mathematics

MDPI

Article

Privacy-Preserving Incentive Allocation for Fair and Resilient Data Sharing in Resource-Constrained Edge Computing Networks

Yanfang Wang [1], Shaobo Li [1,2,3,*], Kangkun Chen [3], Ran Guo [3] and Judy Li [3]

[1] State Key Laboratory of Public Big Data, Guizhou University, Guiyang 550025, China; gs.yanfangwang22@gzu.edu.cn
[2] Guizhou Institute of Technology, Guiyang 550003, China
[3] School of Computer Science and Engineering, Central South University, Changsha 410083, China; 8208221512@csu.edu.cn (K.C.); 8208221321@csu.edu.cn (R.G.); 8208240701@csu.edu.cn (J.L.)
* Correspondence: lishaobo@gzu.edu.cn

Abstract: Efficient and secure data sharing is paramount for advancing modern digital ecosystems, especially within edge computing environments characterized by resource-constrained nodes and dynamic network topologies. In such settings, privacy preservation, computational efficiency, and system resilience are critical for user engagement and overall system performance. However, existing approaches face three primary challenges: (i) limited optimization of privacy protection and absence of dynamic privacy budget scheduling for resource-constrained scenarios, (ii) static incentive mechanisms that overlook individual differences in data quality and resource consumption, and (iii) inadequate strategies to ensure resilience in environments with limited resources and unstable networks. This paper introduces the Federated Learning-based Dynamic Incentive Allocation Framework (FL-DIAF) to address these issues. FL-DIAF integrates differential privacy into the federated learning paradigm deployed on edge nodes, enabling collaborative model training that safeguards individual data privacy while maintaining computational efficiency and system resilience. Additionally, the framework employs a Shapley value-based dynamic incentive allocation model to ensure equitable and transparent distribution of incentives by accurately quantifying each participant's contribution within an elastic edge computing infrastructure. Comprehensive experimental evaluations on diverse datasets demonstrate that FL-DIAF achieves a 9.573% reduction in the objective function value under typical conditions and attains a 100% task completion rate across all tested resilient edge scenarios.

Keywords: federated learning; dynamic incentive allocation; privacy protection; resource optimization; Shapley value

MSC: 68P27; 68T05; 90C26; 37N40

check for updates

Academic Editor: Jonathan Blackledge

Received: 25 December 2024
Revised: 20 January 2025
Accepted: 25 January 2025
Published: 27 January 2025

Citation: Wang, Y.; Li, S.; Chen, K.; Guo, R.; Li, J. Privacy-Preserving Incentive Allocation for Fair and Resilient Data Sharing in Resource-Constrained Edge Computing Networks. *Mathematics* **2025**, *13*, 422. https://doi.org/10.3390/math13030422

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Data sharing, which is the process of sharing and using data among individuals, organizations, or systems, has become more and more important for promoting technological innovation and collaborative work, especially in edge computing. With the rise of big data and artificial intelligence, data sharing is a fundamental backbone for various applications, especially those deployed on distributed edge nodes. Edge computing leverages decentralized Internet technologies to distribute tasks or projects to a diverse group of participants,

facilitating the completion of these tasks through collective effort and enhancing system resilience [1].

It is essential to implement effective and suitable incentive mechanisms to prompt data owners to share their data and boost user engagement in edge computing networks. First of all, a strong incentive mechanism can draw in a larger user base and heighten platform activity by motivating participation across distributed edge nodes. By offering direct monetary rewards or points, edge computing platforms motivate users to invest their time and computational resources. Secondly, introducing tiered rewards can incentivize participants to improve the quality of their task completions [2–4]. Given the substantial variability in the data quality provided by individual users, including the quality of user contributions in the incentive mechanism encourages participants to engage more carefully. This, in turn, improves the overall quality of the output and ensures system flexibility [5,6]. Lastly, incentive mechanisms enable platforms to collect behavioral data and user feedback, facilitating the continuous optimization of task designs and fostering a virtuous improvement cycle [7].

Figure 1 illustrates the comprehensive flow of an edge-based data-sharing platform, highlighting the design and role of the incentive mechanism, strategies for enhancing task completion quality, existing challenges, optimization solutions, and the logical framework of the closed-loop optimization system. Specifically, the incentive mechanism optimizes the platform's activities by increasing user participation, improving data quality, and gathering behavioral feedback. Concurrently, the design of incentives encounters two primary challenges: the diversification of incentives and the decay of long-term incentives, both of which constrain the platform's long-term growth potential in dynamic edge environments. By establishing a closed-loop optimization system, platforms can build a sustainable cycle that combines user engagement, improvement of task quality, and continuous optimization. In this way, the system's resilience and flexibility can be ensured.

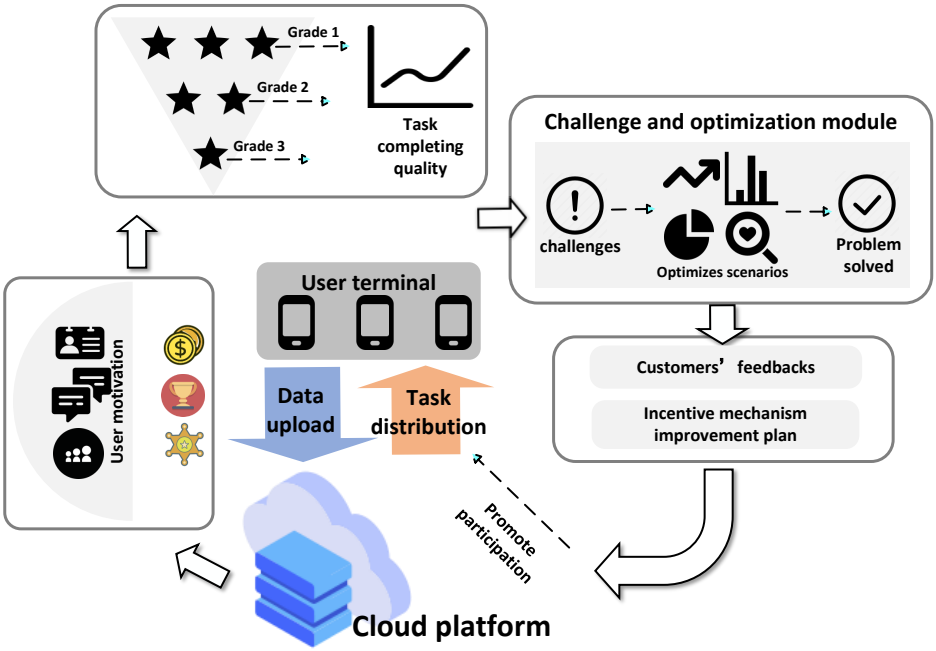


Figure 1. Framework of technical routes and ideas for data-sharing incentives in edge computing.

Although existing incentive mechanisms effectively motivate user participation in data sharing across various domains, such as federated learning [8,9] and crowdsourcing spectrum sensing, challenges remain in designing mechanisms that attract and sustain long-term user engagement within edge computing environments. A primary challenge lies in the wide-ranging motivations of users. These motivations consist of financial incentives,

social recognition, and personal interests. As a result, it becomes arduous for general-purpose incentives to tackle multiple motivational factors concurrently. Additionally, shifts in user interests and needs over time can lead to decreased participation, rendering existing incentive structures less effective in maintaining system elasticity and resilience.

Much of the current research focuses on developing dynamic incentive models to address the challenge of attracting and maintaining user participation in edge computing networks. These models monitor user participation behavior and feedback in real time, allowing for the adjustment of incentive strategies to align with evolving user motivations and needs [10,11]. Dynamic incentive mechanisms are generally more effective than static ones in sustaining continuous user engagement, as they can adapt to changes in user motivations and prevent user attrition due to evolving interests, thereby enhancing the system's resilience and elasticity.

However, dynamic incentive models must balance privacy protection with their effectiveness in enhancing user participation within distributed edge nodes. Developing dynamic incentive strategies typically requires collecting and analyzing extensive user data, which can raise concerns about privacy breaches and diminish users' willingness to participate [12]. Techniques such as differential privacy can be employed to mitigate these privacy risks. While differential privacy enhances data security, it may also reduce data accuracy, potentially undermining the incentive mechanism's effectiveness [13,14]. Thus, there is a need for further exploration to balance privacy safeguards with the effectiveness of incentive mechanisms, ensuring that user privacy protection does not compromise the mechanism's ability to maintain long-term user motivation and system resilience.

This study addresses significant challenges in resource-constrained edge computing environments by proposing the Federated Learning and Dynamic Incentive Allocation Framework (FL-DIAF). The framework optimally balances privacy protection, incentive fairness, computational efficiency, and system resilience. Fundamentally, this research presents a scalable, transparent, and privacy-safeguarding solution crafted to boost user participation and system performance in distributed data-sharing networks. In comparison with current solutions, the proposed approach shows better performance in terms of fairness, efficiency, and user engagement. The main contributions of this paper are as follows:

1. **Federated Learning (FL)**: We propose a multi-party collaborative training framework where participants can jointly train machine learning models without uploading raw data to a central server. This approach ensures that sensitive data remains decentralized, thus reducing the risk of privacy leakage. By leveraging federated averaging (FedAvg) and incorporating local training updates, the framework effectively mitigates the need for data sharing while maintaining high model accuracy and ensuring data confidentiality. This method is particularly beneficial in resource-constrained environments, as it allows each participant to only transmit model updates instead of full datasets.
2. **Dynamic Incentive Allocation**: To enhance fairness in resource-constrained edge computing environments, we integrate a dynamic incentive allocation mechanism that adapts to varying user contributions and data privacy requirements. Unlike traditional static mechanisms (e.g., fixed or auction-based incentives), our model uses Shapley values to measure the individual contribution of each participant based on the quality of their data, computational effort, and resource consumption. Additionally, the incentive distribution is adjusted dynamically based on each node's privacy requirements, ensuring that both high privacy and low privacy demands are considered, offering a balanced trade-off between efficiency and fairness.

3. **Algorithm with Fairness and Privacy Mechanisms**: Our framework harmoniously combines differential privacy (DP) with the incentive allocation strategy to ensure both data privacy and fair resource distribution. By introducing a dynamic privacy budget, we adjust the level of noise added to the model updates according to the participant's privacy needs, enabling a flexible privacy mechanism without compromising overall system efficiency. This privacy-preserving approach allows participants to control their data exposure, while the incentive allocation ensures that participants are fairly compensated for their contributions, even in a heterogeneous environment with varying privacy demands.
4. **Resource Constraints and Optimization Strategy**: Recognizing the resource limitations of edge nodes (e.g., computational power, storage, and communication bandwidth), we design an optimization framework that operates effectively under such constraints. The framework incorporates adaptive task offloading strategies and resource allocation models that optimize the use of local resources while minimizing network overhead. Additionally, we propose a game-theoretic model that facilitates fair resource sharing among participants, ensuring that no single node is overburdened and that the overall system remains resilient to fluctuating network conditions and resource constraints.

2. Related Work

In recent years, incentivizing data owners to participate in data sharing while safeguarding privacy has emerged as a critical research area. Current studies can be broadly categorized into four domains: privacy-preserving incentives, group-aware incentives, mobile group perception incentives, and privacy data processing and distribution. Despite the various approaches to tackling these challenges, practical applications still confront several constraints. Primarily, achieving an optimal balance between data privacy and sharing incentives remains unresolved. Additionally, existing incentives lack fairness and transparency, and computational efficiency in resource-constrained mobile environments is not adequately addressed. This section categorizes and synthesizes the prevailing research approaches, evaluates their strengths and weaknesses, and identifies future research directions.

2.1. Privacy-Preserving Incentives

Privacy-preserving incentive mechanisms aim to ensure data privacy while sharing through differential privacy and data scrambling techniques. Despite their effectiveness in reducing privacy leakage and enhancing data-sharing willingness [15–22], these methods often incur high computational costs and face privacy leakage risks in large-scale data sharing. For instance, Miao et al. [15] introduces a lightweight privacy-preserving truth discovery framework that employs differential privacy to protect data integrity, while Andrés et al. [19] presents a geographic differential privacy scheme to secure location data sharing without revealing specific user locations. Similarly, Zhang et al. [6] leverages state-dependent noise to perturb data, effectively mitigating the risk of sensitive information leakage. Furthermore, Gan et al. [21] and Wang et al. [5] propose dual privacy protection mechanisms based on federated learning, ensuring data security in multi-party scenarios by localizing data processing. Wang et al. [23] propose an efficient homomorphic encryption-based secure search scheme in a multi-owner setting. Feng et al. [24] design and build Panther, a lightweight and efficient secure 2P-NN inference system, which has great efficiency in evaluating 2P-NN inference while safeguarding the privacy of the server and the client. Although these methods have shown their efficacy in numerous real-world applications, the constantly growing volume of data presents a challenge regarding computational resource consumption. Improving computational efficiency and the applicability

of differential privacy algorithms in large-scale settings are still crucial areas for future research. In addition, Xuan et al. [25] proposes a blockchain-based data-sharing incentive model using evolutionary game theory. This aims to attract more data owners to participate in the FL process through a goal-first approach while effectively preventing information leakage.

2.2. Group-Aware Incentives

While privacy-preserving mechanisms address data leakage concerns, incentive fairness and effectiveness issues persist. To mitigate these problems, researchers have developed group-aware incentive models based on auction mechanisms and game-theoretic frameworks. These models not only ensure equitable incentive distribution but also enhance data quality [26–31]. For example, Feng et al. [29] proposes the TRAC auction mechanism, which incorporates location awareness to dynamically evaluate and reward data contributions. Similarly, the BidGuard framework introduced by Lin et al. [32] balances privacy protection and incentive fairness by minimizing social costs and reducing resource consumption among participants. Additionally, Lin et al. [31] develops an auction-based privacy-preserving mechanism tailored for mobile group perception. Yan et al. [28] employs a game-theoretic model to optimize incentive allocation strategies, thereby improving data quality and privacy protection. These studies demonstrate that well-designed incentive mechanisms can significantly improve fairness and participant motivation. However, challenges in enhancing these mechanisms' transparency and dynamic adaptability, particularly in large-scale group-aware environments, warrant further in-depth investigation. He et al. [33] introduces a hierarchical federated learning incentive mechanism using contract theory to address data volume, quality, and cost in an information asymmetry scenario, improving user participation and maximizing model owners' utility.

2.3. Mobile Group Perception Incentives

In mobile environments, group-aware incentive mechanisms must also address the constraints of limited computational resources and device energy consumption. Researchers have tackled these challenges by introducing personalized privacy protection and leveraging deep learning techniques to optimize computational resource usage and data utilization [8,34–38]. For instance, Sun et al. [8] designs an incentive mechanism grounded in personalized privacy protection, utilizing contract theory to ensure privacy security in mobile data sharing. On the other hand, Sun et al. [35] integrates privacy protection with personalized incentive allocation in federated learning through deep learning, thereby enhancing data processing efficiency in mobile scenarios. Additionally, Jin and Zhang [34] develops a differential privacy protection mechanism for spectrum sensing in mobile group contexts, effectively balancing privacy needs with data-sharing requirements. Despite these advancements, existing methods still face limitations in managing computational resources on mobile devices. Future research should focus on optimizing privacy protection mechanisms within resource-constrained environments to further enhance data processing efficiency in mobile settings.

