

Article

# Federated Learning Incentive Mechanism Setting in UAV-Assisted Space–Terrestrial Integration Networks

Chun Zhu <sup>1,†</sup> , Mengqi Sui <sup>2,†</sup>, Haitao Zhao <sup>3,\*</sup>, Keqi Chen <sup>2</sup> , Tianyu Zhang <sup>2</sup> and Chongyu Bao <sup>4</sup>

<sup>1</sup> College of Science, Nanjing University of Posts and Telecommunications, Nanjing 210003, China; zhuchun@njupt.edu.cn

<sup>2</sup> School of Communication and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China; 1022010216@njupt.edu.cn (M.S.); b21011021@njupt.edu.cn (K.C.); b21011619@njupt.edu.cn (T.Z.)

<sup>3</sup> College of Internet of Things, Nanjing University of Posts and Telecommunications, Nanjing 210003, China

<sup>4</sup> Portland Institute, Nanjing University of Posts and Telecommunications, Nanjing 210003, China; p21000108@njupt.edu.cn

\* Correspondence: zhaoh@njupt.edu.cn

† These authors contributed equally to this work.

**Abstract:** The UAV-assisted space–terrestrial integrated network provides extensive coverage and high flexibility in communication services. UAVs and ground terminals collaborate to train models and provide services. In order to protect data privacy, federated learning is widely used. However, the participation of UAVs and ground terminals is not gratuitous, and reasonable incentives for federated learning need to be set up to encourage their participation. To address the above issues, this paper proposes a federated reliable incentive mechanism based on hierarchical reinforcement learning. The mechanism allocates inter-round incentives at the upper level to ensure the maximisation of the server’s utility, and performs inter-client incentive allocation at the lower level to ensure the minimisation of each round’s latency. The reasonable incentive allocation enables the central server to achieve higher model training accuracy under the limited incentive budget, which reduces the cost of model training. At the same time, an attack detection mechanism is implemented to identify malicious clients participating in federated learning, preventing their involvement in aggregation and revoking their incentives. This better ensures the security of model training. Finally, we conducted experiments on Fmnist, and the results indicate that this method effectively improves the accuracy and security of model training.

**Keywords:** UAV; space–terrestrial integration networks; federated learning; incentive mechanism; deep reinforcement learning



**Citation:** Zhu, C.; Sui, M.; Zhao, H.; Chen, K.; Zhang, T.; Bao, C. Federated Learning Incentive Mechanism Setting in UAV-Assisted Space–Terrestrial Integration Networks. *Electronics* **2024**, *13*, 1129. <https://doi.org/10.3390/electronics13061129>

Academic Editor: Carlos Tavares Calafate

Received: 4 February 2024

Revised: 15 March 2024

Accepted: 18 March 2024

Published: 20 March 2024



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

## 1. Introduction

The space–terrestrial integrated network is based on the existing network on the ground, and expands to the space network, so as to realise a communication network with global coverage [1]. Although the basic communication facilities can basically meet the daily communication needs, when there is an unexpected situation or in unconventional temporary scenarios, relying only on ground wireless communication facilities is not enough [2]. For example, network reconstruction in major natural disasters, temporary communication deployment in remote areas, and wireless resource allocation at the scene of gathering activities during major holidays [3]. In order to effectively improve the quality of wireless communication in these scenarios, Unmanned Aerial Vehicles (UAVs) can be deployed to assist communication [4]. As small flying devices, UAVs have many advantages that can make mobile communication more convenient [5]. Therefore, UAV-assisted communication is a potential indispensable technology in mobile networks. By acting as a relay, UAVs can meet the requirements of dynamic adjustment and timely

deployment, effectively solving the transmission problem of temporary communication hotspots [6–8]. In addition, UAVs are not easily blocked by obstacles (e.g., buildings) when flying at high altitude, which is also an important advantage [9].

With the widespread use of 5G communications and the development of 6G communications, the demand for communications is increasing [10,11]. Based on their own advantages of versatility, high mobility, ease of deployment, and low cost, UAVs can be used as airspace-assisted communication platforms [12,13]. In addition, UAVs can also be used as user terminals with high mobility and be responsible for data collection in scenarios such as environmental monitoring. Processing and training large amounts of data by UAVs remains a challenge due to the absence of large storage capacity for content caching [14] and limited computational power [15]. Traditional machine learning solutions require uploading the local data from the UAVs to a central server, which may pose significant privacy and security concerns for the terminals. In addition, uploading all the data directly from UAVs to the central server requires a much larger transmission bandwidth and also consumes a lot of energy. To protect the UAVs' data privacy, we consider the use of federated learning between the UAVs and the ground terminals, which enables the UAVs to work with the ground devices to provide extensive coverage.

