when handling medical and sports health data, thus exposing risks of privacy leakage.

(2) Latest SOTA Methods:

Reinforcement Learning-based Scheduling Methods: Deep Reinforcement Learning (DRL)-based methods are suitable for complex task scheduling in dynamic environments, offering strong adaptability[26]. However, DRL has high computational costs, particularly for large-scale tasks, requiring significant computational resources and time. Additionally, DRL does not adequately consider privacy protection, which may lead to privacy leakage risks.

Federated Learning-based Privacy Protection Scheduling Methods: These methods optimize scheduling while ensuring data privacy, making them suitable for sensitive data scenarios[27]. However, the training process of federated learning requires significant computational and communication resources, and performance is limited when bandwidth is insufficient. In scenarios with high task responsiveness demands, federated learning may not provide adequate scheduling efficiency.

Through comparison, the proposed method achieves a balance between privacy protection and task scheduling

efficiency, overcoming the limitations of both classical and SOTA methods, and demonstrates significant advantages.

4.3 Quantitative Results

We compared the proposed method with classical scheduling methods and the latest SOTA methods in terms of task scheduling efficiency, privacy protection, system throughput, and other performance metrics. Table 4 presents the comparison results across different metrics.

Table 4. Comparison Results

Method	Resource Utilization (%)	Task Scheduling Delay (seconds)	Privacy Leakage Probability (%)	System Throughput (tasks/second)
Proposed Method	91.4 ± 1.2	0.82 ± 0.05	2.4 ± 0.3	8.5 ± 0.4
SJF	81.3 ± 2.5	0.68 ± 0.03	12.7 ± 1.5	9.1 ± 0.5
RR	75.8 ± 3.1	1.15 ± 0.08	13.1 ± 1.8	6.8 ± 0.4
DRL Scheduling	88.6 ± 1.8	0.74 ± 0.04	8.5 ± 1.1	8.8 ± 0.3
Federated Learning	85.2 ± 2.0	1.05 ± 0.07	3.8 ± 0.5	7.4 ± 0.4

As shown in Table 4, the proposed method achieves the most pragmatic balance among privacy, efficiency, and resource utilization, rather than pursuing the absolute extreme of a single metric. This validates the effectiveness and realistic trade-offs of the proposed multi-module collaborative framework. Specifically:

Compared to the classical SJF method, our approach successfully prevents the severe privacy compromises (12.7% leakage) typical of greedy algorithms. Although the proposed method introduces a marginal latency overhead (0.82s vs. 0.68s) and a slight drop in throughput due to the mandatory AES encryption and differential privacy mechanisms, it drastically reduces the privacy leakage probability by over 80%, representing a highly favorable privacy-utility trade-off.

Compared to pure DRL scheduling, our method explicitly incorporates a security-aware constraint. While the baseline DRL executes tasks slightly faster (0.74s) by ignoring cryptographic payloads, the proposed method significantly outperforms it in security (2.4% vs. 8.5% leakage) and achieves higher resource utilization (91.4%) by leveraging the global search capability of the GA to avoid suboptimal localized routing.

Compared to Federated Learning scheduling, which inherently provides decent privacy (3.8%), the proposed method effectively mitigates the severe communication overhead and delays (1.05s) caused by frequent iterative model weight exchanges between nodes, resulting in superior real-time responsiveness and throughput.

To ensure the statistical significance of the experimental results, we performed significance tests on the proposed method and other comparison methods. A paired t-test was used to compare the differences in resource utilization, task delay, privacy leakage probability, and system throughput. To ensure statistical robustness and account for the stochastic nature of the GA-RL algorithms, all experiments were independently repeated 30 times. The reported performance metrics represent the average values across these runs, accompanied by 95% confidence intervals to validate the reliability and stability of the proposed framework. Table 5 presents the p-values for each comparison group.