2.4. Privacy Data Processing and Distribution

Ensuring privacy security during data distribution remains a significant challenge, even with advancements in privacy protection and data processing efficiency within group-aware incentive mechanisms. Researchers have proposed various data processing and publishing techniques based on differential privacy to address privacy leakage and enhance security and transparency in data publishing [8,18,34,39–41]. For example, Jin and Zhang [34] introduces a privacy protection mechanism for spectrum sensing that employs differential privacy techniques to prevent sensitive information disclosure during

data distribution. Similarly, Liu et al. [9] and Zhang et al. [3] propose solutions that enhance transparency and traceability in the data distribution process, ensuring secure data release in IoT environments and improving data trustworthiness. These studies highlight the potential of differential privacy in securing data processing and publishing. Nevertheless, there is still a necessity to boost the computational efficiency of differential privacy algorithms in large-scale data distribution scenarios, which remains a key area for future exploration.

Current privacy and incentive mechanisms in federated learning often focus on single aspects and use static models, overlooking the needs of multi-participant and resource-constrained environments. This paper introduces the Federated Learning and Dynamic Incentive Allocation Framework (FL-DIAF), which integrates differential privacy with game-theoretic incentives. FL-DIAF offers adaptable strategies to balance privacy protection, user motivation, and resource utilization, ensuring scalable and efficient solutions for large-scale distributed networks.

3. Problem Description and Formulation

3.1. System Model

In resource-constrained data-sharing environments, users share data via mobile devices, while the platform is responsible for data collection, processing, and distribution. To achieve the goals of privacy protection and incentive allocation while optimizing resource utilization, the system relies on seamless interactions between user devices and the data-sharing platform.

Figure 2 illustrates a data-sharing system model. In a resource-constrained data-sharing environment, users share data via mobile devices, while the data-sharing platform is responsible for data collection, processing, and distribution. The system consists of four core components: user devices, the data-sharing platform, the incentive mechanism, and the privacy protection mechanism. User devices upload data D_i and receive corresponding incentives R_i. The data-sharing platform uses a differential privacy mechanism to protect user data privacy and calculates the incentive based on each user's resource consumption and privacy protection level. The incentive mechanism motivates users to actively participate in data sharing, while the privacy protection mechanism ensures data privacy by controlling the privacy level parameter ϵ.

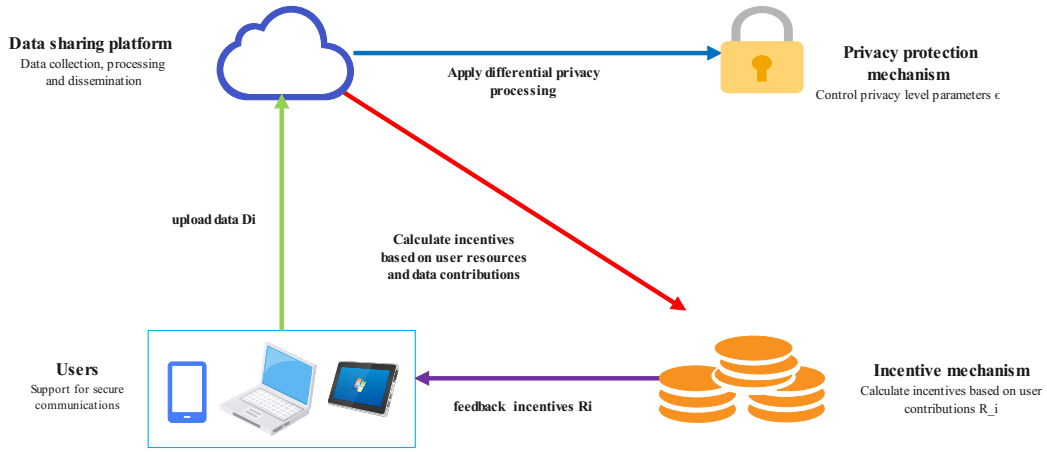


Figure 2. System model for resource-constrained data-sharing systems.

In the system, user devices and the data-sharing platform interact through secure communication protocols. The uploaded data D_i are processed through a differential privacy mechanism, and the platform calculates the incentive allocation scheme. This scheme is based on user data contribution and effective system resource utilization, which

results in the incentive R_i being fed back to the user. The goal of the system is to improve user participation through the incentive mechanism while optimizing overall system performance under resource constraints and privacy protection.

3.2. Problem Definition

The input to this problem includes the set of users $\mathcal{U} = \{1, 2, \ldots, N\}$, the amount of data uploaded by each user D_i, each user's resource consumption C_i, and the privacy protection level ϵ. The output is the incentive allocation scheme $\mathbf{R} = [R_1, R_2, \ldots, R_N]$. The goal is to maximize the user's utility U_i through the incentive mechanism while minimizing the system's energy consumption. The user utility function is defined as $U_i = \alpha R_i - \beta C_i$, where R_i represents the user's incentive, C_i represents the user's resource consumption, and α and β are weighting factors that balance the influence of incentive and resource consumption.

To address this problem, we formulate it as a multi-objective optimization problem involving incentive allocation, resource optimization, and privacy protection, among other constraints. Given the dynamic resource allocation and privacy protection mechanism, the complexity of solving the problem is high, and it is proven to be NP-hard.

3.3. Constraints

To ensure the feasibility and practicality of the solution, several constraints must be considered:

3.3.1. Resource Constraint

Each user's resource consumption C_i must not exceed the maximum allowable resource limit per user $C_{\max}$:

$$C_i \leq C_{\max}, \quad \forall i \in \mathcal{U} \tag{1}$$

This constraint ensures that individual users do not deplete their device resources excessively, which could deter participation.

3.3.2. Total Resource Constraint

The aggregate resource consumption must not exceed the system's total resource capacity C_{total}:

$$\sum_{i=1}^{N} C_i \leq C_{\text{total}} \tag{2}$$

3.3.3. Non-Negativity of Incentive

Incentives allocated must be non-negative:

$$R_i \geq 0, \quad \forall i \in \mathcal{U} \tag{3}$$

3.3.4. Incentive Budget Constraint

The total incentives distributed should not exceed the platform's budget R_{budget}:

$$\sum_{i=1}^{N} R_i \leq R_{\text{budget}} \tag{4}$$

3.3.5. Privacy Protection Constraint

The differential privacy mechanism must satisfy the ϵ_i-differential privacy for each user:

$$\Pr[f(D_i) = o] \leq e^{\epsilon_i} \cdot \Pr[f(D_i') = o], \quad \forall o \tag{5}$$

where f is the randomized algorithm implemented by the privacy mechanism. D_i and D_i' are datasets differing in a single individual's data. This constraint ensures that the presence or absence of a user's data does not significantly affect the outcome, thereby protecting individual privacy.

3.3.6. Incentive Fairness Constraint

To promote fairness, the platform imposes a constraint on the incentive disparity between users:

$$|R_i - R_j| \leq \delta, \quad \forall i, j \in \mathcal{U} \tag{6}$$

where δ is the maximum allowable difference in incentives between any two users. This prevents significant disparities that could be perceived as unfair.

3.3.7. Quality of Data Constraint

The data quality Q_i provided by each user should meet a minimum threshold $Q_{\min}$:

$$|R_i - R_j| \leq \delta, \quad \forall i, j \in \mathcal{U} \tag{7}$$

This ensures that the collected data are of sufficient quality for meaningful analysis.

3.4. Objective Function

The primary objective is to optimize the trade-offs between user utility, system energy consumption, and fairness under the given constraints. The multi-objective optimization problem can be formulated as follows:

$$\max_{\mathbf{R},\mathbf{C},\boldsymbol{\epsilon}} \sum_{i=1}^{N} U_i = \sum_{i=1}^{N} (\alpha R_i - \beta C_i - \gamma P_i) \tag{8}$$

$$\min_{\mathbf{C}} E_{\text{total}} = \sum_{i=1}^{N} E_i \tag{9}$$

$$\min_{\mathbf{R}} F = \sum_{i=1}^{N} \sum_{j=1}^{N} |R_i - R_j| \tag{10}$$

To simplify the problem, we can aggregate the objectives into a single scalar objective function using weighting coefficients:

$$\min_{\mathbf{R},\mathbf{C},\boldsymbol{\epsilon}} f = \lambda_1 \left(- \sum_{i=1}^{N} U_i \right) + \lambda_2 E_{\text{total}} + \lambda_3 F \tag{11}$$

$$\text{s.t.} \quad C1\text{–}C7$$

where λ_1, λ_2, and λ_3 are non-negative coefficients that reflect the relative importance of user utility, energy consumption, and fairness, respectively.

The inclusion of the privacy cost P_i in the utility function captures the users' sensitivity to privacy loss. A common model for P_i is

$$P_i = \phi(\epsilon_i) \tag{12}$$

where $\phi(\cdot)$ is a convex, decreasing function, indicating that a higher privacy level (smaller ϵ_i) results in a higher privacy cost due to reduced data utility.

Theorem 1. *The incentive allocation problem Equation (11) is NP-hard.*

Proof. We use a reduction from the classical 0–1 knapsack problem, which is known to be NP-hard. The 0–1 knapsack problem is defined as follows: given a set of items, each item i has a value v_i and a weight w_i. The goal is to select a subset of items such that the total weight does not exceed a given capacity $W_{\max}$, while maximizing the total value. The mathematical formulation is $\max \sum_{i=1}^{n} v_i x_i$ subject to $\sum_{i=1}^{n} w_i x_i \leq W_{\max}, \quad x_i \in \{0,1\}$.

We establish a correspondence between the variables in the incentive allocation problem and the knapsack problem. First, consider the set of users $\mathcal{U} = \{1, 2, \ldots, N\}$, where each user i corresponds to an item. For each user i, the data uploaded D_i are analogous to the weight w_i, the incentive R_i is analogous to the value v_i, and the user's resource consumption C_i corresponds to the weight in the knapsack. The total resource limit $C_{\max}$ corresponds to the knapsack's weight capacity $W_{\max}$.

The objective of the incentive allocation problem is to maximize the total user utility U_i, which is $\max \sum_{i=1}^{N} (\alpha R_i - \beta C_i)$.

To reduce the problem to the 0–1 knapsack problem, we make the following assumption: each user can either participate in data sharing or not. Let $x_i \in \{0,1\}$, where $x_i = 1$ indicates that user i participates, and $x_i = 0$ indicates non-participation. Thus, the incentive allocation problem can be rewritten as: $\max \sum_{i=1}^{N} (\alpha R_i - \beta C_i) x_i$.

The resource constraint is then expressed as $\sum_{i=1}^{N} C_i x_i \leq C_{\max}$.

This structure matches the 0–1 knapsack problem, where R_i corresponds to the value v_i, and C_i corresponds to the weight w_i. The total resource limit $C_{\max}$ is equivalent to the knapsack's capacity $W_{\max}$. Therefore, the goal is to select a subset of users to maximize the utility under resource constraints, similar to selecting items to maximize value in the knapsack problem. Since the 0–1 knapsack problem is NP-hard, and the incentive allocation problem can be reduced to the knapsack problem, the incentive allocation problem is also NP-hard and cannot be solved in polynomial time using an exact algorithm. □

4. Solving Algorithm and Methodology

This paper proposes a solution framework to address privacy protection challenges, user incentive allocation, and resource optimization in resource-constrained data-sharing environments. The method consists of two main components: a federated learning framework and a dynamic incentive allocation model, which handles privacy-preserving model training and contribution-based incentive distribution.

In the federated learning framework, users participate in global model training through local updates while employing differential privacy mechanisms to prevent raw data leakage. This process satisfies privacy and resource constraints. A Stackelberg game models the interaction between the platform and users, optimizing incentive allocation and privacy levels to balance user utility and system performance.

The dynamic incentive allocation model quantifies users' contributions to the system utility using the Shapley value. To handle large-scale user scenarios, a Monte Carlo approximation method is used for Shapley value computation. The model dynamically adjusts incentive distribution based on resource consumption, privacy costs, and user contributions, ensuring fairness and compliance with budget constraints.

This framework integrates advanced designs to collaboratively achieve privacy protection, incentive allocation, and resource optimization. And it works to provide a practical solution for data sharing in resource-constrained environments.

4.1. Federated Learning Framework with Privacy and Incentive Mechanisms

Optimizing user participation incentives and system performance while ensuring data privacy is a significant challenge in resource-constrained data-sharing systems. We

propose a strategy that combines federated learning, differential privacy, and a Stackelberg game-based incentive mechanism. This method balances user utility, resource consumption, and privacy protection under the constraints outlined in the problem model. Our innovation is seamlessly integrating these components to address the multi-objective optimization problem in resource-constrained environments.

Traditional centralized data-sharing systems require users to upload their raw data D_i to a central server for processing. This raises significant privacy concerns and leads to excessive resource consumption on user devices. Federated learning enables users to train a global model collaboratively without sharing their raw data, which addresses these issues. We enhance privacy protection by incorporating differential privacy into the federated learning process.

We consider a set of users $\mathcal{U} = \{1, 2, \ldots, N\}$, where each user i has

Local dataset D_i of size $n_i = |D_i|$, containing data samples (x_{ij}, y_{ij}); resource consumption C_i, including computation cost C_i^{cmp} and communication cost C_i^{com}; and privacy protection level ϵ_i.

Other key variables are essential for understanding the model. The local model parameters of the user i are denoted as $\boldsymbol{\theta}_I \in \mathbb{R}^d$, where d represents the dimensionality of the parameter space. After incorporating differential privacy mechanisms, the parameters are transformed into noisy model parameters, denoted by $\tilde{\boldsymbol{\theta}}_I$. The platform aggregates these local parameters from all users to obtain the global model parameters, represented as $\boldsymbol{\theta}$. Each user i introduces Gaussian noise to their parameters with a standard deviation σ_i to ensure privacy. The sensitivity of the loss function for user i, which captures the impact of adding or removing a single sample, is indicated by Δ_i. For a given data sample (x_{ij}, y_{ij}), the loss function is expressed as $f(\boldsymbol{\theta}_I; x_{ij}, y_{ij})$, reflecting how well the model fits the data. To incentivize participation, the platform allocates a reward to the user i, denoted by R_i. However, the chosen privacy level ϵ_i incurs a privacy cost for the user, represented by $P_i(\epsilon_i)$. Balancing these various aspects, the user's utility function incorporates weighting coefficients α, β, and γ, which govern the trade-offs between incentives, resource consumption, and privacy costs, ensuring the model aligns with system goals and user interests.

Our federated learning process consists of the following steps, aiming to balance model performance, resource consumption, and data privacy.

4.1.1. Differential Privacy and Federated Learning Mechanism

This mechanism is designed to satisfy the privacy protection constraint Equation (5) in the problem model by ensuring that each user's contribution to the global model is differentially private. It also addresses the resource constraints Equations (1) and (2) by optimizing resource consumption during local training and communication.