Federated learning, a distributed learning technique, allows UAVs and ground terminals to retain data locally and upload the trained parameters to a central server after using the local data for model training [16]. Federated learning protects the privacy of UAVs and ground terminals while reducing their data transfer costs and improving global model performance. UAVs and ground terminals need to consume their own arithmetic resources and communication overheads when participating in federated learning, and in the era of big data, data as a valuable asset have a certain inherent value [17]. Therefore, only if the central server provides a certain amount of compensation will UAVs and ground terminals be of interest.

Therefore, the incentive settings for UAVs and ground terminals in federated learning become a top priority. When designing incentives, the interests of all participants, as well as the overall performance of the system, need to be taken into account. A well-designed incentive scheme can promote the sustainable development of the federated learning system and ensure that all participants can benefit from it, so that they can better cope with the challenges of federated learning. At the same time, not all UAVs and ground terminals participating in federated learning are honest and reliable, and the presence of malicious clients can have an impact on the performance of the model. Therefore, malicious clients need to be detected and recognised when setting up incentives to ensure the security of the system.

To address the above challenges, this paper proposes a federated learning incentive mechanism with malicious defence. The mechanism identifies and rejects malicious clients by detecting clients participating in federated learning. Different weights are set for the costs of UAVs and ground terminals due to their different data transmission capabilities and computational capabilities. In order to compensate the computational and communication costs of UAVs and ground terminals, the central server determines the incentive allocation strategy through hierarchical reinforcement learning. The upper level reinforcement learning incentive coordinates the total amount of incentives and allocates the incentives for each round. Lower-level reinforcement learning subdivides the total incentive for each round to regulate the frequency of client training and minimise the federated learning latency.

The main contributions of this paper are as follows:

- The central server utilises hierarchical reinforcement learning for federated learning incentive allocation, and this incentive allocation mechanism balances the accuracy and latency of federated learning. It aims to enhance the performance of federated learning and reduce incentive expenditure.
- We set up a malicious client detection mechanism to prevent malicious clients from participating in federated learning and ensure the security of federated learning.

- The algorithm is tested using the F-MNIST dataset and the experimental results reflect the advantages of the algorithm in terms of federated learning accuracy and reliability.

The rest of the paper is organised as follows: Section 2 summarises the related work and Section 3 includes the scenario and problem description. We formulate the problem and present the objective of the incentives in Section 4, and Section 5 proposes the algorithm, i.e., Federated Learning Hierarchical Incentives. Section 6 describes the experimental setup, and the next section discusses the experimental results and effectiveness evaluation. The last section concludes the paper.

## 2. Related Work

With the popularity and development of mobile communication and IIoT technologies, various applications have increasingly high requirements for network capacity and coverage, and it is impossible to provide wireless access services with high data rates and high reliability anywhere on the earth by relying only on terrestrial communication systems, especially in environmentally hostile areas such as oceans and mountains [18]. To fill this gap and take advantage of the complementary strengths of different network segments [19], UAV-assisted space–terrestrial integrated networks have emerged.

### 2.1. Federated Learning in UAVs

The space–terrestrial integrated network (STIN) is in full swing, and in Ref. [20], caching and allocation strategies for satellite–terrestrial cooperation are investigated to reduce cache redundancy and improve cache hit rate and network throughput. The key features and challenges of adaptive interference-based approaches to prevent degradation of wireless link performance due to excessive interference in the space–terrestrial integrated network (STIN) are comprehensively investigated in Ref. [21]. However, satellite-dependent Internet users face a wide range of Internet access disruptions, and UAVs can be used to temporarily provide alternative links to ensure continuous underlying connectivity in the event of link disruptions between satellites and terrestrial components [22]. At the same time, UAVs and ground terminals complement each other to provide ubiquitous connectivity to a variety of underserved areas [23].

UAVs and ground terminals collaborate with each other through federated learning, which protects the data privacy of both and also improves the training performance of the model. Reinforcement learning and federated learning are combined in Ref. [24] to greatly improve the positioning accuracy of ground users by relying on RSS technology with the UAV as the base station. A UAV-assisted hierarchical federated learning is proposed in Ref. [25] to minimise the time required for federated learning to achieve the target learning accuracy by finding the optimal UAV location, user UAV association, channel assignment, and user selection. A synchronous federation learning (SFL) architecture for multiple UAVs is constructed in Ref. [26] and a comparative analysis of asynchronous federation learning (AFL) and synchronous federation learning (SFL) is performed.