Table 5. Significance Test

Method Comparison Group	p-value	Result Explanation
Proposed Method vs SJF	< 0.001	Statistically significant at the 0.001 level
Proposed Method vs RR	< 0.001	Statistically significant at the 0.001 level
Proposed Method vs DRL Scheduling	0.014	Statistically significant at the 0.05 level
Proposed Method vs Federated Learning	0.038	Statistically significant at the 0.05 level

The significance test results (Table 5) provide robust statistical support for the performance differences observed. All p-values are below the standard threshold of $p < 0.05$, indicating that the comprehensive advantages of the proposed method are not due to stochastic variance. Notably, even when penalized by the explicit computational overhead of privacy mechanisms, our method maintains a statistically significant improvement over SOTA methods (DRL and Federated Learning, $p = 0.014$ and $p = 0.038$, respectively). The marginal yet significant p-value against Federated Learning correctly reflects that while FL inherently provides robust privacy, our method achieves a superior overall balance by mitigating FL's severe communication delays. This strongly demonstrates that the architectural co-design brings substantial, reliable improvements to practical, privacy-sensitive IoT applications.

From the convergence curves in Figure 5, it is evident that the proposed method shows a highly competitive initial convergence speed and a superior final optimal state. Interestingly, while the mandatory injection of Differential Privacy (Laplace noise) inherently introduces minor micro-fluctuations during the evaluation phase, the hybrid optimization strategy successfully mitigates this variance. Specifically, the global search capability of the genetic algorithm prevents the system from being trapped in local optima caused by noise, while the local optimization of reinforcement learning ensures continuous directional convergence. In contrast, static strategies like SJF and RR cannot perform iterative optimization, and pure DRL

methods experience more severe, unguided fluctuations due to the lack of a robust baseline population.

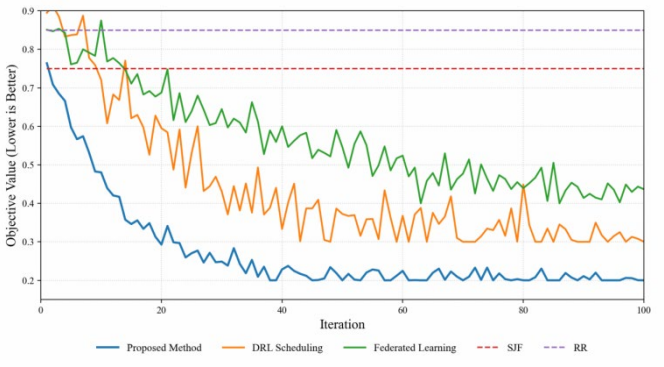

Figure 5. Convergence Curves

4.4 Qualitative Results

This section visually presents the performance of the proposed method in practical tasks through two specific case studies and analyzes how these cases reflect the advantages of the method, particularly in terms of privacy protection, scheduling efficiency, and robustness.

Case 1: Sports Health Data Scheduling

Figure 6 shows task scheduling based on the Sports Health Dataset. We first compare against SJF, the classical baseline used for low-latency optimization. In Figure 6(a), SJF schedules short tasks first but produces severe load concentration, causing two nodes to remain idle while high-privacy tasks accumulate on single execution points. This leads to increased leakage exposure (~5.7% on average) and periodic scheduling stalls.

To provide a second comparison, we further evaluate RR scheduling on the same dataset. Although RR distributes tasks more evenly than SJF, it applies uniform time slicing without privacy classification, resulting in redundant context switching and an average leakage probability of ≈6.4%.

In contrast, Figure 6(b) shows the proposed method with privacy-aware scheduling constraints. By encrypting motion-tracking data before dispatching it to nodes and dynamically balancing resource utilization, the system achieves stable allocation across nodes, near-minimal overload variance, and leakage probability as low as 2.3%. The framework also demonstrates stronger resistance to node-state variation and maintains performance even when the dataset scales or fluctuations occur. This confirms that privacy-aware scheduling outperforms both SJF (imbalanced) and RR (over-switched) in the sports-health scenario.