Each user $i \in \mathcal{U}$ trains a local model by minimizing a local loss function $F_i(\boldsymbol{\theta}_i)$:

$$F_i(\boldsymbol{\theta}_i) = \frac{1}{n_i} \sum_{j=1}^{n_i} f(\boldsymbol{\theta}_i; x_{ij}, y_{ij}) \tag{13}$$

where $f(\boldsymbol{\theta}_i; x_{ij}, y_{ij})$ could be, for example, the mean squared error (MSE) for regression tasks:

$$f(\boldsymbol{\theta}_i; x_{ij}, y_{ij}) = \left(y_{ij} - \boldsymbol{\theta}_i^\top x_{ij} \right)^2 \tag{14}$$

or the cross-entropy loss for classification tasks:

$$f(\boldsymbol{\theta}_i; x_{ij}, y_{ij}) = -y_{ij} \log\left(\sigma(\boldsymbol{\theta}_i^\top x_{ij}) \right) - (1 - y_{ij}) \log\left(1 - \sigma(\boldsymbol{\theta}_i^\top x_{ij}) \right) \tag{15}$$

where $\sigma(\cdot)$ denotes the sigmoid function. Users perform multiple local iterations of stochastic gradient descent (SGD) to minimize $F_i(\boldsymbol{\theta}_i)$:

$$\boldsymbol{\theta}_i^{(k+1)} = \boldsymbol{\theta}_i^{(k)} - \eta \nabla F_i(\boldsymbol{\theta}_i^{(k)}) \tag{16}$$

where η is the learning rate, and k indexes the local iterations. To protect user privacy, each user applies a differential privacy mechanism to their model updates. Specifically, each user adds Gaussian noise to their model parameters:

$$\tilde{\boldsymbol{\theta}}_i = \boldsymbol{\theta}_i + \mathcal{N}(0, \sigma_i^2 \mathbf{I}) \tag{17}$$

where $\mathcal{N}(0, \sigma_i^2 \mathbf{I})$ denotes a multivariate Gaussian distribution with mean zero and covariance matrix $\sigma_i^2 \mathbf{I}$, and $\mathbf{I}$ is the identity matrix of size $d \times d$.

The l_2-sensitivity Δ_i is defined as

$$\Delta_i = \max_{D_i, D_i'} \left\| F_i(\boldsymbol{\theta}_i; D_i) - F_i(\boldsymbol{\theta}_i; D_i') \right\|_2 \tag{18}$$

where D_i and D_i' are neighboring datasets differing in a single data point. For functions that are averages over data samples, the sensitivity can be bounded as:

$$\Delta_i = \frac{1}{n_i} \max_{\boldsymbol{\theta}_i, x_{ij}, y_{ij}} \left\| f(\boldsymbol{\theta}_i; x_{ij}, y_{ij}) - f(\boldsymbol{\theta}_i; x_{ij}', y_{ij}') \right\|_2 \tag{19}$$

Assuming the loss function is bounded, and the data are normalized, Δ_i can be considered a known constant. To achieve ϵ_i-differential privacy with a failure probability δ, the noise scale σ_i is calculated as

$$\sigma_i = \frac{\Delta_i}{\epsilon_i} \sqrt{2 \ln \frac{1.25}{\delta}} \tag{20}$$

Lemma 1. *Adding Gaussian noise as per Equation* (17) *ensures that the mechanism satisfies* (ϵ_i, δ)*-differential privacy [42].*

Proof. To establish that adding Gaussian noise to the function f satisfies (ϵ, δ)-differential privacy, we proceed by applying the Gaussian Mechanism for differential privacy. According to the Gaussian Mechanism, a function f achieves (ϵ, δ)-differential privacy if Gaussian noise with standard deviation σ is added, where the noise distribution is $\mathcal{N}(0, \sigma^2)$. The relationship between the noise scale σ and the desired privacy parameters ϵ and δ is given by

$$\sigma \geq \frac{\Delta f}{\epsilon} \sqrt{2 \ln \frac{1.25}{\delta}}$$

Δf denotes the l_2-sensitivity of the function f, defined as the maximum possible change in the output of f over any two neighboring datasets D and D'. For our specific case, the sensitivity is $\Delta f = \Delta_i$, which corresponds to the sensitivity of the loss function used in our federated learning framework.

Setting the noise scale σ_i according to the above equation, we ensure that the amount of noise added aligns with the desired privacy parameters. Thus, the noisy version of the function f, denoted as $f(D_i) + \mathcal{N}(0, \sigma_i^2)$, ensures that the privacy requirements are met. To verify this formally, consider two neighboring datasets D_i and D_i' that differ by at most one element.

For any measurable subset $S \subseteq \mathbb{R}^d$, the Gaussian Mechanism ensures that the probability of the noisy function's output falling within S is bounded by

$$\Pr[f(D_i) + \mathcal{N}(0, \sigma_i^2) \in S] \leq e^{\epsilon_i} \Pr[f(D_i') + \mathcal{N}(0, \sigma_i^2) \in S] + \delta$$

This inequality shows that the presence or absence of a single element in the dataset has a limited impact on the probability distribution of the noisy output, controlled by the parameters ϵ_i and δ. The term e^{ϵ_i} ensures that the probability ratio between the two neighboring datasets remains bounded, while δ accounts for a small probability of failure where the guarantee might not hold.

The Gaussian noise with variance σ_i^2, scaled as described, ensures that this bound is respected. Consequently, as required, the noisy function satisfies the definition of (ϵ_i, δ)-differential privacy. □

The platform aggregates the noisy model parameters received from all users to update the global model:

$$\boldsymbol{\theta} = \frac{1}{N} \sum_{i=1}^{N} \tilde{\boldsymbol{\theta}}_i \tag{21}$$

This aggregation reduces the variance introduced by the added noise due to the averaging effect. The variance of the aggregated model is reduced by a factor of N:

$$\mathrm{Var}[\boldsymbol{\theta}] = \frac{1}{N^2} \sum_{i=1}^{N} \sigma_i^2 \mathbf{I} = \frac{\sigma^2}{N} \mathbf{I} \tag{22}$$

assuming all $\sigma_i = \sigma$, the updated global model $\boldsymbol{\theta}$ is then distributed back to all users, who update their local models accordingly for the next iteration.

This process ensures that the privacy protection constraint is met while the resource constraints are managed by controlling the number of local iterations and the communication frequency, as a result, optimizing resource consumption.

Under standard assumptions for convex loss functions and bounded gradients, federated learning with differential privacy converges to a neighborhood of the optimal solution. The added noise introduces a bias in estimating the global model, but the impact can be minimized with sufficient iterations and appropriate noise scaling.

Theorem 2 (Convergence of Federated Learning with Differential Privacy)**.** *Let* $F(\boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1}^{N} F_i(\boldsymbol{\theta})$ *be the global loss function. Under assumptions of Lipschitz continuity and strong convexity of* F_i*, the federated learning algorithm converges to a global optimum within a bounded error due to noise.*

Proof. Assume each local loss function $F_i(\boldsymbol{\theta})$ is L-Lipschitz continuous, meaning that the gradient of the function is bounded. For any two model parameters $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$, the following inequality holds:

$$\|\nabla F_i(\boldsymbol{\theta}) - \nabla F_i(\boldsymbol{\theta}')\|_2 \leq L \|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_2.$$

Additionally, assume that $F_i(\boldsymbol{\theta})$ is μ-strongly convex, indicating that the function grows at least quadratically within the parameter space:

$$F_i(\boldsymbol{\theta}) \geq F_i(\boldsymbol{\theta}') + \nabla F_i(\boldsymbol{\theta}')^\top (\boldsymbol{\theta} - \boldsymbol{\theta}') + \frac{\mu}{2} \|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_2^2.$$

The goal is to minimize the global loss function $F(\boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1}^{N} F_i(\boldsymbol{\theta})$ through federated learning. In each iteration t, the model parameters are updated using stochastic gradient descent (SGD) with the following update rule:

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \eta \nabla F(\boldsymbol{\theta}^{(t)}) + \xi^{(t)},$$

where $\xi^{(t)}$ represents the noise term drawn from a Gaussian distribution with variance σ^2. We need to analyze the convergence by estimating the expected squared distance between the model parameters $\boldsymbol{\theta}^{(t)}$ and the optimal solution $\boldsymbol{\theta}^*$.

We introduce the following recurrence relation to describe the distance between $\boldsymbol{\theta}^{(t)}$ and $\boldsymbol{\theta}^*$ at each iteration:

$$\mathbb{E}\left[\|\boldsymbol{\theta}^{(T)} - \boldsymbol{\theta}^*\|_2^2\right].$$

Using the assumptions of Lipschitz continuity and strong convexity, we obtain the following bound for the gradient difference:

$$\|\nabla F(\boldsymbol{\theta}^{(t)}) - \nabla F(\boldsymbol{\theta}^*)\|_2 \leq L\|\boldsymbol{\theta}^{(t)} - \boldsymbol{\theta}^*\|_2.$$

Substituting this bound into the model update equation and applying the strong convexity condition, we derive the following recursive inequality:

$$\|\boldsymbol{\theta}^{(t+1)} - \boldsymbol{\theta}^*\|_2^2 \leq \left(1 - \frac{2\eta\mu L}{\mu + L}\right)\|\boldsymbol{\theta}^{(t)} - \boldsymbol{\theta}^*\|_2^2 + \eta^2\sigma^2.$$

By expanding the recurrence relation over T iterations and solving the resulting inequality, we obtain the following upper bound:

$$\mathbb{E}\left[\|\boldsymbol{\theta}^{(T)} - \boldsymbol{\theta}^*\|_2^2\right] \leq \left(1 - \frac{2\eta\mu L}{\mu + L}\right)^T \|\boldsymbol{\theta}^{(0)} - \boldsymbol{\theta}^*\|_2^2 + \frac{\eta\sigma^2}{\mu}.$$

As $T \to \infty$, the first term in the bound converges to zero, meaning that the parameters converge to a neighborhood around the optimal solution $\boldsymbol{\theta}^*$. The radius of this neighborhood is proportional to the noise variance σ^2, as shown by the following expression:

$$\lim_{T\to\infty} \mathbb{E}\left[\|\boldsymbol{\theta}^{(T)} - \boldsymbol{\theta}^*\|_2^2\right] \leq \frac{\eta\sigma^2}{\mu}.$$

Therefore, with appropriate noise scaling and a sufficiently small learning rate, the federated learning algorithm converges to a solution that is close to the optimal parameters. This demonstrates that the algorithm achieves stable convergence even in the presence of noise. □

4.1.2. Incentive Mechanism Based on Stackelberg Game Theory

This incentive mechanism addresses the incentive allocation and fairness constraints Equations (4) and (7) in the problem model by modeling the interaction between the platform and users as a Stackelberg game. The platform (leader) aims to maximize overall system utility, while users (followers) adjust their strategies to maximize their individual utilities under the constraints specified.

We model the interaction between the platform and the users as a Stackelberg game, where the platform aims to optimize the overall system performance by strategically allocating incentives R_i and determining acceptable privacy levels ϵ_i for each user $i \in \mathcal{U}$. The users, in turn, decide on their resource consumption C_i and privacy levels ϵ_i to maximize their individual utilities, given the platform's strategies. The utility function for each user i is defined as

$$U_i(R_i, C_i, \epsilon_i) = \alpha R_i - \beta C_i - \gamma P_i(\epsilon_i), \tag{23}$$

where α, β, γ are weighting coefficients reflecting the importance of incentive, resource consumption, and privacy cost, respectively. The privacy cost $P_i(\epsilon_i)$ is modeled as

$$P_i(\epsilon_i) = \frac{\kappa_i}{\epsilon_i}, \tag{24}$$

where κ_i being the privacy sensitivity coefficient of user i. The resource consumption C_i comprises computation and communication costs:

$$C_i = C_i^{\text{cmp}} + C_i^{\text{com}}, \tag{25}$$

where $C_i^{\text{cmp}} = \beta_i n_i \eta_i$ is the computation cost, n_i is the size of user i's dataset, η_i is the number of local training epochs, and β_i is the unit computation cost. The communication cost is $C_i^{\text{com}} = \gamma_i d$, with γ_i being the unit communication cost and d the dimension of the model parameters.

The users aim to maximize their utilities $U_i(R_i, C_i, \epsilon_i)$ by choosing appropriate C_i and ϵ_i, subject to the following constraints:

$$C_i \leq C_{\max}, \quad \forall i \in \mathcal{U}, \tag{26}$$

ensuring that the resource consumption does not exceed the maximum allowable limit.

$$\epsilon_i \geq \epsilon_{\min}, \quad \forall i \in \mathcal{U}, \tag{27}$$

guaranteeing a minimum level of privacy protection.

$$C_i \geq 0, \quad \epsilon_i \geq 0, \quad \forall i \in \mathcal{U}. \tag{28}$$

The platform's decision variables are the incentives R_i and the acceptable privacy levels ϵ_i. The platform is subject to the following constraints:

$$\sum_{i=1}^{N} R_i \leq R_{\text{budget}}, \tag{29}$$

where R_{budget} is the total incentive budget available to the platform.

$$|R_i - R_j| \leq \delta, \quad \forall i, j \in \mathcal{U}, \tag{30}$$

ensuring that the difference in incentives between any two users does not exceed a threshold δ, promoting fairness.

$$R_i \geq 0, \quad \forall i \in \mathcal{U}. \tag{31}$$

The platform's objective is to maximize the overall system utility, which can be represented as the sum of all users' utilities, while adhering to budget and fairness constraints:

$$\max_{\{R_i\},\{\epsilon_i\}} \sum_{i=1}^{N} U_i(R_i, C_i, \epsilon_i). \tag{32}$$

To solve this hierarchical optimization problem, we use backward induction. First of all, we derive the users' best responses to the platform's strategies by solving their individual optimization problems. Then, we incorporate these responses into the platform's optimization problem to determine the optimal incentives and acceptable privacy levels.

Each user i aims to maximize their utility $U_i(R_i, C_i, \epsilon_i)$ with respect to C_i and ϵ_i, given R_i and the constraints in Equations (26) and (27). Formally, the user's optimization problem is

$$\begin{aligned} \max_{C_i, \epsilon_i} \quad & U_i(R_i, C_i, \epsilon_i) = \alpha R_i - \beta C_i - \gamma \frac{\kappa_i}{\epsilon_i} \\ \text{s.t.} \quad & C_i \leq C_{\max}, \\ & \epsilon_i \geq \epsilon_{\min}, \\ & C_i \geq 0, \\ & \epsilon_i \geq 0. \end{aligned} \tag{33}$$

Since R_i is given from the platform's strategy, the term αR_i is constant with respect to the user's decision variables C_i and ϵ_i. To solve the user's optimization problem, we formulate the Lagrangian function:

$$\mathcal{L}_i = \alpha R_i - \beta C_i - \gamma \frac{\kappa_i}{\epsilon_i} + \lambda_i^C (C_{\max} - C_i) + \lambda_i^\epsilon (\epsilon_i - \epsilon_{\min}), \tag{34}$$

where $\lambda_i^C \geq 0$ and $\lambda_i^\epsilon \geq 0$ are the Lagrange multipliers associated with the resource and privacy constraints, respectively. Taking the partial derivatives of the Lagrangian with respect to C_i and ϵ_i, we have

$$\frac{\partial \mathcal{L}_i}{\partial C_i} = -\beta - \lambda_i^C = 0, \tag{35}$$

$$\frac{\partial \mathcal{L}_i}{\partial \epsilon_i} = \gamma \frac{\kappa_i}{\epsilon_i^2} + \lambda_i^\epsilon = 0. \tag{36}$$

From Equation (35), we find

$$\lambda_i^C = -\beta. \tag{37}$$

Since $\lambda_i^C \geq 0$ and $\beta > 0$, this leads to a contradiction unless $\beta = 0$, which is not practical. Therefore, the resource constraint is active, and the user consumes the maximum allowable resources:

$$C_i^* = C_{\max}. \tag{38}$$

Similarly, from Equation (36), we obtain

$$\lambda_i^\epsilon = -\gamma \frac{\kappa_i}{\epsilon_i^2}. \tag{39}$$

Since $\lambda_i^\epsilon \geq 0$ and $\gamma, \kappa_i > 0$, it follows that

$$\epsilon_i^* = \epsilon_{\min}. \tag{40}$$

Thus, the user opts for the minimum acceptable privacy level to maximize utility. Substituting the users' optimal responses into the platform's objective function Equation (32), we have

$$\max_{\{R_i\}} \sum_{i=1}^{N} \left(\alpha R_i - \beta C_{\max} - \gamma \frac{\kappa_i}{\epsilon_{\min}} \right). \tag{41}$$

Since $C_{\max}$ and $\epsilon_{\min}$ are constants, the platform's problem simplifies to maximizing $\sum_{i=1}^{N} \alpha R_i$ under the constraints in Equations (29)–(31). To solve this, we observe that maximizing $\sum_{i=1}^{N} R_i$ will maximize the platform's objective, given that $\alpha > 0$. Therefore, the platform should allocate the entire budget R_{budget} to the users.