### 2.2. Incentives for Federated Learning

In order to motivate UAVs and ground terminals to participate in federated learning while minimising the training expenditure for model training, the incentives need to be set fairly and reasonably. A new federated learning crowdsourcing framework is presented in Ref. [27]. The framework demonstrates incentive-based interactions between a crowdsourcing platform and independent strategies of participating clients to train a global learning model in which each party maximises its own benefits. The authors in Ref. [28] derive a Nash equilibrium that allows the parameter server to accurately assess the client's contribution to the training accuracy. In Ref. [29], a reputation-based reliable federated learning worker selection scheme is designed, which combines reputation with contract theory to effectively improve the accuracy of federated learning. Shapley values are used in Ref. [30] to calculate the importance of grouped features for fair credit allocation. A blockchain-based value-driven incentive mechanism is proposed in Ref. [31] to force

participants to behave correctly and provide auditability for the entire training process. In Ref. [32], a random auction framework is used for incentive mechanism design, where the base station receives submitted bids and requires an algorithm to select the winning bidder and determine the corresponding reward to minimise the social cost.

However, most studies on federated learning incentives have focused on improving the accuracy of federated learning without considering the problem of limited incentive costs. In addition to this, few incentive mechanisms simultaneously take into account the security and defence functions. The federated learning incentive mechanism proposed in this paper, which includes malicious defense, integrates security mechanisms and incentive mechanisms for federated learning. It enhances the security of federated learning without introducing additional computational overhead. The goal of the central server when performing model training is to obtain highly accurate models. For the client, the more incentives the central server provides, the faster the client's model training is within the limit of the client's training frequency can reach. In the UAV-assisted space-terrestrial integrated network, model training has a timeliness requirement, so we should minimise the time delay during model training. With limited incentive cost, we allocate incentives through hierarchical reinforcement learning, which improves the model training accuracy and reduces the model training delay.

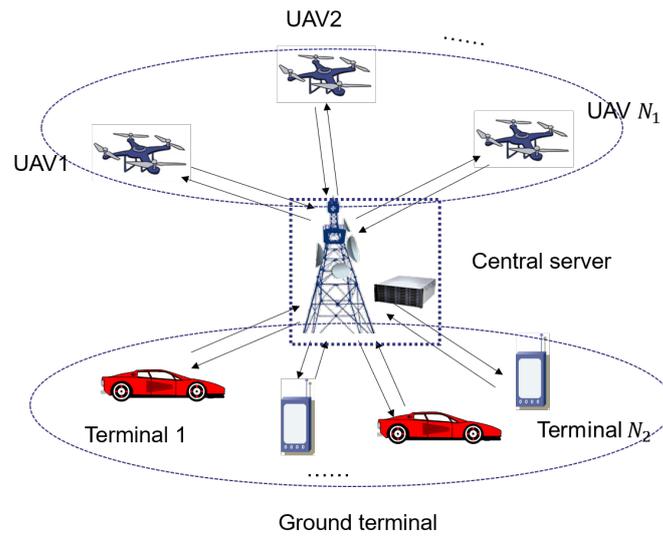
### 3. System Model

In this section, we describe the UAV-assisted space-terrestrial integrated network scenario and the application of federated learning in that scenario.

#### 3.1. Scenario Description

An UAV-assisted space-terrestrial integrated networks refers to the use of UAV technology to work in concert with ground terminal equipments to form an integrated and seamless communication and data processing network. Federated learning is a distributed machine learning technique that has a wide range of applications in UAV-assisted space-terrestrial integrated networks, for example, in the fields of agricultural monitoring, environmental protection, traffic monitoring, disaster commentaries, network connectivity, and so on. These application scenarios fully demonstrate the advantages of UAV-assisted space-terrestrial integrated networks in terms of improving efficiency, reducing costs, and enhancing security. With the continuous development of technology, more innovative applications are expected to emerge.

UAVs as well as ground terminals act as clients for federated learning in UAV-assisted space-terrestrial integrated networks. UAVs are used to collect data from high altitude platforms due to their wide coverage and ease of deployment. Ground terminals, such as vehicle-mounted sensors, and smart devices, such as mobile phones, are used to collect detailed data on the ground, and the two complement each other. With federated learning, UAVs and ground terminals do not need to share local data information, but only need to upload model parameters to the parameter server located at the base station, which protects data privacy and security. Assuming that there are  $N_1$  UAVs participating in federated learning and their local dataset is  $\{X_1, X_2, \dots, X_{N_1}\}$ , and there are  $N_2$  ground terminals participating in federated learning and their local dataset is  $\{X'_1, X'_2, \dots, X'_{N_2}\}$ . The UAV-assisted space-terrestrial integration network is shown in Figure 1, where the UAVs are located in the high altitude platform and can be used for data collection, data processing, etc. The UAVs serve as an important complement to the ground terminals, expanding their service coverage in remote areas such as rural areas and in various emergency situations. Wireless communication infrastructure such as edge servers or cloud servers located at the base station act as a central server for uniting the UAVs with the ground terminals.



**Figure 1.** Schematic diagram of a UAV-assisted space–terrestrial integration network.

### 3.2. A Federated Learning Model Based on UAVs and Ground Terminals

In federated learning, a central server located at the base station or cloud server is responsible for sending learning tasks, aggregation of parameters, and incentive issuance. Pedestrian users, self-driving cars, and other ground terminals, together with UAVs, act as clients and are responsible for training with their own data and uploading the training parameters to the central server for aggregation. The specific federated learning process is shown below.