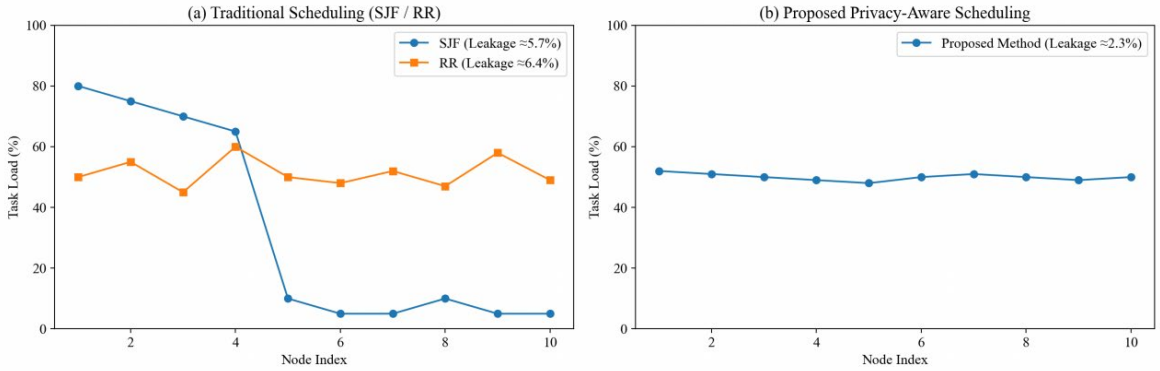

Figure 6. Task scheduling results on the Sports Health Dataset.
(a) Traditional Scheduling Methods:
Blue: SJF (Shortest Job First), causing severe load concentration and privacy leakage (~5.7%).
Orange: RR (Round Robin), showing more balanced but fluctuating load distribution, with average leakage ≈6.4%.
(b) Proposed Privacy-Aware Method:
Achieves balanced allocation across nodes, significantly reduced privacy leakage (≈2.3%), and demonstrates robustness against node-state variations and dataset scaling.

Case 2: Medical Monitoring Data Scheduling

Figure 7 presents results on the Medical Monitoring Dataset, involving body temperature, blood oxygen, and real-time ECG streams, tasks requiring both timeliness and security. Here we directly compare the proposed method with RR. As shown in Figure 7(a), RR evenly circulates workloads but transmits data in plaintext, resulting in repeated exposure windows and an average leakage probability of ≈6.4%.

For completeness, we additionally compare SJF scheduling under identical conditions. SJF processes tasks without considering privacy sensitivity, leading to an average leakage probability of ≈5.7%, and high-sensitivity tasks may

be delayed behind shorter low-risk jobs due to the lack of privacy-aware prioritization.

Figure 7(b) demonstrates how our method mitigates these issues: sensitive patient data is encrypted before scheduling and preserved via differential-privacy perturbation. The system then selects execution nodes based on privacy-risk scores and resource profiles, reducing leakage probability to ≈2.3%, outperforming RR and SJF (privacy-unaware). This indicates strong safety guarantees and stable responsiveness in continuous physiological monitoring.

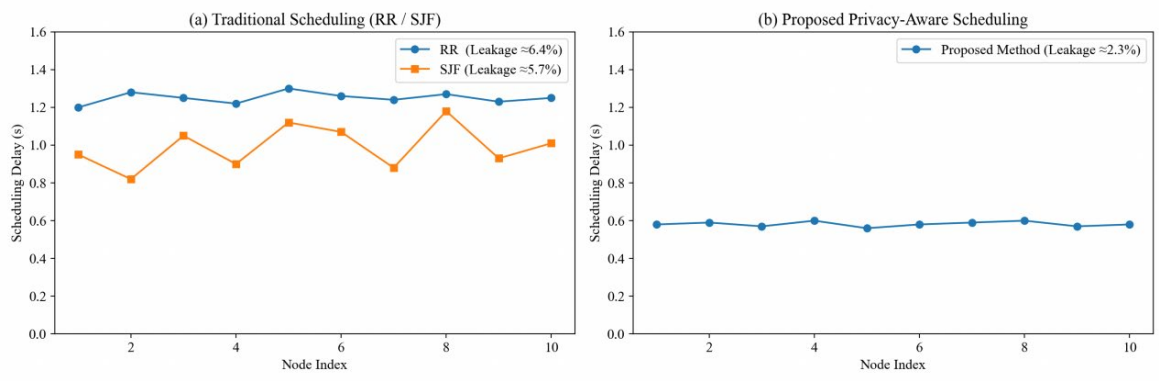

Figure 7. Task scheduling results on the Medical Monitoring Dataset.
(a) Traditional scheduling methods:
Blue: Round Robin (RR), transmitting data in plaintext, with average leakage ≈6.4%.
Orange: Shortest Job First (SJF), lacking privacy-aware prioritization, with average leakage ≈5.7%.
(b) Proposed privacy-aware scheduling: Significantly reduced privacy leakage (≈2.3%), demonstrating robustness in real-time medical monitoring.