Considering the fairness constraint Equation (30), an equitable distribution of incentives is

$$R_i^* = \frac{R_{\text{budget}}}{N}, \quad \forall i \in \mathcal{U}, \tag{42}$$

which satisfies $|R_i - R_j| = 0 \leq \delta$. To ensure that the obtained solutions satisfy the Karush-Kuhn-Tucker (KKT) conditions, we verify the complementary slackness conditions.

For the user:

Resource constraint:

$$\lambda_i^C(C_{\max} - C_i^*) = \lambda_i^C(C_{\max} - C_{\max}) = 0. \tag{43}$$

Privacy constraint:

$$\lambda_i^\epsilon(\epsilon_i^* - \epsilon_{\min}) = \lambda_i^\epsilon(\epsilon_{\min} - \epsilon_{\min}) = 0. \tag{44}$$

For the platform:

Incentive budget constraint:

$$\lambda_0\left(\sum_{i=1}^{N} R_i^* - R_{\text{budget}}\right) = \lambda_0(R_{\text{budget}} - R_{\text{budget}}) = 0. \tag{45}$$

The incentive fairness constraint is satisfied by the equal distribution. Therefore, the KKT conditions are satisfied, confirming that the solutions $C_i^* = C_{\max}$, $\epsilon_i^* = \epsilon_{\min}$, and $R_i^* = R_{\text{budget}}/N$ are optimal.

In the federated learning framework, the users perform local model training using their maximum allowable resources and adopt the minimum acceptable privacy levels. The platform aggregates the users' contributions and updates the global model accordingly. At each iteration t, users update their local models:

$$\boldsymbol{\theta}_i^{(t)} = \boldsymbol{\theta}^{(t-1)} - \eta \nabla F_i(\boldsymbol{\theta}^{(t-1)}), \tag{46}$$

where η is the learning rate, and $\nabla F_i(\boldsymbol{\theta}^{(t-1)})$ is the gradient of the local loss function. Then, users add differential privacy noise:

$$\tilde{\boldsymbol{\theta}}_i^{(t)} = \boldsymbol{\theta}_i^{(t)} + \mathcal{N}\left(0, \sigma_i^2 \mathbf{I}\right), \tag{47}$$

with noise scale computed using:

$$\sigma_i = \frac{\Delta_i}{\epsilon_{\min}} \sqrt{2 \ln \frac{1.25}{\delta}}, \tag{48}$$

where Δ_i is the sensitivity of the loss function, and δ is the failure probability.

The platform aggregates the noisy models:

$$\boldsymbol{\theta}^{(t)} = \frac{1}{N} \sum_{i=1}^{N} \tilde{\boldsymbol{\theta}}_i^{(t)}. \tag{49}$$

Figure 3 presents the architecture of the proposed federated learning framework with a Stackelberg game-based incentive mechanism. The platform, positioned at the top, acts as the leader, allocating incentives and setting privacy levels. Each user device, represented by rounded rectangles, performs local model training and applies differential privacy before sending noisy parameters to the platform. The platform aggregates these parameters to form the global model. This figure illustrates the interaction flow between the platform and user devices, balancing incentives, resource consumption, and privacy protection to optimize learning performance.

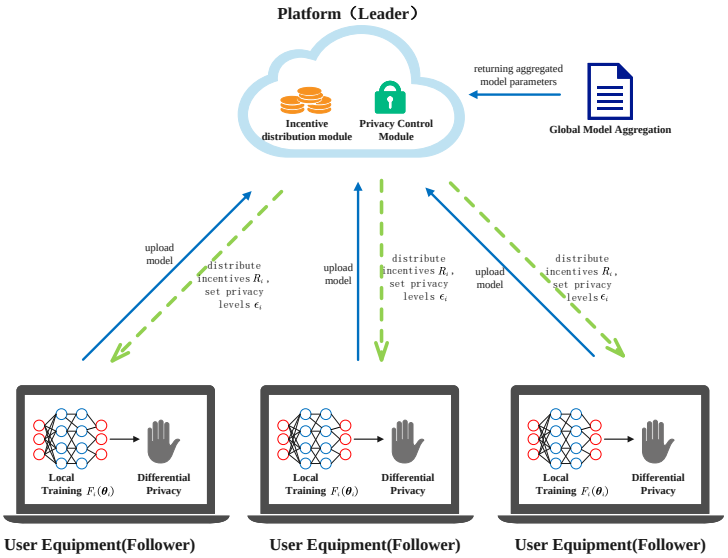


Figure 3. Architecture of federated learning framework with differential privacy and incentive mechanism.

Algorithm 1, Federated Learning with Stackelberg Game-Based Incentive Mechanism (FL-SGIM), integrates a federated learning process with a Stackelberg game-based incentive mechanism. It aims to efficiently train a global model while balancing user participation, privacy protection, and resource usage. By encouraging active user participation within their resource and privacy constraints, the algorithm ensures fairness and scalability.

Algorithm 1 FL-SGIM For Model Optimization

1: **Input:** Initial global model parameters $\boldsymbol{\theta}^{(0)}$, incentive budget R_{budget}, minimum privacy level $\epsilon_{\min}$, maximum resource capacity $C_{\max}$, learning rate η, convergence threshold ε.
2: **Output:** Optimized global model $\boldsymbol{\theta}^*$.
3: Initialize incentives $R_i^{(0)} = \frac{R_{\text{budget}}}{N}$ for all users $i \in \mathcal{U}$.
4: Set privacy levels $\epsilon_i^{(0)} = \epsilon_{\min}$ for all users.
5: Initialize iteration counter $t \leftarrow 0$.
6: **repeat**
7: $\quad t \leftarrow t + 1$
8: $\quad$ **for** each user $i \in \mathcal{U}$ **in parallel do**
9: $\quad\quad$ Perform local training using Equation (46)
10: $\quad\quad$ Compute noise scale σ_i using Equation (48):
11: $\quad\quad$ Generate noisy model parameters using Equation (47):
12: $\quad\quad$ Ensure resource consumption $C_i \leq C_{\max}$.
13: $\quad\quad$ Send $\tilde{\boldsymbol{\theta}}_i^{(t)}$ to the platform.
14: $\quad$ **end for**
15: $\quad$ Aggregate the global model using Equation (49):
16: **until** $\left\| \boldsymbol{\theta}^{(t)} - \boldsymbol{\theta}^{(t-1)} \right\|_2 \leq \varepsilon$
17: **return** Optimized global model $\boldsymbol{\theta}^* = \boldsymbol{\theta}^{(t)}$.

The algorithm starts by evenly distributing the total incentive budget, giving each user $R_i^{(0)} = \frac{R_{\text{budget}}}{N}$ (Line 3). It assigns the minimum privacy level $\epsilon_{\min}$ to all users to ensure baseline privacy protection (Line 4) and initializes the iteration counter $t = 0$ (Line 5). During each iteration, the counter t is incremented (Line 7). All users $i \in \mathcal{U}$ perform parallel local training using stochastic gradient descent (SGD) as per Equation (46) (Line 9). To maintain differential privacy, each user computes the noise scale σ_i (Line 10) and adds Gaussian noise to generate private model parameters $\tilde{\boldsymbol{\theta}}_i$ (Line 11). Users ensure their resource consumption C_i stays within the allowed limit $C_{\max}$ (Line 12) and send the

noisy parameters $\tilde{\boldsymbol{\theta}}_i^{(t)}$ to the platform (Line 13). The platform aggregates the models using Equation (49) (Line 15) and checks if the difference between the current and previous global models, measured by $\left\|\boldsymbol{\theta}^{(t)} - \boldsymbol{\theta}^{(t-1)}\right\|_2$, falls below the threshold ε (Line 16). If not, further iterations are performed. Once convergence is achieved, the optimized global model $\boldsymbol{\theta}^*$ is returned, completing the training process (Line 17).

The computational complexity for each user is $O(n_i d)$, where n_i is the data size and d is the model dimension. The communication complexity per user is $O(d)$. The overall complexity scales linearly with the number of users N, making it efficient for large-scale systems.

Under the standard assumptions of convexity and Lipschitz continuity of the loss functions, the algorithm converges to a global model within a bounded error. The noise introduced by differential privacy diminishes with the number of users due to the averaging effect during aggregation. The choice of parameters like $\epsilon_{\min}$ involves a trade-off between privacy protection and model accuracy. A smaller $\epsilon_{\min}$ offers stronger privacy but may introduce more noise, degrading model performance. Careful tuning of the learning rate η and convergence threshold ε ensures the stability and efficiency of the learning process.

This algorithm integrates a Stackelberg game-based incentive mechanism into federated learning, addressing key user participation, privacy, and resource allocation challenges. It ensures fair and transparent incentive distribution while maintaining user privacy through differential privacy techniques. The algorithm also accommodates user heterogeneity by managing resource consumption limits and supporting personalized incentives. The platform achieves a scalable, efficient, and privacy-preserving federated learning system through this approach.

In this method, the incentive allocation mechanism between the platform and users is implemented by introducing Stackelberg game theory. Specifically, as the leader of the game, the platform first formulates incentive strategies and privacy levels based on the overall goals and constraints of the system. When formulating these strategies, the platform takes into account the optimal response of users, that is, how users optimize their resource consumption C_i and privacy choices ϵ_i) to maximize their own utility under given incentives and privacy requirements. As followers of the game, users choose the optimal resource consumption and privacy level based on their own utility function $U_i(R_i, C_i, \epsilon_i)$ after the platform strategy is determined.

Through this leadership–follower game structure, the platform is able to anticipate and guide user behavior, thereby optimizing overall system utility. In each round of the game, the platform adjusts the incentive amount and privacy parameters based on the current incentive allocation and privacy requirements, in order to encourage users to achieve a balance between resource utilization and privacy protection. Due to the incentive allocation strategy of the platform being based on the optimal response of users, a balanced state is ultimately achieved, making incentive allocation both fair and transparent, while meeting budget and fairness constraints. By combining the Shapley value dynamic incentive allocation model, the platform can not only quantify the contribution of each user but also ensure fairness and efficient use of resources in the incentive allocation process through game theory methods, thereby enhancing user trust and participation enthusiasm.

4.2. Dynamic Incentive Allocation Model

In resource-constrained data-sharing networks, achieving a fair, transparent, and efficient incentive allocation mechanism is crucial for promoting user participation and enhancing overall system performance. In the previous section, we described the system architecture, constraint conditions, and objective functions. Building on this foundation, to further address the fairness objective (minimizing FFF) and the incentive fairness constraint Equation (7), this subsection delves into the design and derivation of a dynamic

incentive allocation model based on the Shapley value. This model provides a fair and transparent method for distributing incentives based on each user's contribution to the system's utility.

4.2.1. Fair Incentive Distribution Based on Shapley Value

Figure 4 illustrates the dynamic incentive allocation process based on the Shapley value. In this flow, the set of users and their data characteristics are first taken as inputs to calculate the contribution value of each user in the coalition, and the marginal contribution of each user is evaluated by the Shapley value formula. To ensure computational efficiency in large-scale networks, a Monte Carlo approximation is used to estimate the Shapley value by sample alignment, and finally, the estimation results are used for incentive allocation. This model not only ensures the fairness of incentive allocation but also protects user privacy while maintaining high efficiency.

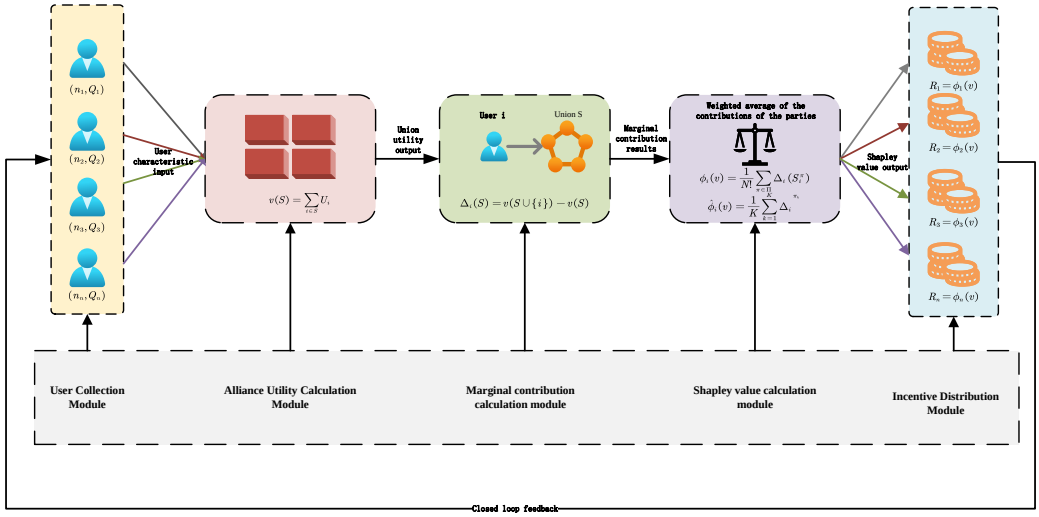


Figure 4. Dynamic incentive allocation process based on Shapley value.

The Shapley value, originating from cooperative game theory, serves as a core method for fairly distributing the total gains generated from cooperation among multiple participants. In our data-sharing network, each user's contribution depends not only on their own data volume and quality but also on the participation of other users. Utilizing the Shapley value allows for a fair assessment of each user's actual contribution, ensuring fairness and transparency in incentive distribution.

As described in the previous section, we consider a user set $\mathcal{U} = \{1, 2, \ldots, N\}$, where each user $i \in \mathcal{U}$ possesses a local dataset D_i with data volume $n_i = |D_i|$ and data quality Q_i. Each user incurs resource consumption C_i and adopts differential privacy mechanisms with privacy parameter ϵ_i, satisfying the constraints outlined earlier.

First, we define the coalition value function $v(S)$ to measure the total utility of a coalition $S \subseteq \mathcal{U}$:

$$v(S) = \sum_{i \in S} U_i. \tag{50}$$

To quantify the individual user's contribution to the coalition, we introduce the concept of marginal contribution. The marginal contribution $\Delta_i(S)$ of user i to coalition S is defined as

$$\Delta_i(S) = v(S \cup \{i\}) - v(S). \tag{51}$$

The Shapley value measures a user's fair contribution by averaging their marginal contributions over all possible user permutations:

$$\phi_i(v) = \frac{1}{N!} \sum_{\pi \in \Pi} \Delta_i(S_i^\pi), \tag{52}$$

where Π is the set of all $N!$ possible permutations of users, and S_i^π denotes the set of users preceding user i in permutation π.