The central server receives the model training task and distributes the initialised model parameters to the UAVs and ground terminals. The UAVs and ground terminals train the local models using the local datasets  $\{X_1, X_2, \dots, X_{N_1}\}$  and  $\{X'_1, X'_2, \dots, X'_{N_2}\}$  and upload the trained local model parameters to the server for aggregation to generate new global model parameters. In this process, a loss function needs to be defined, which is minimised for learning purposes by using the clients' datasets. A total of  $N_1 + N_2$  clients of UAVs and ground terminals participate in the federated training.  $f_i(\omega)$  denotes the loss function, where  $0 \leq i \leq N_1 + N_2 - 1$ , the dataset owned by client  $i$  is  $D_i$ , and the number of this dataset is denoted by  $|D_i|$ . Then, the loss function of this client is:

$$F_i(\omega) = \frac{1}{|D_i|} \sum_{j \in D_i} f_j(\omega) \tag{1}$$

The global loss function  $F(\omega)$  can be expressed as:

$$F(\omega) = \frac{\sum_{i=1}^{N_1+N_2} |D_i| F_i(\omega)}{\sum_{i=1}^{N_1+N_2} |D_i|} \tag{2}$$

The optimal  $\omega$  is obtained through the global loss function, and we use the federated averaging algorithm to approximate the optimal  $\omega$ . The central server aggregates the average gradient of the local model and uses it for updating:

$$\omega_{i,k+1} \leftarrow \omega_{i,k} - \eta \Delta f(\omega_{i,k}) \tag{3}$$

where  $\omega_{i,k}$  denotes the training parameters of the  $k$ -th iteration of the  $i$ -th server,  $\eta$  denotes the learning rate,  $\Delta f(\omega_{i,k})$  is the gradient of the loss function, and finally, the cloud server performs a global aggregation of the client's training parameters.

#### 4. Setting of Federated Learning Incentives

In order to design a reasonable incentive algorithm, we first need to specify the cost required for the client to participate in federated learning, so as to obtain the utility of the client's participation in federated learning. The optimisation goal of both the client and the server is to maximise their own utility.

##### 4.1. Cost of UAVs and Ground Terminals

When participating in federated learning, the client needs to consume its own arithmetic and communication resources. Therefore, to design the incentive mechanism for federated learning, we need to specify the cost of UAVs and ground terminals when they participate in federated learning. We assume that the cost for UAVs to participate in federated learning is  $E_{i,k}^{UAV}$ , and the cost for ground terminals is  $E_{i,k}^{GT}$ . The cost of the clients involved in federated learning, whether UAVs or ground terminals, mainly consists of two aspects: computation cost and communication cost. The computational cost of clients participating in federated learning is mainly related to their computational power, and the CPU computational cost  $E_{i,k}^{cmp}(f_i)$  for the  $k$ th local update of client  $i$  is expressed similar to Ref. [16]. We propose the Hierarchical deep learning (HRL) algorithm to solve the p2 optimisation problem. We describe the HRL algorithm in three parts: client contribution measurement, malicious client detection mechanism, and practical deployment of the hierarchical incentive algorithm.

$$E_{i,k}^{cmp}(f_i) = \zeta_i c_i D_i f_{i,k}^2 \quad (4)$$

where  $\zeta_i$  denotes the effective capacitance factor of the computational chipset of client  $i$ ,  $c_i$  denotes the number of CPU cycles used by client  $i$  to perform the computation of one bit of data samples, and the total number of data that client  $i$  participates in training is  $D_i$ .

After training with local data, clients need to upload the trained model parameters to the central server for aggregation. The communication cost required for uploading the model parameters is related to the time, power, and number of parameters required for uploading the parameters, and the communication cost  $E_{i,k}^{com}$  can be expressed as:

$$E_{i,k}^{com} = p_i T_{i,k}^{com} = p_i \frac{b_i}{R_i} \quad (5)$$

where  $p_i$  denotes the transmission power of client  $i$ ,  $b_i$  is the magnitude of the number of uploaded model parameters, this magnitude being the same for each client, and  $R_i^j$  is the uplink data rate achievable by client  $i$  in subchannel  $j$ , which can be expressed as the data transmission rate of the  $j$ -th subchannel of client  $i$  according to Shannon's formula:

$$R_i^j = W \log \left( 1 + \frac{(A_i - 1) p_i h_i^j}{W N_0} \right) \quad (6)$$

where the bandwidth of the subchannel is  $W$ ,  $p_i$  is the transmission power of client  $i$ ,  $h_i^j$  is the  $j$ -th subchannel power gain between the client and the server,  $N_0$  is the power spectral density of the additive Gaussian white noise during wireless transmission, and  $A_i$  is the number of antennas assigned to client  $i$ . Therefore, the total uplink rate when client  $i$  participates in federated learning is the sum of all subchannel data transmission rates:

$$R_i = \sum_{j=0}^{J_i} R_i^j \quad (7)$$

where  $J_i$  is the total number of subchannels used by client  $i$  for model updating.