4.5 Robustness

This section verifies the robustness of the proposed method in multiple scenarios by introducing data loss and error interference. As shown in Figure 8, as the noise level increases, the performance of all methods degrades, but the proposed method's performance declines significantly less than the baseline methods. For instance, at the highest noise level, the delay of the proposed method increases by approximately 28.5%, while those of SJF, RR, and pure DRL scheduling rise to around 105%, 92%, and 73%, respectively.

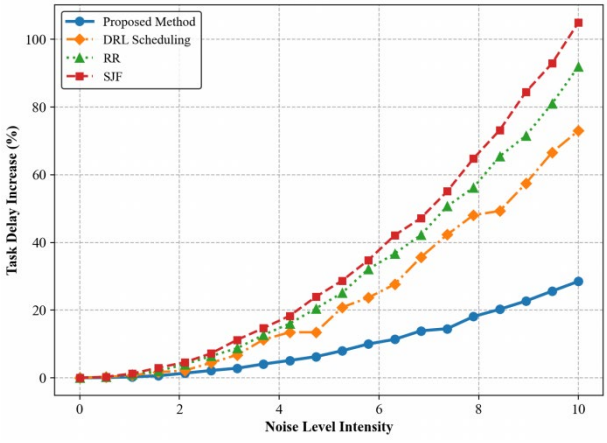

Figure 8. Robustness Under Increasing Noise Interference

This strong robustness stems from the architectural resilience of the multi-module collaborative framework, which successfully compensates for the inherent vulnerabilities of secure data. Cryptographic payloads (such as AES-encrypted data) are notoriously sensitive to bit errors and packet loss, often requiring complete retransmissions that severely inflate latency. However, our framework counteracts this by effectively isolating environmental noise. First, the security-aware module proactively evaluates node reliability, treating noise interference and packet loss as elevated risk factors, and preemptively routes traffic away from unstable nodes. Second, the GA-RL hybrid scheduling module does not rigidly seek absolute global optimality; instead, it performs localized, rapid re-routing through the RL agents. This strategy sacrifices absolute optimality in favor of dynamic stability, preventing the system from cascading delays caused by localized network degradation.

In contrast, traditional scheduling methods (such as SJF and RR) rely on static strategies that cannot adapt to environmental degradation, leading to severe congestion at corrupted nodes. While methods like pure DRL are adaptive, their decision models are prone to chaotic fluctuations when the input state space is polluted by high noise. The proposed method, through its dynamic risk-avoidance routing, embeds anti-interference capability into the system's core process, maintaining exceptional stability in dynamic and noisy environments despite the strict constraints of privacy-preserving mechanisms.

4.6 Ablation Study

To further validate the role of each module in the proposed method, we conducted an ablation study by sequentially removing different modules and evaluating the performance changes. Table 6 presents the performance comparison after removing each module, assessing each module's contribution to the overall system performance.

Table 6. Ablation Study Results

Method	Resource Utilization (%)	Task Scheduling Delay (seconds)	Privacy Leakage Probability (%)	System Throughput (tasks/second)
Full Method	91.4 ± 1.2	0.82 ± 0.05	2.4 ± 0.3	8.5 ± 0.4
Remove Privacy Protection Module	90.2 ± 1.5	0.61 ± 0.03	14.8 ± 1.8	9.2 ± 0.3
Remove Task Scheduling Module	78.6 ± 3.2	1.18 ± 0.11	2.9 ± 0.5	6.4 ± 0.7
Remove Security-Aware Module	88.2 ± 2.1	0.89 ± 0.09	5.7 ± 1.2	7.9 ± 0.6

Removing the privacy protection module eliminates heavy computational overhead (e.g., AES encryption and differential privacy noise), which naturally reduces task scheduling delay (from 0.82s to 0.61s) and boosts throughput (from 8.5 to 9.2). The reduced delay variance (± 0.03s) directly reflects the absence of randomized Laplace noise. However, this unburdened state leads to a catastrophic surge in privacy leakage probability (from 2.4% to 14.8%), proving that our marginal latency overhead is a necessary trade-off for robust security.