The Shapley value has several important properties: efficiency, symmetry, linearity, fairness, and monotonicity. These properties ensure the fairness and effectiveness of incentive allocation and encourage users to participate actively and contribute.

To align the incentive allocation with the user's actual contribution, we define the incentive R_i of user i as their Shapley value:

$$R_i = \phi_i(v). \tag{53}$$

Therefore, the incentive R_i reflects the contribution of user i to the overall utility of the system, ensuring fairness and efficiency in incentive distribution.

The Shapley value is a solution concept in cooperative game theory that fairly distributes the total gains generated by the coalition of all users based on their individual contributions. It considers all possible subsets of users and calculates each user's marginal contribution to every possible coalition, ensuring a fair and unbiased incentive distribution.

Theorem 3. *The dynamic incentive allocation model based on the Shapley value ensures equitable and transparent distribution of incentives by satisfying the following Shapley axioms:*

1. ***Efficiency**: The total incentives are fully distributed among all users, i.e., $\sum_{i=1}^{N} R_i = R_{budget}$.*
2. ***Symmetry**: If two users contribute equally to all possible coalitions, they receive identical incentives, i.e., $R_i = R_j$ whenever $v(S \cup \{i\}) = v(S \cup \{j\})$ for all $S \subseteq \mathcal{U} \setminus \{i, j\}$.*
3. ***Dummy Player**: If a user does not contribute any additional value to any coalition, their incentive is zero, i.e., if $v(S \cup \{i\}) = v(S)$ for all $S \subseteq \mathcal{U} \setminus \{i\}$, then $R_i = 0$.*
4. ***Additivity**: For any two games with characteristic functions v and w, the Shapley value for the game $v + w$ is the sum of the Shapley values for games v and w individually.*

Consequently, the Shapley value-based incentive allocation model guarantees that each user's incentive is fairly determined by their marginal contribution to the overall system utility, ensuring both fairness and transparency in the distribution process.

Proof. Efficiency: By definition, the Shapley value distributes the total value of the grand coalition $v(\mathcal{U})$ among all players such that $\sum_{i=1}^{N} \phi_i(v) = v(\mathcal{U})$. In our model, $v(\mathcal{U}) = R_{\text{budget}}$, ensuring that all incentives are fully allocated.

Symmetry: If two users i and j contribute equally to every possible coalition, then for every $S \subseteq \mathcal{U} \setminus \{i, j\}$, $v(S \cup \{i\}) = v(S \cup \{j\})$. Thus, their Shapley values satisfy $\phi_i(v) = \phi_j(v)$, ensuring equal incentives.

Dummy Player: If a user i does not contribute additional value to any coalition, i.e., $v(S \cup \{i\}) = v(S)$ for all $S \subseteq \mathcal{U} \setminus \{i\}$, then $\phi_i(v) = 0$. This ensures that users who do not enhance system utility receive no incentives.

Additivity: For any two characteristic functions v and w, the Shapley value for the combined game $v + w$ satisfies $\phi_i(v + w) = \phi_i(v) + \phi_i(w)$. This property ensures that incentives from independent sources can be aggregated linearly.

Since the Shapley value satisfies all four axioms, it guarantees that the incentive distribution is both fair and transparent, as each user's incentive precisely reflects their contribution to the system utility. □

The Shapley value-based dynamic incentive allocation model thus provides a principled approach to distributing incentives fairly and transparently, ensuring that each user's reward is directly proportional to their actual contribution to the system's overall utility.

4.2.2. Approximation Method for Scalable Incentive Allocation

Given the computational complexity of calculating the Shapley value directly, we employ an approximation method using the Monte Carlo sampling technique. We use this approach to ensure scalability in large-scale systems while still satisfying the constraints and objectives in our problem model, particularly those related to computational efficiency and privacy protection.

Specifically, we set an appropriate sample size K, satisfying $K \ll N!$, to balance computational complexity and estimation accuracy. Then, we randomly generate K user permutations $\{\pi_k\}_{k=1}^{K}$, where each permutation π_k is a random ordering of $\mathcal{U}$.

For each user i and each permutation π_k, we compute their marginal contribution in that permutation:

$$\Delta_i^{\pi_k} = v\big(S_i^{\pi_k} \cup \{i\}\big) - v\big(S_i^{\pi_k}\big). \tag{54}$$

By averaging the marginal contributions over all permutations, we obtain the approximate Shapley value:

$$\hat{\phi}_i(v) = \frac{1}{K} \sum_{k=1}^{K} \Delta_i^{\pi_k}. \tag{55}$$

According to the law of large numbers, when K is sufficiently large, the approximate Shapley value $\hat{\phi}_i(v)$ will converge to the true Shapley value $\phi_i(v)$. Increasing K can reduce the estimation variance and improve approximation accuracy.

To establish a complete incentive allocation model, we derive the specific expressions for the resource consumption C_i and privacy cost P_i used in the utility function U_i.

First, the resource consumption C_i includes computational cost C_i^{c} and communication cost C_i^{t}, expressed as

$$C_i^{\mathrm{c}} = \rho n_i, \tag{56}$$

$$C_i^{\mathrm{t}} = \tau n_i, \tag{57}$$

where ρ is the computational cost per unit of data, τ is the transmission cost per unit of data, and n_i is the data volume of user i. The total resource consumption is

$$C_i = C_i^{\mathrm{c}} + C_i^{\mathrm{t}} = (\rho + \tau) n_i. \tag{58}$$

Next, the privacy cost P_i is modeled using the privacy cost function $\phi(\epsilon_i)$, which is a convex decreasing function of the privacy parameter ϵ_i:

$$P_i = \phi(\epsilon_i) = \frac{\kappa}{\epsilon_i}, \tag{59}$$

where κ is the privacy sensitivity coefficient, representing the user's sensitivity to privacy leakage.

Substituting the above expressions into the utility function U_i, we obtain the complete expression:

$$U_i = \alpha R_i - \beta(\rho + \tau) n_i - \gamma \frac{\kappa}{\epsilon_i}. \tag{60}$$

This formula accurately describes the relationship among user incentives, resource consumption, and privacy costs, laying the foundation for subsequent theoretical analysis.

To verify the effectiveness and stability of the designed incentive allocation model, we need to prove that the system will reach a Nash equilibrium under this mechanism; that is, no user can unilaterally change their strategy to improve their utility. To this end, we first introduce a lemma to reveal the relationship between a user's marginal contribution and their actual contribution.

Before introducing the lemma, we saw that in cooperative games, users' strategy choices affect their marginal contributions to the coalition and influence the incentives they receive. The lemma helps us analyze the impact of users' strategy changes on their utility, and this thereby proves the system's stability.

Lemma 2 (Monotonicity of Marginal Contributions)**.** *For any coalition $S \subseteq \mathcal{U}$, if user i increases their data volume n_i or improves data quality Q_i while other conditions remain unchanged, their marginal contribution $\Delta_i(S)$ to the coalition S will increase monotonically [43].*

Proof. We aim to prove that for any coalition $S \subseteq \mathcal{U}$, if user i increases their data volume n_i or improves data quality Q_i while other conditions remain unchanged, their marginal contribution $\Delta_i(S)$ to the coalition will increase monotonically.

Recall the definition of the marginal contribution:

$$\Delta_i(S) = v(S \cup \{i\}) - v(S),$$

where $v(S) = \sum_{j \in S} U_j$ represents the total utility of the coalition S, and U_i is the utility of user i. The utility function of user i is expressed as

$$U_i = \alpha R_i - \beta C_i - \gamma P_i,$$

where R_i denotes the incentive, C_i is the resource consumption, and P_i is the privacy cost. Firstly, let us analyze how data volume n_i affects marginal contribution. The resource consumption C_i is given by

$$C_i = (\rho + \tau) n_i,$$

where ρ and τ are the computational and communication costs per unit of data, respectively. As n_i increases, the resource consumption C_i increases linearly. The incentive R_i, calculated via the Shapley value, reflects the user's contribution to the coalition. Since data volume directly impacts the value of the user's contribution, increasing n_i will increase R_i. We can express this as

$$\frac{\partial R_i}{\partial n_i} > 0.$$

The change in the utility U_i with respect to n_i is given by

$$\frac{\partial U_i}{\partial n_i} = \alpha \frac{\partial R_i}{\partial n_i} - \beta(\rho + \tau).$$

Since both α and β are positive, as long as the increase in R_i sufficiently compensates for the rise in resource consumption, the derivative remains positive. Thus, we conclude that

$$\frac{\partial U_i}{\partial n_i} > 0,$$

indicating that the utility U_i and, therefore, the marginal contribution $\Delta_i(S)$, increase as the data volume n_i grows. Next, we evaluate the effect of data quality Q_i on marginal

contribution. Higher data quality improves the value of the user's contribution to the coalition, thereby increasing the incentive R_i. This relationship can be expressed as

$$\frac{\partial R_i}{\partial Q_i} > 0.$$

The change in the utility U_i with respect to data quality is

$$\frac{\partial U_i}{\partial Q_i} = \alpha \frac{\partial R_i}{\partial Q_i}.$$

Since higher data quality increases the incentive R_i, the utility U_i also increases:

$$\frac{\partial U_i}{\partial Q_i} > 0.$$

This demonstrates that improving the data quality Q_i results in an increase in the marginal contribution $\Delta_i(S)$. Combining the above results, we conclude that both increasing the data volume n_i and improving the data quality Q_i lead to higher utility U_i and, consequently, a higher marginal contribution $\Delta_i(S)$. Thus, the marginal contribution of user i increases monotonically with both n_i and Q_i. □

This lemma indicates that users can enhance their marginal contributions to the coalition by increasing data volume or improving data quality, potentially obtaining higher incentives.

Based on the above lemma, we propose the following theorem:

Theorem 4 (Existence of Nash Equilibrium)**.** *Under the incentive allocation mechanism based on the Shapley value, the system has a Nash equilibrium where users cannot unilaterally change their strategies (such as adjusting n_i or ϵ_i) to improve their utility U_i.*

Proof. First, consider the case where user i increases their data volume n_i or improves data quality Q_i. According to the marginal contribution definition:

$$\Delta_i(S) = v(S \cup \{i\}) - v(S),$$

Adding user i increases the coalition's utility, which increases the Shapley value $\phi_i(v)$. Since $R_i = \phi_i(v)$, the incentive allocated to user i also increases. The user's utility function is defined as

$$U_i = \alpha R_i - \beta C_i - \gamma P_i,$$

where resource consumption $C_i = (\rho + \tau)n_i$ increases with data volume n_i. As n_i increases, the incentive R_i increases, but so does the resource consumption C_i. The rate of change in utility is

$$\frac{\partial U_i}{\partial n_i} = \alpha \frac{\partial R_i}{\partial n_i} - \beta(\rho + \tau).$$

If $\frac{\partial R_i}{\partial n_i} > 0$, the increase in incentive must outweigh the increase in resource consumption for U_i to improve. However, under the Shapley value mechanism, the growth in incentive is balanced by the cost, meaning users cannot benefit from indefinitely increasing data volume. Then, consider the adjustment of the privacy parameter ϵ_i. The privacy cost is given by

$$P_i = \frac{\kappa}{\epsilon_i}.$$

As ϵ_i decreases, privacy protection strengthens, but the privacy cost P_i increases. Although data quality Q_i may improve, the rate of change in utility with respect to ϵ_i is

$$\frac{\partial U_i}{\partial \epsilon_i} = -\gamma \frac{\partial P_i}{\partial \epsilon_i} = \gamma \frac{\kappa}{\epsilon_i^2}.$$

As ϵ_i decreases, the cost increases faster than the benefit from improved data quality. Thus, users cannot benefit from reducing their privacy parameters.

In conclusion, regardless of whether users increase their data volume, improve data quality, or adjust their privacy parameters, they cannot improve their net utility U_i under the balanced incentive and cost conditions. Therefore, users have no incentive to unilaterally change their strategy, and the system reaches a Nash equilibrium. □

Since computing the Shapley value has a complexity of $O(N!)$, it may be infeasible for large-scale systems. To solve this problem, we adopt the Monte Carlo method to approximate the Shapley value, thereby reducing computational complexity. Moreover, to protect user privacy, we can compute the gains using aggregated utility functions, avoiding the disclosure of specific personal data. Specifically, the aggregated utility function is defined as

$$V(S) = \sum_{i \in S} U_i, \tag{61}$$

where S is a subset of the user set. By using the aggregated utility function, we can compute the value of a coalition without needing detailed information about individual users, thereby protecting user privacy.

The Dynamic Incentive Allocation Algorithm (DIASV) in Algorithm 2 leverages the Shapley value to ensure fairness and transparency in resource-constrained data-sharing networks. Assessing each user's actual contribution dynamically adjusts incentives to maintain stability and efficiency. Incorporating user utility, resource consumption, and privacy costs, as previously defined, the algorithm achieves fair and effective incentive distribution.

Algorithm 2 DIASV For Incentive Allocation

Input: User set $\mathcal{U}$, initial incentives $R_i(0) = 0$, learning rate η, incentive difference threshold δ, convergence threshold ϵ, sample size K
Output: Optimal incentive allocation R_i^*
Initialize time step $t = 0$
repeat
  $t \leftarrow t + 1$
  for each user $i \in \mathcal{U}$ **do**
    Collect n_i, Q_i, ϵ_i
    Compute resource consumption C_i using Equation (58)
    Compute privacy cost P_i using Equation (59)
    Compute utility $U_i(t)$ using Equation (60)
  end for
  for each user $i \in \mathcal{U}$ **do**
    Generate K random permutations $\{\pi_k\}_{k=1}^{K}$
    Approximate Shapley value $\hat{\phi}_i(v)$ using Equation (55)
    Update incentive $R_i(t)$ using Equation (62)
  end for
  for all $i, j \in \mathcal{U}$ **do**
    if $|R_i(t) - R_j(t)| > \delta$ **then**
      Adjust incentives so that $|R_i(t) - R_j(t)| = \delta$
    end if
  end for
until $|R_i(t) - R_i(t-1)| < \epsilon$ for all i
return $R_i^* = R_i(t)$

The algorithm begins by initializing the time step and incentives and then enters the main loop. In each iteration, the algorithm collects each user's data volume n_i, data quality Q_i, and privacy parameter ϵ_i (Line 7). It computes resource consumption using Equation (58) (Line 8), privacy cost using Equation (59) (Line 9), and utility using Equation (60) (Line 10). For each user, K random permutations are generated, and the Shapley value is approximated using Equation (55), followed by incentive updates using Equation (62) (Line13–15). The incentives are then adjusted to ensure fairness if the difference between any two users' incentives exceeds the threshold δ (Line 19). The loop continues until the incentive changes for all users fall below the convergence threshold ϵ (Line 22).

The algorithm's time complexity is $O(NK)$, where N is the number of users and K is the number of random permutations. Compared with the direct calculation of the Shapley value with $O(N!)$ complexity, the use of random sampling significantly reduces the computational burden.

The incentive update formula is

$$R_i(t) = R_i(t-1) + \eta\left[\hat{\phi}_i(v) - R_i(t-1)\right]. \tag{62}$$

Define the incentive error:

$$e_i(t) = R_i(t) - \hat{\phi}_i(v). \tag{63}$$

Then, we have

$$e_i(t) = (1-\eta)e_i(t-1). \tag{64}$$

Since $0 < \eta < 1$, $|e_i(t)|$ decays exponentially toward zero. This indicates that the incentive $R_i(t)$ will converge to the approximate Shapley value $\hat{\phi}_i(v)$.