The training cost of client  $i$  is the sum of the computational and communication costs:

$$E_{i,k} = \alpha E_{i,k}^{cmp}(f_i) + \beta E_{i,k}^{com} \tag{8}$$

where  $\alpha$  and  $\beta$  are the adjustment coefficients of computational cost and communication cost. The UAV has good communication performance but limited arithmetic power, and the ground terminal has high communication cost but stronger arithmetic power, so in the setting of parameters,  $\alpha_{UAV} > \alpha_{GT}$ ,  $\beta_{UAV} < \beta_{GT}$ .

#### 4.2. Setting of Incentives

Clients participate in federated learning by consuming their own resource costs, such as arithmetic, energy, time, and traffic. Clients will only participate in federated learning if the central server provides a certain incentive to do so. Based on the incentives provided by the central server, clients determine the optimal computation frequency for training their local neural network models, so as to maximise the benefits of their participation in federation learning. The client maximises its own benefit within its own achievable frequency range, and the optimisation problem can be expressed as:

$$P1 : \max_{f_{i,k}} income_{i,k}^c \tag{9}$$

$$s.t. \begin{cases} income_{i,k} \geq 0 \\ f_{i,k} \in [f_{i,k}^{\min}, f_{i,k}^{\max}] \end{cases}$$

The utility  $income_{i,k}^c$  of client  $i$  when participating in the  $k$ -th training round of federated learning is the difference between the incentives gained by the client from participating in federated learning and the cost consumed by participating in learning, which can be expressed as:

$$income_{i,k}^c = pay_{i,k} \zeta_{i,k} - E_{i,k} \tag{10}$$

where  $pay_{i,k}$  denotes the incentive payoff per CPU cycle frequency given to client  $i$  by the server in the  $k$ -th round of training; the form of incentive payoff is decided by the server in the actual deployment, which can be money, information resources, access to models and other forms of resources. Moreover,  $\zeta_{i,k}$  denotes the frequency of CPU cycles used by client  $i$  when it participates in local local model training in  $k$  rounds of training.

The client, in order to determine its own optimal training frequency, calculates the first order derivative of  $income_{i,k}^c$  with respect to  $\zeta_{i,k}$  as:

$$\frac{\partial income_{i,k}^c}{\partial \zeta_{i,k}} = pay_{i,k} - 2\zeta_{i,k} c_i D_i f_i \tag{11}$$

After the second order derivation of  $income_{i,k}^c$  on  $\zeta_{i,k}$ , it can be seen that  $income_{i,k}^c$  is a convex function of  $\zeta_{i,k}$ , so for a given  $income_{i,k}^c$ , the local can be calculated when  $\frac{\partial income_{i,k}^c}{\partial \zeta_{i,k}} = 0$ . The client's unique optimal training frequency value is:

$$\zeta_{i,k}^{Best} = \frac{pay_{i,k}}{2\zeta_{i,k} c_i D_i} \tag{12}$$

The client, in order to maximise its revenue, defaults to the best local training frequency  $\zeta_{i,k}^{Best}$  after receiving the offer from the server.

The goal of the central server is to achieve higher training accuracy using as little incentive as possible. Thus, for the central server, the goal of incentive design is to achieve higher

training accuracy within a limited budget through a reasonable incentive allocation method. Therefore, the goal of the central server is defined as the optimisation problem of P2:

$$\begin{aligned}
 P2 : \max_{pay_{i,k}} \text{income}^s &= \lambda A(\omega) - \sum_{k=0}^{K-1} \sum_{i=0}^{N-1} pay_{i,k} f_{i,k} \\
 s.t. \sum_{k=0}^{K-1} \sum_{i=0}^{N-1} pay_{i,k} f_{i,k} &\leq B_{sever}
 \end{aligned} \tag{13}$$

The central server requires reasonable incentive settings in order to achieve the P2 optimisation problem, and the central server incentive setting algorithm is described in Section 5 below.

### 5. Hierarchical Incentives for Federated Learning

We propose the HRL algorithm to solve the p2 optimisation problem. We describe the HRL algorithm in three parts: client contribution measurement, malicious client detection mechanism, and practical deployment of the hierarchical incentive algorithm.