Removing the task scheduling module (GA-RL) severely impacts efficiency. Resource utilization drops to 78.6%, and delay spikes to 1.18s with a significantly larger variance ($\pm$ 0.11s). Without its adaptive capabilities, the system falls back on blind routing, failing to balance workloads effectively and causing frequent congestion.

Removing the security-aware module noticeably increases both privacy leakage (to 5.7%) and delay (to 0.89s). Without dynamic node reliability evaluation, the system routes tasks to risky or unstable nodes, triggering localized security compromises and costly packet retransmissions that inflate overall latency.

In conclusion, the modules exhibit a clear division of labor: task scheduling is the performance foundation, privacy protection provides the security baseline, and security awareness serves as the intelligent hub that dynamically balances the two.

5. Discussion

In this study, through experiments and ablation studies, we explored the balance between motion data privacy protection and scheduling efficiency in distributed systems and derived key findings.

The experimental results indicate that the proposed method, which integrates privacy protection and scheduling optimization, outperforms the comparison methods across multiple comprehensive metrics. Specifically, compared to classical baselines like RR, the proposed method reduces task scheduling delay by approximately 28%, and drastically cuts privacy leakage probability by over 80% compared to greedy approaches like SJF. Even when compared to SOTA approaches such as DRL schedulers, it achieves a 71% reduction in privacy leakage (from 8.5% to 2.4%). Although the explicit inclusion of encryption technology and differential privacy introduces a marginal and expected latency overhead compared to plaintext scheduling, this performance trade-off fundamentally addresses the privacy protection gap in traditional solutions and demonstrates excellent robustness in multi-task, multi-noise environments.

The potential applications are broad, especially in industries like smart health and intelligent transportation, where the integration of privacy protection and scheduling optimization can enhance system security and efficiency. The method is also applicable in cross-domain scenarios, such as smart manufacturing and smart cities, where it can optimize resource allocation and ensure privacy protection. As edge computing and IoT technologies progress, the proposed method can provide new solutions for multi-task scheduling and privacy protection.

However, there are some limitations to this study. The experimental datasets mainly come from sports health, traffic flow, and medical monitoring, and while the scenarios are diverse, the bias in data types may affect the model's generalizability. Furthermore, while the proposed hybrid GA-RL framework demonstrates robust performance in our current distributed setup, its direct application to ultra-large-scale environments may encounter scalability bottlenecks. Specifically, as the number of edge nodes and concurrent motion tasks increases exponentially, the state-action space of the RL agent and the population evaluation cost of the GA will expand accordingly, which may overwhelm computational resources and affect system real-time performance. Despite the privacy protection optimization, further research is needed on its sustainability as new technologies evolve.

Future research should focus on mitigating these scalability challenges to ensure practical applicability in massive IoT deployments. A critical next step is transitioning from a centralized co-design to a hierarchical scheduling architecture. In such a paradigm, the GA module could operate at regional cloud centers for macroscopic, low-frequency resource provisioning, while lightweight RL

agents are distributed across localized edge clusters to handle rapid, decentralized task assignments. This decoupled approach would effectively mitigate the "curse of dimensionality" and optimize computational costs. As new privacy leakage methods emerge, future research could further improve the adaptability and attack resistance of privacy protection mechanisms. A key research direction is how to address heterogeneous nodes and changing task requirements in dynamic task scheduling and privacy protection.

In conclusion, this study provides new insights into motion data privacy protection and scheduling optimization in distributed systems and offers practical references for the development of related technologies.