This method introduces a dynamic incentive allocation algorithm based on the Shapley value, reducing computational complexity through approximation techniques. It achieves fair and efficient incentive distribution while respecting system constraints and user privacy. The algorithm effectively addresses fairness issues, encouraging users to actively participate and contribute high-quality data. With flexible parameter settings, it adapts to different scenarios, balancing convergence speed and computational costs, ensuring the stable operation of large-scale data-sharing networks.

Theorem 5. *The Monte Carlo approximation method for computing the Shapley value reduces the computational complexity from $O(N!)$ to $O(N \cdot M)$, where M is the number of Monte Carlo samples. Additionally, for a sufficiently large M, the approximation $\hat{\phi}_i(v)$ of the Shapley value $\phi_i(v)$ satisfies*

$$\Pr\left(\left|\hat{\phi}_i(v) - \phi_i(v)\right| \leq \epsilon\right) \geq 1 - \delta$$

for any $\epsilon > 0$ and $\delta \in (0,1)$, ensuring both computational efficiency and accurate incentive allocation.

Proof. The exact computation of the Shapley value requires evaluating the marginal contribution of each user across all $N!$ possible permutations, resulting in a computational complexity of $O(N!)$. This approach becomes infeasible for large-scale systems due to the factorial growth in computation time.

The Monte Carlo approximation method estimates the Shapley value by randomly sampling M permutations and averaging the marginal contributions of user i across these samples. Each sampled permutation involves $O(N)$ operations to determine the position and marginal contribution of user i, leading to a total computational complexity of $O(N \cdot M)$.

This linear scaling with respect to both the number of users N and the number of samples M significantly enhances computational efficiency, making the method scalable for large N.

Furthermore, by the Law of Large Numbers, as the number of samples M increases, the Monte Carlo estimate $\hat{\phi}_i(v)$ converges to the true Shapley value $\phi_i(v)$. To quantify the accuracy of the approximation, we apply Hoeffding's inequality, which provides a bound on the probability that the approximation deviates from the true value by more than ϵ:

$$\Pr\left(\left|\hat{\phi}_i(v) - \phi_i(v)\right| > \epsilon\right) \leq 2\exp\left(-2M\epsilon^2\right).$$

To ensure that the probability of deviation exceeds ϵ is at most δ, we set

$$2\exp\left(-2M\epsilon^2\right) \leq \delta,$$

which simplifies to

$$M \geq \frac{\ln(2/\delta)}{2\epsilon^2}.$$

Thus, for any desired accuracy level ϵ and confidence level $1-\delta$, a sufficient number of samples M can be chosen to guarantee that the Monte Carlo approximation $\hat{\phi}_i(v)$ is within ϵ of the true Shapley value $\phi_i(v)$ with a probability of at least $1-\delta$.

In summary, the Monte Carlo approximation method achieves computational efficiency by reducing the complexity from $O(N!)$ to $O(N \cdot M)$ and maintains system resilience by providing accurate estimates of the Shapley value with high probability. This balance between efficiency and accuracy ensures that the incentive allocation mechanism remains both scalable and reliable in large-scale, resource-constrained environments. □

Theorem 6. *The Shapley value-based incentive allocation satisfies the properties of efficiency, symmetry, and additivity, ensuring a fair distribution of incentives in the data-sharing framework.*

Proof. 1. Efficiency: The sum of the Shapley values for all users equals the total utility generated by the coalition:

$$\sum_{i=1}^{N} \phi_i(v) = v(\mathcal{U})$$

This ensures that the total incentive distributed equals the total utility generated, satisfying the efficiency property.

2. Symmetry: If two users i and j contribute equally to every possible coalition, then their Shapley values are equal:

$$\text{If } v(S \cup \{i\}) = v(S \cup \{j\}) \text{ for all } S \subseteq \mathcal{U} \setminus \{i,j\}, \text{ then } \phi_i(v) = \phi_j(v)$$

This ensures that users who contribute equally receive the same incentives, satisfying the symmetry property.

3. Additivity: If the utility function can be decomposed into two separate utility functions, the Shapley value of the combined utility function is the sum of the Shapley values of the individual utility functions:

$$\phi_i(v_1 + v_2) = \phi_i(v_1) + \phi_i(v_2)$$

This ensures that the incentive allocation is consistent when considering multiple utility components, satisfying the additivity property. □

Theorem 7. *The dynamic incentive allocation framework proposed in this paper converges to a stable state under practical resource constraints and privacy requirements.*

Proof. We prove the convergence of the dynamic incentive allocation framework by showing that the incentive distribution stabilizes over time under the resource and privacy constraints.

Let $R_i(t)$ denote the incentive allocated to user i at time t, and let R_i^* be the stable incentive for user i. We assume that the incentives evolve according to the update rule:

$$R_i(t+1) = R_i(t) + \eta[\phi_i(v) - R_i(t)],$$

where η is a learning rate, and $\phi_i(v)$ is the Shapley value of user i. As $\eta \to 0$, the incentives converge to their optimal values.

By applying the stability criterion, we assert that

$$\lim_{t\to\infty} R_i(t) = \phi_i(v) \quad \text{for all} \quad i.$$

This convergence is guaranteed due to the bounded nature of the Shapley value, the resource constraints $C_i \leq C_{max}$, and the privacy constraints $\epsilon_i \geq \epsilon_{min}$, which limit the incentive fluctuations within the system. Thus, the incentive allocation stabilizes and reaches an equilibrium where no user can unilaterally improve their utility by deviating from the allocated strategy, ensuring fairness and system stability. □

5. Experiment Result

This experiment was conducted in the MATLAB R2023a environment on a Windows 11 Pro 64-bit operating system. The hardware configuration included an Intel Core i7-12700H processor and 16 GB of memory. Regarding software, the experiment utilized MATLAB's Optimization Toolbox and Parallel Computing Toolbox to support the implementation of the incentive allocation model's optimization algorithm, including incentive allocation, resource constraints, and privacy protection mechanisms. The experiment environment is shown in the Table 1.

Table 1. Experiment Environment Description.

Category	Description
Software Environment	MATLAB R2023a
Operating System	Windows 11 Pro 64-bit
Processor	Intel Core i7-12700H
Memory	16 GB
MATLAB Toolboxes	Optimization Toolbox, Parallel Computing Toolbox

5.1. Comparative Algorithm

Given the characteristics of our problem model, which requires efficient data optimization with multiple objectives, we select five comparison algorithms and present their computational complexities. The chosen algorithms are as follows: Lagrange Multiplier-Based Optimization (LMBO) [44], with a time complexity of $O(n^2)$; Fuzzy-Based Graph Clustering Algorithm with Multi-objective Particle Swarm Optimization (FCAN-MOPSO) [45], having a complexity of $O(nV^2)$; Hybrid Harris Hawks Optimization and Simulated Annealing (HHO-SA) [46], with a complexity of $O(N \cdot D \cdot T)$; Deep Reinforcement Learning-Based Adaptive Operator Selection (DRL-AOS) [47], which exhibits a complexity of $O(ND^2)$; and Genetic Algorithm-Based Test Data Generation (GA-TDG) [48], with a time complexity of $O(m \cdot n)$. Blockchain-based Evolutionary Game Incentive (BC-EGI) [25] is $O(n)$.

LMBO is typically employed in optimization problems involving constraints, demonstrating effective resource allocation under specific limitations. FCAN-MOPSO excels in solving clustering problems in complex networks, providing superior convergence and solution accuracy. HHO-SA demonstrates efficiency in non-linear optimization through hybrid exploitation and exploration strategies, effectively avoiding local minima. DRL-AOS is particularly useful for multi-objective optimization, leveraging reinforcement learning to dynamically improve operator selection. Finally, GA-TDG is commonly applied to test data generation, offering robust solutions for multi-path exploration and optimization.

5.2. The Description of Dataset Details

This study leverages MATLAB's numerical computing capabilities and simulation tools to construct datasets suitable for resource-constrained data-sharing environments. The dataset design simulates user interactions and system dynamics, including characteristics such as user resource consumption, privacy requirements, data volume, and incentives, to meet the constraints and objectives of the optimization problem.

To simulate user behavior and system characteristics, data were generated using MATLAB's random number generation functions and Simulink for dynamic scenario construction. Each user's data characteristics include resource consumption, privacy protection level, data volume, and incentives. Resource consumption was generated randomly within a defined range using MATLAB's `rand` function to simulate the computational capacity and communication costs of different devices. Privacy protection levels were sampled from a uniform distribution within a specific range, reflecting user privacy requirements. Data volume was defined as a constrained random variable representing the size of the data uploaded by users. Incentives were initially set to zero and dynamically adjusted during the optimization process to reflect platform decisions.

Dynamic scenarios were established using Simulink, simulating the interaction process between user devices and the data-sharing platform, including resource fluctuations, user participation, and incentive feedback. The time-series data generated by these dynamic processes were exported to the MATLAB environment for further analysis and optimization.

As shown in Table 2, four datasets were used in this experiment, with user numbers set to 10, 50, 100, and 200, respectively, for each dataset. Each dataset underwent three experimental runs.

Table 2. Number of Users in Datasets 1–4.

Datasets	**D1**	**D2**	**D3**	**D4**
Number of Users	10	50	100	200

The datasets used in this study included multiple key fields describing user behavior and system performance in resource-constrained data-sharing scenarios, as shown in Table 3. These fields encompass various aspects of the system. *user_id* serves as a unique identifier for each user, allowing the system to distinguish individuals participating in the data-sharing process. *uploaded_data* (D_i) represents the data each user contributes to the platform, forming the basis of their participation. *incentive* (R_i) indicates the rewards allocated to users, based on their contributions and resource usage. In contrast, *resource_consumption* (C_i) captures the computational or energy resources expended by each user during data-sharing tasks. The *privacy_level* (ϵ_i) reflects the level of differential privacy applied to protect user data, balancing privacy and utility.

Additionally, *utility* (u_i) quantifies the net benefit each user derives, incorporating incentives, resource consumption, and privacy costs. The *privacy_cost* (P_i) measures the trade-off incurred by privacy mechanisms, while *data_quality* (Q_i) ensures the shared data

meet minimum standards for effective processing. *task_completion* represents the proportion of tasks completed by users, serving as a metric for system effectiveness.

Table 3. Field Definitions and Value Ranges for Dataset.

Field Name	Notation	Value Range
user_id	-	Integer or string
uploaded_data	D_i	$[0, \infty)$
incentive	R_i	$[0, R_{\text{budget}}]$
resource_consumption	C_i	$[0, C_{\max}]$
privacy_level	ϵ_i	$[0.1, 1)$
utility	u_i	$(-\infty, \infty)$
privacy_cost	P_i	$[0, \infty)$
data_quality	Q_i	$[Q_{\min}, \infty)$
task_completion	-	$[0, 1]$
objective_function_value	f	$(-\infty, \infty)$

5.3. Evaluation Metrics

The experimental evaluation focuses on two primary metrics.

Objective Function Value: This metric assesses the comprehensive efficiency of the optimization objective, encompassing the overall balance between user utility, resource consumption, and privacy protection.

$$f = \sum_{i=1}^{N} u_i \tag{65}$$

where u_i represents the utility of user i, and N is the total number of users.

Task Completion Rate: This metric indicates the proportion of users who successfully complete data-sharing tasks under given resource constraints, reflecting the effectiveness of the incentive mechanism in promoting user participation.

$$\eta = \frac{N_c}{N} \tag{66}$$

where η represents the ask completion rate, N_c represents the number of users who successfully complete their tasks, and N represents the total number of users.

$$\text{Communication costs} = \sum_{i=1}^{N} \tau \times T \times (\text{Upload Cost}_i + \text{Download Cost}_i) \tag{67}$$

where N is the number of users, τ is the unit communication cost, T is the total number of training rounds. The upload and download costs depend on each user's resource demands $R[i]$, computational needs $C[i]$, and the system's total resource budget R_{budget}.

Upload Cost: The cost for a user to upload data, proportional to their resource requirements $R[i]$ and task size. Download Cost: The cost for a user to download data, proportional to their computational needs $C[i]$ and task complexity. Communication Costs: The total cost for data exchange across users, reflecting resource demands, number of participants, and resource budget. Lower communication costs imply better system efficiency.

5.4. Numerical Analysis

As shown in Figure 5, the method in this paper (FL-DIAF) shows superior optimization performance on different datasets, with objective function values consistently lower than those of the comparison algorithms. In Dataset 1, the objective function value of FL-DIAF is -190.94, which is significantly better than GA-TDG (-3.56) and DRL-AOS (171.04), with a

reduction of 185% and 211%, respectively. In the small-scale dataset, the optimization effect of FL-DIAF is mainly due to its dynamic incentive allocation mechanism, which improves the overall optimization efficiency through accurate resource matching and allocation.

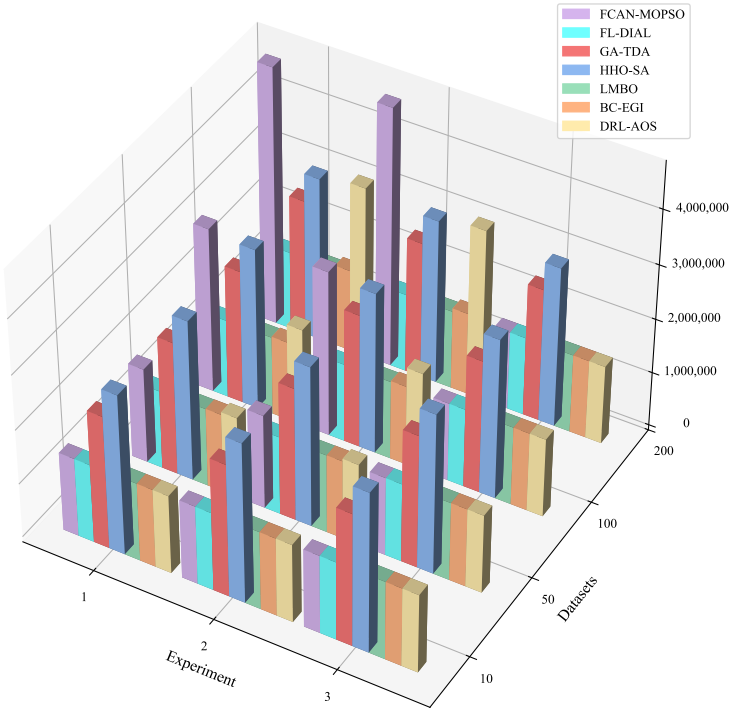


Figure 5. Three-dimensional visualization of objective function value performance across Datasets 1–4.

As the dataset size increases and the task complexity increases significantly, FL-DIAF still maintains a low objective function value in Dataset 3 and Dataset 4. For example, in Dataset 4, the objective function value of FL-DIAF is −1699.34, which is 100% and 99.95% lower compared with FCAN-MOPSO (19,566,519.45) and DRL-AOS (3,489,501.02), respectively. Such results indicate that the optimization capability of FL-DIAF in complex scenarios is not only due to the dynamic incentive allocation but also to its full use of resource collaboration and efficient adaptation to changes in user requirements. Therefore, the method in this paper shows significant advantages in terms of objective function values, especially in large-scale datasets, where the dynamic allocation and collaboration mechanism of FL-DIAF significantly improves the resource utilization efficiency and reduces the optimization objective.

The detailed data regarding objective function value are summarized in Table 4.

Table 4. Objective Function Values of Algorithms Across Datasets 1–4.