#### 5.1. Client Contribution Measurement

Since the individual clients are cooperating with each other in the game, certain benefits are generated through cooperation. The Shapley value is used to determine the contribution made by each participant in the cooperation and assign them a reasonable share of the benefits. The core idea of the Shapley value is that the contribution of a participant should be measured by their marginal contribution in all possible coalitions. Marginal contribution is the change in value to the whole coalition when a participant joins or leaves a coalition. The Shapley value provides a better measure of the client’s contribution in multidimensional and complex situations. The Shapley value measures the value of the client’s contribution as:

$$S_i^k(F) = \sum_{S \subseteq F, i \notin S} \frac{|S|!(|F| - |S| - 1)!}{|F|!} \{v(S \cup i) - v(S)\} \tag{14}$$

We denote the data federation as  $F = \langle Users, v \rangle$ , where *Users* denotes all clients, *v* is the model test accuracy, and *S* is the subset that does not contain client *i*. Using  $\frac{|S|!(|F| - |S| - 1)!}{|F|!}$  denotes the probability of client *i* appearing in each cooperation order, and  $\{v(S \cup i) - v(S)\}$  denotes the marginal contribution of client *i*, i.e., the difference between the model test accuracy when the subset containing client *i* participates in training and the model test accuracy when the subset not containing client *i* participates in training.

The central server determines client incentives based on the client’s contribution value  $S_i^k(F)$ . The larger the contribution value  $S_i^k(F)$  of client *i* is, the more the client contributes to the global training, and we provide more incentives for such a client.

#### 5.2. Malicious Client Detection Mechanism

The Shapley value is used to measure the contribution of the client and there are differences in the calculated Shapley value for different clients. The more positive the contribution of the client to the system model, the larger the Shapley value. Thus, we can use the Shapley value for noisy data detection and toxic data detection to improve the robustness of the system. Based on the client’s contribution value, we can identify the malicious clients in the system, provide 0 incentive to the malicious clients, and prevent the malicious clients from participating in the aggregation. Meanwhile, the malicious records of the clients are saved in the central server, and the clients with too many malicious records will no longer be selected in the subsequent tasks. We set the coefficient  $\theta(S_i^k(F), \gamma)$  to distinguish honest clients from malicious clients. In the case where a client has malicious behaviours such as flip-tag attack and adversely affects the global model training accu-

racy, the Shapley value of that client is less than zero, so we generally set  $\gamma = 0$  in our experiments.

$$\theta(S_i^k(F), \gamma) = \begin{cases} 1, S_i^k(F) > \gamma \\ 0, S_i^k(F) < \gamma \end{cases} \quad (15)$$

Subject to the malicious client detection parameter  $\theta(S_i^k(F), \gamma)$ , the incentive payoff returned by the server for client  $i$ 's participation in the model training at round  $k$  can be expressed as:

$$pay_{i,k} = \alpha_{i,k} \theta(S_i^k(F), \gamma) S_i^k(F) \quad (16)$$

where  $\alpha_{i,k}$  is a tuning parameter, and the central server dynamically adjusts the value of  $\alpha_{i,k}$  by using a proximal policy optimisation (PPO) algorithm. Moreover,  $\alpha_{i,k}$  controls the incentive to client  $i$  to adjust the training frequency of client  $i$ , thus equalising the learning time of different clients to minimise the training time for each round and enhance the time utilisation.

### 5.3. Hierarchical Incentive Setting

UAVs and ground terminals cotrain machine learning models through federated learning, which are often complex, and their accuracy gains are difficult to obtain through simple mathematical reasoning. At the same time, due to the privacy protection of the clients by federated learning, it is difficult to obtain information about its private information and training capability. In the incentive setting, we have to satisfy both the need to reduce the incentive cost and the need to minimise the latency. Satisfying these two needs at the same time, is not possible with only a single layer of constraints. In order to ensure better global model training performance, this paper adopts a deep reinforcement learning approach based on PPO to design an optimal hierarchical incentive strategy. This strategy not only saves the incentive cost, but also ensures the security of the training process and reduces the training latency to a certain extent. The PPO algorithm allows servers and clients to dynamically adjust their training strategies to optimise their benefits without prior knowledge of each other's behavioural norms.

The steps for the implementation of this hierarchical incentive mechanism are shown below: the upper state of the  $k$ -th round consists of two main aspects: the historical information of the previous  $k - 1$  rounds (including the server's historical pricing policy, the client's training time, etc.) and the current state (the remaining budget).

Upper-level actions are used to allocate the total incentive for each round. From the upper-layer actions, the lower-layer states, i.e., the total incentives for each round, can be obtained. Since the iteration time of each round of federation learning depends on the time  $\max_{i \in N} T_i^k$  used by the client with the largest training time, the lower layer action is used to distribute the total incentive of the round among the clients to minimise the iteration time of each round of federated learning. Here,  $T_i^k$  denotes the local model training time of client  $i$  when it participates in the  $k$ -th round of federated learning.