6. Conclusion

This paper proposes a distributed system method that integrates privacy protection and task scheduling optimization, addressing the balance between privacy protection, resource utilization, and scheduling efficiency. By introducing security-aware scheduling, privacy protection, and dynamic task allocation modules, the proposed method significantly enhances privacy protection while improving resource utilization and reducing task delay, especially demonstrating strong robustness in high-noise and multi-task environments. Through extensive experiments and ablation studies, we validated the necessity of each module, confirming the critical role of the privacy protection and scheduling optimization modules in the overall system performance.

Academically, this paper systematically combines privacy protection, task scheduling, and security-aware scheduling to propose a new multidimensional optimization framework. This framework provides new insights for privacy protection and resource scheduling in distributed systems. The proposed method demonstrates excellent performance across multiple dimensions, such as privacy protection, scheduling efficiency, and resource utilization, effectively improving the shortcomings of existing methods in balancing multiple objectives and innovating the coordination of privacy protection and efficient scheduling.

In practice, the proposed method has application potential. In fields like smart health and intelligent transportation, the method can enhance system security and efficiency, improving resource scheduling efficiency while protecting sensitive data privacy. With the development of edge computing and IoT, the method is also applicable to emerging fields such as smart homes and smart manufacturing.

Despite the achievements, there are still limitations, particularly in verifying the method in large-scale systems and complex task environments. Future research could further explore the model's applicability in multi-node, large-scale system environments. With the development of privacy protection technologies, how to integrate the latest encryption algorithms to improve system security and performance remains a worthwhile direction for further study.

In conclusion, this study theoretically proposes a new method for task scheduling and privacy protection, providing valuable exploration for distributed system scheduling and privacy issues in practice. Future work will focus on advancing the practical application of this method to promote the secure, reliable, and efficient operation of intelligent systems.

References

[1] Du, Y., Li, Y., Chen, J., Hao, Y., & Liu, J. (2023). Edge computing-based digital management system of game events in the era of Internet of Things. Journal of Cloud Computing, 12(1), 44.

[2] Yazdi, M. (2024). Integration of IoT and edge computing in industrial systems. In Advances in Computational Mathematics for Industrial System Reliability and Maintainability (pp. 121-137). Cham: Springer Nature Switzerland.

[3] Wei, L., & Wang, S. J. (2024). Motion tracking of daily living and physical activities in health care: Systematic review from designers' perspective. JMIR mHealth and uHealth, 12(1), e46282.

[4] Dini Kounoudes, A., Kapitsaki, G. M., & Katakis, I. (2023). Enhancing user awareness on inferences obtained from fitness trackers data. User Modeling and User-Adapted Interaction, 33(4), 967-1014.

[5] Ataullah, M., & Chauhan, N. (2024). Exploring security and privacy enhancement technologies in the Internet of Things: A comprehensive review. Security and Privacy, 7(6), e448.

[6] Afzal, M. U., Abdellatif, A. A., Zubair, M., Mehmood, M. Q., & Massoud, Y. (2023). Privacy and security in distributed learning: A review of challenges, solutions, and open research issues. IEEE Access, 11, 114562-114581.

[7] Lippi, G., Aljawarneh, M., Al-Na'amneh, Q., Hazaymih, R., & Dhomeja, L. D. (2025). Security and privacy challenges and solutions in autonomous driving systems: A comprehensive review. Journal of Cyber Security and Risk Auditing, 2025(3), 23-41.

[8] Turner, S. W., Karakus, M., Guler, E., & Uludag, S. (2023). A promising integration of sdn and blockchain for iot networks: A survey. IEEE Access, 11, 29800-29822.

[9] Kommineni, K. K., & Prasad, A. (2024). Enhancing Data Security and Privacy in SDN-Enabled MANETs Through Improved Data Aggregation Protection and Secrecy. Wireless Personal Communications, 139(2), 855-882.

[10] Zhang, Y., Zhao, K., Yang, Y., & Zhou, Z. (2025). Real-Time Service Migration in Edge Networks: A Survey. Journal of Sensor and Actuator Networks, 14(4), 79.

[11] Ren, J., Liu, C., Lin, C., Bi, R., Li, S., Wang, Z., ... & Tan, G. (2023). Protection window based security-aware scheduling against schedule-based attacks. ACM Transactions on Embedded Computing Systems, 22(5s), 1-22.