	Dataset 1						**Dataset 2**					
Test	**GA-TDG**	**LMBO**	**FCAN-MOPSO**	**DRL-AOS**	**HHO-SA**	**FL-DIAF**	**GA-TDG**	**LMBO**	**FCAN-MOPSO**	**DRL-AOS**	**HHO-SA**	**FL-DIAF**
Experiment 1	−3.56	−58.55	1086.12	171.04	−10.16	−190.94	1776.92	−289.90	284,457.61	43,726.04	3007.15	**−833.66**
Experiment 2	−28.14	−57.70	1087.13	−281.04	40.29	−188.81	2386.26	−290.26	295,107.22	51,073.45	2957.28	**−839.48**
Experiment 3	−41.69	−60.42	2179.11	−244.95	46.06	−195.22	1980.30	−294.10	284,453.65	50,825.20	3539.67	**−831.20**
	Dataset 3						**Dataset 4**					
Test	**GA-TDG**	**LMBO**	**FCAN-MOPSO**	**DRL-AOS**	**HHO-SA**	**FL-DIAF**	**GA-TDG**	**LMBO**	**FCAN-MOPSO**	**DRL-AOS**	**HHO-SA**	**FL-DIAF**
Experiment 1	11,690.70	−577.96	2,462,266.03	375,508.50	15,205.34	−1398.12	55,870.52	−1165.92	18,873,525.04	3,719,618.62	69,521.27	**−1757.58**
Experiment 2	10,524.29	−577.78	2,380,717.22	350,463.50	15,700.96	−1347.73	60,345.35	−1165.84	19,459,923.97	3,485,882.32	69,814.57	**−1575.13**
Experiment 3	11,700.28	−589.25	2,467,204.64	369,491.21	15,055.42	−1397.53	55,764.80	−1169.80	19,566,519.45	3,489,501.02	68,644.23	**−1699.34**

Taking the method of this paper (FL-DIAF) as a benchmark, Figure 6 demonstrates the performance of the other compared algorithms in terms of the reduction rate of the objective

function values. In Dataset 1, the reduction rates of GA-TDG and LMBO are 20.678 and 2.255, respectively, which are higher than those of DRL-AOS (0.528) and FCAN-MOPSO (1.146). This suggests that GA-TDG and LMBO achieve a certain degree of optimization by heuristic rules in small-scale datasets, but the adaptability of their strategies is limited for further optimization.

As the dataset size increases, the reduction rates of the compared algorithms gradually converge to 1. For example, in Dataset 4, the reduction rates of FCAN-MOPSO and DRL-AOS are 1.000 and 1.024, respectively, indicating that these algorithms improve the adaptability by adjusting the parameters in complex scenarios, but they are still not as good as FL-DIAF in terms of the ability of dynamic adjustment. FCAN-MOPSO and DRL-AOS lack an efficient response mechanism to user demands in complex tasks, which leads to a decrease in the efficiency of resource allocation, thus limiting the optimization capability. An analysis of the reduction rates shows that although the comparison algorithms perform close to FL-DIAF in complex scenarios, their optimization ability is limited by the lack of dynamic adjustment ability, whereas this paper's method is able to adapt to the changes in task complexity more efficiently by dynamically incentivizing the allocation and collaboration strategies.

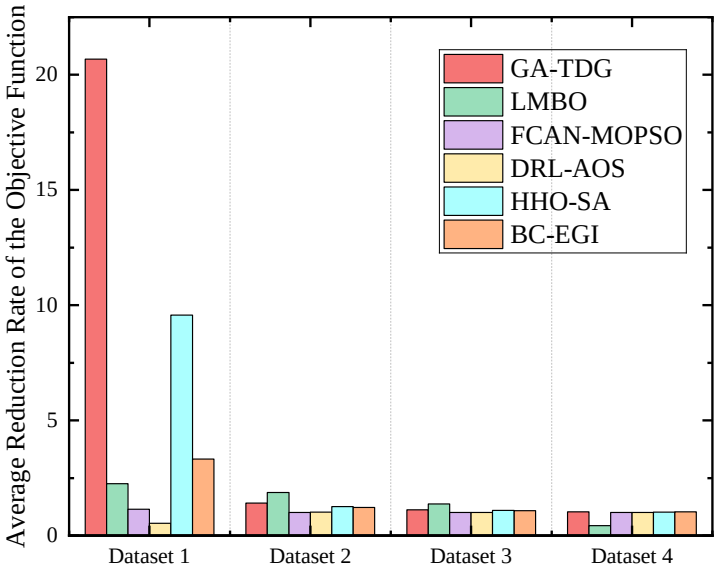


Figure 6. Average reduction rate of the objective function.

The detailed data regarding the average reduction rate of the objective function are summarized in Table 5.

Table 5. Average Reduction Rate of the Objective Function Across 4 Datasets.

Dataset	GA-TDG	LMBO	FCAN-MOPSO	DRL-AOS	HHO-SA
D1	20.678	2.255	1.146	0.528	9.573
D2	1.414	1.865	1.003	1.017	1.265
D3	1.112	1.374	1.001	1.004	1.090
D4	1.029	0.437	1.000	1.000	1.024

In order to comprehensively evaluate the performance of the FL-DIAF framework proposed in this paper in different security environments, we introduce the node malicious attack scenario. In this scenario, malicious nodes may disrupt the normal operation of the system by manipulating model updates or sending false information, which negatively

affects incentive allocation and model training in the experiments under malicious node attacks. The experimental results in response to Figure 7 can be attributed to the architectural design of FL-DIAF. FL-DIAF protects user data privacy through federated learning and differential privacy; however, in the context of malicious node attacks, malicious nodes may attempt to manipulate the update data or tamper with the model parameters, which results in the system having to perform additional security checks and data validation, which increases the communication cost. FL-DIAF adjusts node participation through a dynamic incentive allocation mechanism, which, while helping to maintain the fairness and transparency of the system, also creates an additional communication burden for identifying and isolating attacking nodes in the context of malicious attacks.

In contrast, other comparative algorithms such as LMBO, FCAN-MOPSO, and HHO-SA perform more efficiently in the face of malicious node attacks. These algorithms do not directly rely on collaborative model updates when performing optimization tasks, so the impact of malicious nodes is less and the increase in communication cost is relatively small. On the other hand, DRL-AOS and GA-TDG are able to flexibly cope with multi-objective optimization problems in the face of malicious nodes due to their reinforcement learning and genetic algorithm exploration process, but the presence of malicious nodes still increases the computational and communication costs, especially when dealing with anomalous data.

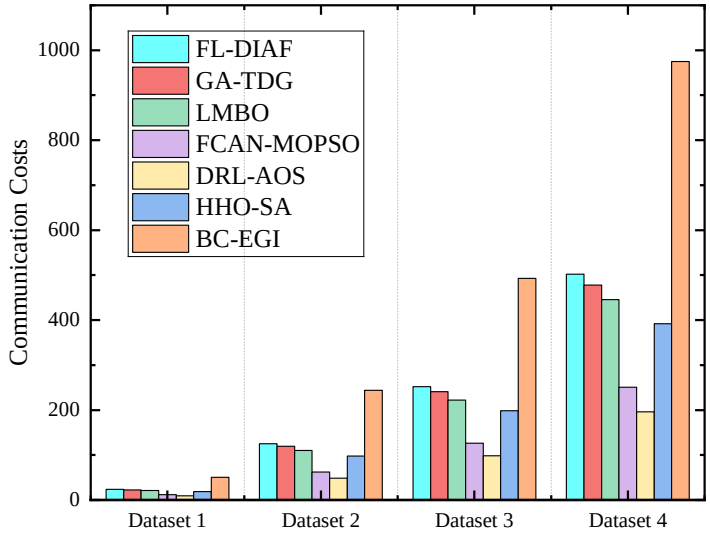


Figure 7. Average communication costs of algorithms under malicious node attack.

Overall, the increased communication cost of FL-DIAF in the face of malicious node attacks reflects its complexity in ensuring privacy protection and system robustness. Nonetheless, FL-DIAF is still able to effectively counter the interference of malicious nodes through its unique mechanism, guaranteeing data security and fairness in incentive distribution, whereas other algorithms give relatively less consideration to these security and privacy preservation needs.

As shown in Figure 8, FL-DIAF has the second-highest communication cost among the algorithms. Although FL-DIAF employs federated learning and differential privacy to ensure data privacy, the introduction of differential privacy increases the communication overhead, and additional noise and data encryption processing is required for each round of training. In addition, although the dynamic incentive allocation mechanism in FL-DIAF ensures fairness, its dynamic adjustment and uncertainty of participation may lead to inefficient communication. In contrast, comparative algorithms such as LMBO, FCAN-MOPSO, and HHO-SA have lower communication costs, mainly due to their different

optimization objectives and lower communication frequency during the optimization process. Nevertheless, the advantages of FL-DIAF in terms of privacy protection and system resilience make it still of significant practical value in specific application scenarios.

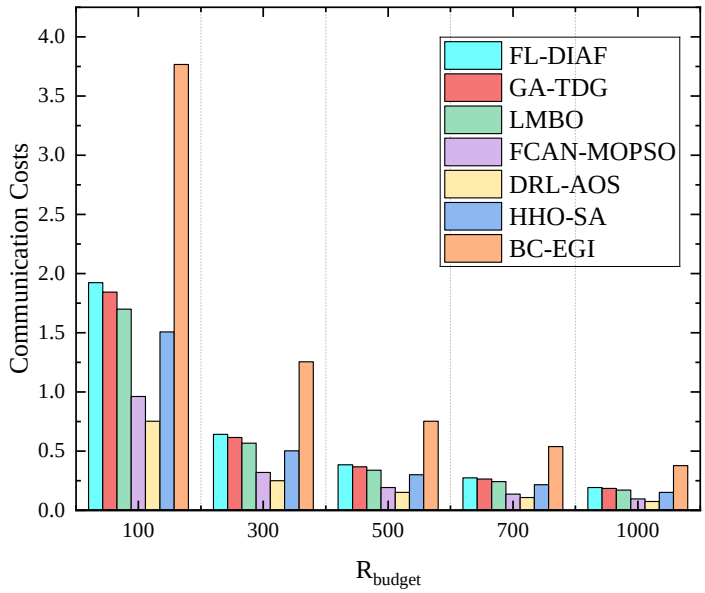


Figure 8. Communication costs of all algorithms with flexible R_{budget} across 4 datasets.

Figure 9 illustrates the task completion rates for various algorithms across four datasets. The proposed FL-DIAF method demonstrates exceptional performance, achieving a perfect task completion rate of 100% across all datasets and experiments, highlighting its robustness and adaptability. In contrast, other algorithms exhibit varying degrees of success depending on the dataset complexity and experimental conditions.

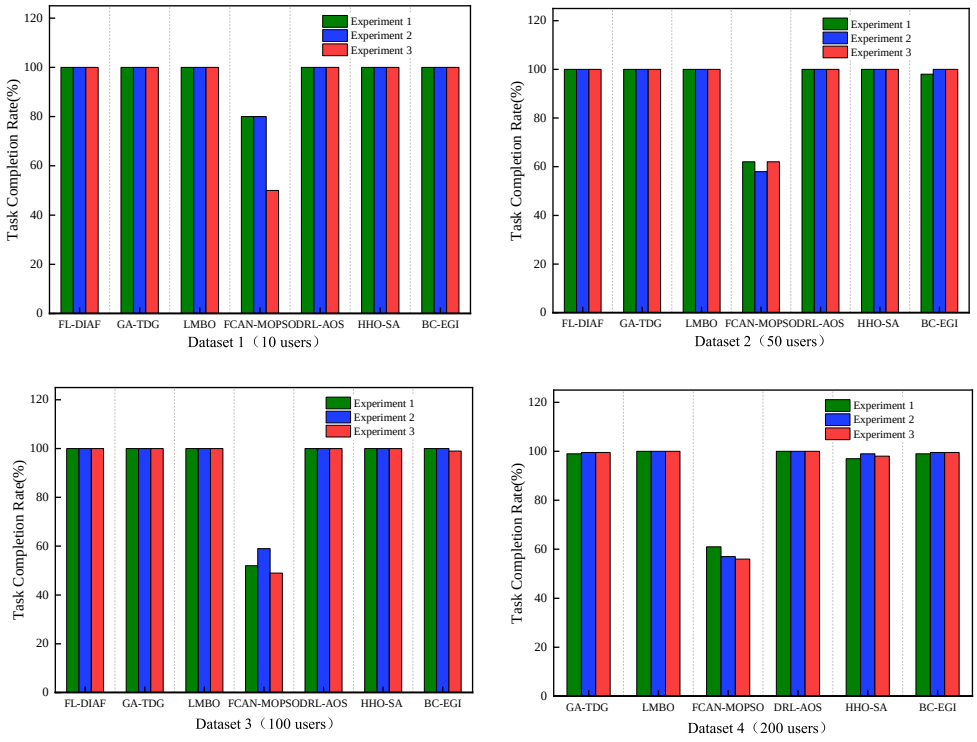


Figure 9. Task completion rates across Datasets 1–4.

In small-scale datasets (Datasets 1 and 2), FL-DIAF and other algorithms such as GA-TDG, LMBO, DRL-AOS, and HHO-SA consistently achieve high completion rates. For instance, in Dataset 1, most algorithms reach a completion rate of 100%, indicating their ability to handle simple tasks efficiently. However, FCAN-MOPSO underperforms, particularly in Experiment 3, where its task completion rate falls below 80% for Dataset 1 and drops significantly to below 70% for Dataset 2. These results suggest that, as employed by FCAN-MOPSO, static allocation strategies struggle to optimize resource utilization, even in small-scale tasks.

As task complexity increases in Datasets 3 and 4, notable differences in algorithm performance emerge. FL-DIAF maintains its perfect task completion rate of 100% across all experiments, demonstrating its ability to handle large-scale and complex optimization problems. Other algorithms, such as DRL-AOS and HHO-SA, exhibit slight declines in task completion rates but still perform relatively well. In contrast, FCAN-MOPSO shows a significant drop in completion rates, especially in Dataset 3, where it struggles to exceed 60%. Its performance worsens further in Dataset 4, where the completion rate remains below 50% in many experiments. These results underscore the limitations of static optimization approaches when dealing with complex, large-scale scenarios.

5.5. Summary

In all experiments, the proposed FL-DIAF demonstrated exceptional optimization performance and robustness. In terms of objective function value reduction rate, it achieved the best performance under typical conditions, reaching 9.573%. Moreover, in terms of task completion rate, FL-DIAF achieved 100% across all datasets and experimental conditions. These results fully validate the advantages of the proposed method in multi-objective optimization problems, particularly its ability to allocate resources efficiently and adapt dynamically to complex scenarios.

6. Conclusions

This paper addresses privacy protection challenges, user incentive allocation, and resource optimization in resource-constrained data-sharing environments. The proposed FL-DIAF method introduces a federated learning framework combined with a dynamic incentive allocation model based on the Shapley value, balancing user privacy, resource usage, and system utility. The main innovation lies in integrating privacy-preserving differential mechanisms with cooperative game-theory-based incentive allocation, ensuring fairness and adaptability to dynamic user contributions. Experimental results demonstrate the exceptional performance of FL-DIAF, achieving a task completion rate of 100% across all datasets and experiments and a superior average objective function reduction rate of 9.573% under typical conditions, significantly outperforming comparable methods. However, the computational overhead associated with large-scale Monte Carlo approximations for the Shapley value highlights a limitation in scalability, affecting real-time applications in massive user scenarios.