The upper level reward  $R_{up}$  is to maximise the benefit to the server, which is expressed as the difference between the accuracy gain  $\lambda A(\omega_k)$  and the incentive paid to the client:

$$\max R_{up} = \lambda A(\omega_k) - \sum_{i \in F} \alpha_{i,k} \theta(S_i^k(F), \gamma) S_i^k(F) \zeta_{i,k} \quad (17)$$

where  $\lambda$  denotes a tuning parameter for the accuracy gain, which is used to adjust the accuracy gain to the same order of magnitude as the incentive payoff that the server pays to the clients, and  $A(\omega_k)$  is the test accuracy of the global model after the  $k$ -th round of training or is some other model performance metric.

The lower incentive mechanism is used to minimise the training time of each round of federated learning; the server can adjust the size of  $pay_{i,k}$  by changing the value of  $\alpha_{i,k}$  so as to control the training time of the client's local model. Therefore, the lower layer reward is:

$$\max R_{low} = - \max_{i \in N} T_i^k \tag{18}$$

where  $T_i^k$  denotes the local model training time of client  $i$  when it participates in the  $k$ -th round of federated learning.

The local model training time  $T_i^k$  of client  $i$  while participating in the  $k$ -th round of federated learning consists of two parts: communication elapsed time  $T_{i,K}^{com}$  and computation elapsed time  $T_{i,cmp}^k$ :

$$\begin{aligned} T_i^k &= T_{i,k}^{cmp} + T_{i,k}^{com} \\ &= \frac{c_i D_i}{f_{i,k}} + \frac{b_i}{\sum_{j=0}^{J_i} W \log \left( 1 + \frac{(A_i-1)p_i h_i^j}{WN_0} \right)} \end{aligned} \tag{19}$$

where  $c_i$  denotes the number of CPU cycles for client  $i$  to execute one bit of training data and  $b_i$  is the size of the model parameters uploaded by client  $i$ . During the iteration process, the upper and lower layers gradually explore the optimal policies for their respective goals to maximise the reward  $R_{up}$  and  $R_{low}$  of the upper and lower layers. With each iteration of federated learning, the incentive strategy of the server is dynamically adjusted accordingly to approach the optimal gradually, and the iteration is stopped when the server’s budget is exhausted or the federated learning model reaches convergence. The implementation of the algorithm is shown in Algorithm 1, and the flowchart of the algorithm is shown in Figure 2.

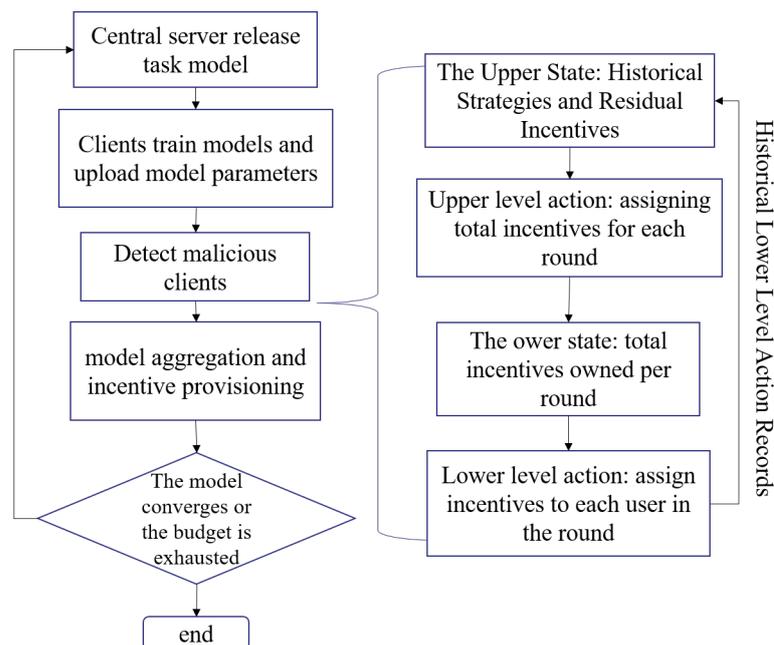


Figure 2. Schematic of hierarchical incentives for federated learning.

**Algorithm 1:** Hierarchical incentive algorithms for federated learning.

---

**Require:** Precision correction factor  $\lambda$ , number of clients  $C$ , higher-level policy  $\pi_{\text{H}}(a_k^{up}, s_k^{up})$ , lower-level policy  $\pi_{\text{L}}(a_k^{low}, s_k^{low})$ , learning rate  $\eta$ ;