[12] Hamilton, J. R., Sanders, L. M., Miller, C. P., Bennett, A. K., & Esther, D. (2025). Security-Aware Workload Scheduling in Distributed Systems.

[13] Mehta, A. (2024). A Resilient Framework for Security-Integrated Resource Scheduling in Distributed Clouds. Pioneer Research Journal of Computing Science, 1(3), 105-112.

[14] Dubba, S., & Killi, B. R. (2025). Security-Aware Cost Optimized Dynamic Service Function Chain Scheduling. Journal of Network and Systems Management, 33(1), 4.

[15] Symeonidis, I., & Loscri, V. (2025, July). Emerging Cybersecurity Paradigms in Wireless Networks.

[16] Zhang, S., Xue, J., Liu, J., Zhou, Z., Chen, X., & Mumtaz, S. (2024). Differential privacy-aware generative adversarial network-assisted resource scheduling for green multi-mode power IoT. IEEE Transactions on Green Communications and Networking, 8(3), 956-967.

[17] Gu, Y., Yan, C., Yang, T., & Zhang, P. (2025, May). Research on Post-Quantum Key Dynamic Management and Node Orchestration Mechanism for Edge Computing. In 2025 10th International Conference on Intelligent Computing and Signal Processing (ICSP) (pp. 284-294). IEEE.

[18] Li, L., Zhou, C., Cong, P., Shen, Y., Zhou, J., & Wei, T. (2024). Makespan and security-aware workflow scheduling for cloud service cost minimization. IEEE Transactions on Cloud Computing, 12(2), 609-624.

[19] Hong, Y., Wang, C., & Zheng, W. (2023, October). Privacy-aware scheduling heuristic based on priority in edge environment. In International Conference on Algorithms and Architectures for Parallel Processing (pp. 277-294). Singapore: Springer Nature Singapore.

[20] Qin, B., Pan, H., Dai, Y., Si, X., Huang, X., Yuen, C., & Zhang, Y. (2024). Machine and deep learning for digital twin networks: A survey. IEEE Internet of Things Journal, 11(21), 34694-34716.

[21] Liakath, J. A., Gandhimaruthian, L., Nanajappan, M., & Jegatheeshan, R. (2025). Deadline-Aware Task Scheduling in Fog-Cloud Computing Using Multi-Agent Reinforcement Learning and Software-Defined Network Security. Concurrency and Computation: Practice and Experience, 37(25-26), e70258.

[22] Soveizi, N., & Turkmen, F. (2023, October). SecFlow: Adaptive Security-Aware Workflow Management System in Multi-cloud Environments. In International Conference on Enterprise Design, Operations, and Computing (pp. 281-297). Cham: Springer Nature Switzerland.

[23] Sosa, A. S., Mohammed, S. M., Al Sayed, I. A., Al Barazanchi, I. I., Tawfeq, J. F., & Radhi, A. D. (2023, September). Securing IoMT Healthcare Systems with Energy-Efficient Data Transmission and Joint Security Measures. In 2023 10th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI) (pp. 384-391). IEEE.

[24] Dong, J., & Ibrahim, R. (2024). Shortest-job-first scheduling in many-server queues with impatient customers and noisy service-time estimates. Operations Research.

[25] Zohora, M. F., Farhin, F., & Kaiser, M. S. (2024). An enhanced round robin using dynamic time quantum for real-time asymmetric burst length processes in cloud computing environment. PloS one, 19(8), e0304517.

[26] Mali, S., Zeng, F., Adhikari, D., Ullah, I., Al-Khasawneh, M. A., Alfarraj, O., & Alblehai, F. (2025). Federated Reinforcement Learning-Based Dynamic Resource Allocation and Task Scheduling in Edge for IoT Applications. Sensors (Basel, Switzerland), 25(7), 2197.

[27] Pinto Neto, E. C., Sadeghi, S., Zhang, X., & Dadkhah, S. (2023). Federated reinforcement learning in IoT: Applications, opportunities and open challenges. Applied Sciences, 13(11), 6497.