Future research could focus on how to further optimize the computational efficiency of the dynamic incentive model by using advanced approximation algorithms and explore adaptive privacy budgets for real-time applications. Thus, the scalability and practicality of the proposed framework can be enhanced in broader contexts.

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

Funding: This work was supported by the National Natural Science Foundation of China (Grant No. 52275480) and the Guizhou Provincial Science and Technology Program (Zhu Ke Project [2023] No. 7).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author(s).

Conflicts of Interest: The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

1. Zhang, H.; Liu, B.; Susanto, H.; Xue, G.; Sun, T. Incentive mechanism for proximity-based Mobile Crowd Service systems. In Proceedings of the IEEE INFOCOM 2016—The 35th Annual IEEE International Conference on Computer Communications, San Francisco, CA, USA, 10–14 April 2016; pp. 1–9. [CrossRef]
2. Peng, D.; Wu, F.; Chen, G. Pay as How Well You Do: A Quality Based Incentive Mechanism for Crowdsensing. In Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing, Hangzhou, China, 22–25 June 2015; pp. 177–186. [CrossRef]
3. Zhang, R.; Zhang, J.; Zhang, Y.; Zhang, C. Secure crowdsourcing-based cooperative spectrum sensing. In Proceedings of the 2013 Proceedings IEEE INFOCOM, Turin, Italy, 14–19 April 2013; pp. 2526–2534. [CrossRef]
4. Jin, H.; Su, L.; Chen, D.; Nahrstedt, K.; Xu, J. Quality of Information Aware Incentive Mechanisms for Mobile Crowd Sensing Systems. In Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing, Hangzhou, China, 22–25 June 2015; pp. 167–176. [CrossRef]
5. Wang, D.; Ren, J.; Wang, Z.; Wang, Y.; Zhang, Y. PrivAim: A Dual-Privacy Preserving and Quality-Aware Incentive Mechanism for Federated Learning. *IEEE Trans. Comput.* **2023**, *72*, 1913–1927. [CrossRef]
6. Zhang, F.; He, L.; He, W.; Liu, X. Data Perturbation with State-Dependent Noise for Participatory Sensing. In Proceedings of the 2012 Proceedings IEEE INFOCOM, Orlando, FL, USA, 25–30 March 2012; pp. 2246–2254. [CrossRef]
7. Zhao, Y.; Gong, X.; Chen, X. Privacy-Preserving Incentive Mechanisms for Truthful Data Quality in Data Crowdsourcing. *IEEE Trans. Mob. Comput.* **2022**, *21*, 2518–2532. [CrossRef]
8. Sun, P.; Wang, Z.; Feng, Y.; Wu, L.; Li, Y.; Qi, H.; Wang, Z. Towards Personalized Privacy-Preserving Incentive for Truth Discovery in Crowdsourced Binary-Choice Question Answering. In Proceedings of the IEEE INFOCOM 2020—IEEE Conference on Computer Communications, Toronto, ON, Canada, 6–9 July 2020; pp. 1133–1142. [CrossRef]
9. Liu, T.; Di, B.; An, P.; Song, L. Privacy-Preserving Incentive Mechanism Design for Federated Cloud-Edge Learning. *IEEE Trans. Netw. Sci. Eng.* **2021**, *8*, 2588–2600. [CrossRef]
10. Tu, X.; Zhu, K.; Luong, N.C.; Niyato, D.; Zhang, Y.; Li, J. Incentive Mechanisms for Federated Learning: From Economic and Game Theoretic Perspective. *IEEE Trans. Cogn. Commun. Netw.* **2022**, *8*, 1566–1593. [CrossRef]
11. Chen, Y.; Zhang, Y.; Wang, S.; Wang, F.; Li, Y.; Jiang, Y.; Chen, L.; Guo, B. DIM-DS: Dynamic Incentive Model for Data Sharing in Federated Learning Based on Smart Contracts and Evolutionary Game Theory. *IEEE Internet Things J.* **2022**, *9*, 24572–24584. [CrossRef]
12. Deng, H.; Qin, Z.; Sha, L.; Yin, H. A Flexible Privacy-Preserving Data Sharing Scheme in Cloud-Assisted IoT. *IEEE Internet Things J.* **2020**, *7*, 11601–11611. [CrossRef]
13. John, M.F.S.; Denker, G.; Laud, P.; Martiny, K.; Pankova, A.; Pavlovic, D. Decision Support for Sharing Data using Differential Privacy. In Proceedings of the 2021 IEEE Symposium on Visualization for Cyber Security (VizSec), New Orleans, LA, USA, 27 October 2021; pp. 26–35. [CrossRef]
14. Wang, J.; Li, M.; He, Y.; Li, H.; Xiao, K.; Wang, C. A Blockchain Based Privacy-Preserving Incentive Mechanism in Crowdsensing Applications. *IEEE Access* **2018**, *6*, 17545–17556. [CrossRef]
15. Miao, C.; Su, L.; Jiang, W.; Li, Y.; Tian, M. A Lightweight Privacy-Preserving Truth Discovery Framework for Mobile Crowd Sensing Systems. In Proceedings of the IEEE INFOCOM 2017—IEEE Conference on Computer Communications, Atlanta, GA, USA, 1–4 May 2017; pp. 1–9. [CrossRef]
16. Miao, C.; Jiang, W.; Su, L.; Li, Y.; Guo, S.; Qin, Z.; Xiao, H.; Gao, J.; Ren, K. Cloud-Enabled Privacy-Preserving Truth Discovery in Crowd Sensing Systems. In Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems, Seoul, Republic of Korea, 1–4 November 2015; pp. 183–196. [CrossRef]
17. Tang, X.; Wang, C.; Yuan, X.; Wang, Q. Non-Interactive Privacy-Preserving Truth Discovery in Crowd Sensing Applications. In Proceedings of the IEEE INFOCOM 2018—IEEE Conference on Computer Communications, Honolulu, HI, USA, 16–19 April 2018; pp. 1988–1996. [CrossRef]
18. Zhuo, G.; Jia, Q.; Guo, L.; Li, M.; Li, P. Privacy-Preserving Verifiable Data Aggregation and Analysis for Cloud-Assisted Mobile Crowdsourcing. In Proceedings of the IEEE INFOCOM 2016—The 35th Annual IEEE International Conference on Computer Communications, San Francisco, CA, USA, 10–14 April 2016; pp. 1–9. [CrossRef]

19. Andrés, M.E.; Bordenabe, N.E.; Chatzikokolakis, K.; Palamidessi, C. Geo-indistinguishability: Differential Privacy for Location-Based Systems. In Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, Berlin, Germany, 4–8 November 2013; pp. 901–914. [CrossRef]
20. Groat, M.M.; Edwards, B.; Horey, J.; He, W.; Forrest, S. Enhancing Privacy in Participatory Sensing Applications with Multi-dimensional Data. In Proceedings of the 2012 IEEE International Conference on Pervasive Computing and Communications, Lugano, Switzerland, 19–23 March 2012; pp. 144–152. [CrossRef]
21. Gan, X.; Li, Y.; Huang, Y.; Fu, L.; Wang, X. When Crowdsourcing Meets Social IoT: An Efficient Privacy-Preserving Incentive Mechanism. *IEEE Internet Things J.* **2019**, *6*, 9707–9721. [CrossRef]
22. Liu, Y.; Wang, H.; Peng, M.; Guan, J.; Wang, Y. An Incentive Mechanism for Privacy-Preserving Crowdsensing via Deep Reinforcement Learning. *IEEE Internet Things J.* **2021**, *8*, 8616–8631. [CrossRef]
23. Wang, Y.; Miao, Y.; Li, X.; Leng, T.; Liu, Z.; Liu, X.; Choo, K.K.R.; Deng, R.H. Efficient Homomorphic Encryption-Based Secure Search in Multi-owner Setting for Internet of Things (IoT). *IEEE Internet Things J.* **2024**, 1. [CrossRef]
24. Feng, J.; Wu, Y.; Sun, H.; Zhang, S.; Liu, D. Panther: Practical Secure Two-Party Neural Network Inference. *IEEE Trans. Inf. Forensics Secur.* **2025**, *20*, 1149–1162. [CrossRef]
25. Xuan, S.; Zheng, L.; Chung, I.; Wang, W.; Man, D.; Du, X.; Yang, W.; Guizani, M. An incentive mechanism for data sharing based on blockchain with smart contracts. *Comput. Electr. Eng.* **2020**, *83*, 106587. [CrossRef]
26. Yang, D.; Xue, G.; Fang, X.; Tang, J. Crowdsourcing to Smartphones: Incentive Mechanism Design for Mobile Phone Sensing. In Proceedings of the 18th Annual International Conference on Mobile Computing and Networking, Istanbul, Turkey, 22–26 August 2012; pp. 173–184. [CrossRef]
27. Jin, H.; Su, L.; Ding, B.; Nahrstedt, K.; Borisov, N. Enabling Privacy-Preserving Incentives for Mobile Crowd Sensing Systems. In Proceedings of the 2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS), Nara, Japan, 27–30 June 2016; pp. 344–353. [CrossRef]
28. Yan, X.; Ng, W.W.Y.; Zeng, B.; Zhao, B.; Luo, F.; Gao, Y. P2SIM: Privacy-Preserving and Source-Reliable Incentive Mechanism for Mobile Crowdsensing. *IEEE Internet Things J.* **2022**, *9*, 25424–25437. [CrossRef]
29. Feng, Z.; Zhu, Y.; Zhang, Q.; Ni, L.M.; Vasilakos, A.V. TRAC: Truthful Auction for Location-Aware Collaborative Sensing in Mobile Crowdsourcing. In Proceedings of the IEEE INFOCOM 2014—IEEE Conference on Computer Communications, Toronto, ON, Canada, 27 April–2 May 2014; pp. 1231–1239. [CrossRef]
30. Jin, H.; Su, L.; Xiao, H.; Nahrstedt, K. INCEPTION: Incentivizing Privacy-Preserving Data Aggregation for Mobile Crowd Sensing Systems. In Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing, Paderborn, Germany, 5–8 July 2016; pp. 341–350. [CrossRef]
31. Lin, J.; Yang, D.; Li, M.; Xu, J.; Xue, G. Frameworks for Privacy-Preserving Mobile Crowdsensing Incentive Mechanisms. *IEEE Trans. Mob. Comput.* **2018**, *17*, 1851–1864. [CrossRef]
32. Lin, J.; Yang, D.; Li, M.; Xu, J.; Xue, G. BidGuard: A Framework for Privacy-Preserving Crowdsensing Incentive Mechanisms. In Proceedings of the 2016 IEEE Conference on Communications and Network Security (CNS), Philadelphia, PA, USA, 17–19 October 2016; pp. 145–153. [CrossRef]
33. He, G.; Li, C.; Song, M.; Shu, Y.; Lu, C.; Luo, Y. A hierarchical federated learning incentive mechanism in UAV-assisted edge computing environment. *Ad Hoc Netw.* **2023**, *149*, 103249. [CrossRef]
34. Jin, X.; Zhang, Y. Privacy-Preserving Crowdsourced Spectrum Sensing. *IEEE/ACM Trans. Netw.* **2018**, *26*, 1236–1249. [CrossRef]
35. Sun, P.; Che, H.; Wang, Z.; Wang, Y.; Wang, T.; Wu, L.; Shao, H. Pain-FL: Personalized Privacy-Preserving Incentive for Federated Learning. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 3805–3820. [CrossRef]
36. Sun, J.; Zhang, R.; Zhang, J.; Zhang, Y. PriStream: Privacy-preserving distributed stream monitoring of thresholded PERCENTILE statistics. In Proceedings of the IEEE INFOCOM 2016—The 35th Annual IEEE International Conference on Computer Communications, San Francisco, CA, USA, 10–14 April 2016; pp. 1–9. [CrossRef]
37. Xing, K.; Wan, Z.; Hu, P.; Zhu, H.; Wang, Y.; Chen, X.; Wang, Y.; Huang, L. Mutual privacy-preserving regression modeling in participatory sensing. In Proceedings of the 2013 Proceedings IEEE INFOCOM, Turin, Italy, 14–19 April 2013; pp. 3039–3047. [CrossRef]
38. Xiao, M.; Wu, J.; Zhang, S.; Yu, J. Secret-sharing-based secure user recruitment protocol for mobile crowdsensing. In Proceedings of the IEEE INFOCOM 2017—IEEE Conference on Computer Communications, Atlanta, GA, USA, 1–4 May 2017; pp. 1–9. [CrossRef]
39. Li, Q.; Cao, G. Providing Efficient Privacy-Aware Incentives for Mobile Sensing. In Proceedings of the 2014 IEEE 34th International Conference on Distributed Computing Systems, Madrid, Spain, 30 June–3 July 2014; pp. 208–217. [CrossRef]
40. Gao, L.; Hou, F.; Huang, J. Providing long-term participation incentive in participatory sensing. In Proceedings of the 2015 IEEE Conference on Computer Communications (INFOCOM), Hong Kong, China, 26 April–1 May 2015; pp. 2803–2811. [CrossRef]
41. Zhang, X.; Xue, G.; Yu, R.; Yang, D.; Tang, J. Truthful incentive mechanisms for crowdsourcing. In Proceedings of the 2015 IEEE Conference on Computer Communications (INFOCOM), Hong Kong, China, 26 April–1 May 2015; pp. 2830–2838. [CrossRef]

42. Zhan, Y.; Li, P.; Qu, Z.; Zeng, D.; Guo, S. A Learning-Based Incentive Mechanism for Federated Learning. *IEEE Internet Things J.* **2020**, *7*, 6360–6368. [CrossRef]
43. He, X.; Wang, X.; Wang, S.; Xu, S.; Ren, J.; He, C.; Zhang, Y. A Shapley Value-Based Incentive Mechanism in Collaborative Edge Computing. In Proceedings of the 2021 IEEE Global Communications Conference (GLOBECOM), Madrid, Spain, 7–11 December 2021; pp. 1–7. [CrossRef]
44. Tran, H.G.; Ton-That, L.; Thao, N.G.M. Lagrange Multiplier-Based Optimization for Hybrid Energy Management System with Renewable Energy Sources and Electric Vehicles. *Electronics* **2023**, *12*, 4513. [CrossRef]
45. Hu, L.; Yang, Y.; Tang, Z.; He, Y.; Luo, X. FCAN-MOPSO: An Improved Fuzzy-Based Graph Clustering Algorithm for Complex Networks With Multiobjective Particle Swarm Optimization. *IEEE Trans. Fuzzy Syst.* **2023**, *31*, 3470–3484. [CrossRef]
46. Izci, D.; Ekinci, S.; Zeynelgil, H.L. Controlling an automatic voltage regulator using a novel Harris hawks and simulated annealing optimization technique. *Adv. Control Appl. Eng. Ind. Syst.* **2023**, *6*, e121. [CrossRef]
47. Tian, Y.; Li, X.; Ma, H.; Zhang, X.y.; Tan, K.C.; Jin, Y. Deep Reinforcement Learning Based Adaptive Operator Selection for Evolutionary Multi-Objective Optimization. *IEEE Trans. Emerg. Top. Comput. Intell.* **2023**, *7*, 1051–1064. [CrossRef]
48. Yao, X.; Gong, D. Genetic Algorithm-Based Test Data Generation for Multiple Paths via Individual Sharing. *Comput. Intell. Neurosci.* **2014**, *2014*, 591294. [CrossRef]

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