- 1: Initialisation  $\omega^0$ ;
- 2: **if**  $k = 0$  **then**
- 3:   Broadcast initialisation  $\omega^0$  to all clients.
- 4:   Broadcast initialisation lower-level policy  $\pi_{\text{L}}$  pricing policy.
- 5:   **for**  $i \in S$  **do**
- 6:     Selection of optimal training frequency values based on pricing strategy.
- 7:     Return gradient  $\nabla g_0^{(1)}$  training time  $T_0^k$  to central server.
- 8:   **end for**
- 9:   **Central server side:**
- 10:   Average the gradient and update  $\omega_{i,1} \leftarrow \omega_{i,0} - \eta \Delta f(\omega_{i,0})$ .
- 11:   Measuring client contributions for attack identification and contribution allocation.
- 12:   Assign the upper round incentive  $pay_{i,0}$  and update the upper state  $s_1^{up}$
- 13: **end if**
- 14: **if**  $k = 1, 2, \dots, K - 1$  **then**
- 15:   The upper level strategy  $a_k^{up}$  is determined by  $\max target_{up}$  based on the upper level state  $s_k^{up}$ .
- 16:    $a_k^{up} \rightarrow s_k^{low}$
- 17:   **for**  $i \in S$  **do**
- 18:     Broadcast the global model  $\omega_k$  and pricing policy  $pay_{i,k-1}$  to all clients.
- 19:     Client selects the most frequent training frequency to train.
- 20:     Return gradient  $\nabla g_k^{(t)}$  training time  $T_i^k$  to central server.
- 21:   **end for**
- 22:   **Central server side:**
- 23:   **if** Target accuracy not met or budget not exhausted **then**
- 24:     Average gradient, update  $\omega_{i,k+1} \leftarrow \omega_{i,k} - \eta \Delta f(\omega_{i,k})$  and update the upper state  $s_{(k+1)}^{up}$ .
- 25:     Measuring client contributions for attack identification and contribution allocation.
- 26:     Adjust  $\alpha_{i,k}$  by determining the lower level strategy  $a_k^{low}$  from  $\max target_{low}$  based on the lower level state  $s_k^{low}$ .
- 27:     Make single round incentive allocations  $Pay_{i,k} = \alpha_{i,k} \theta(S_1^k(F) S_i^k(F))$
- 28:   **end if**
- 29: **end if**

---

## 6. Experimental Results

In this section, we design experiments to compare the performance of the proposed HRL algorithm with other algorithms and analyse the experimental results.

### 6.1. Experimental Setup

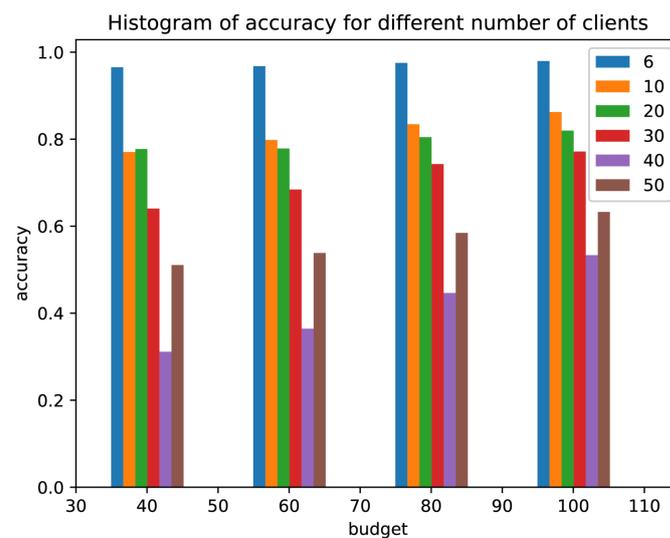
In recent years, UAVs working in conjunction with ground terminals have a wide range of applications in environmental monitoring, disaster response, military fields, etc. CNN and RNN models provide more advanced and sophisticated combinatorial mechanisms to learn representations from machine data. Therefore, we choose the CNN model for our simulation experiments. We use PyTorch to implement HRL and run the experiments with a GeForce MX 450.

Fmnist is a lightweight image classification dataset containing grey scale images of various garments. Compared with the traditional MNIST dataset, Fmnist is closer to the real scene and has better generalisation ability. The Fmnist dataset contains 10 categories of

clothing images covering a wide range of different objects and textures, which can better reflect the diversity of data in the real scene. In UAV-assisted space–terrestrial integrated networks, image classification tasks may be widely used, such as identifying features and detecting targets. Therefore, using the Fmnist dataset can simulate such image classification tasks and verify the effectiveness of federated learning in this scenario.

The Fmnist dataset uses a CNN model for the classification task. The model consists of two convolutional layers. The first convolutional layer has 10 output channels and a convolutional kernel size of 5. The second convolutional layer has 10 input channels, 20 output channels, and a convolutional kernel size of 5. The first fully connected layer has an input size of 320 and an output size of 50. The second fully connected layer has an input size of 50 and an output size of 10, because Fmnist has 10 numerical categories. A Dropout layer is also set up for random neuron dropping during training to prevent overfitting. The size of the input layer is determined by the feature dimension of the data. The image size in the Fashion-MNIST dataset is  $28 \times 28$ , and it is a grey scale image, so the dimension is 1. The number of input layers is determined by the number of output channels, i.e., 10. In the forward propagation method, the features undergo convolution, pooling, spreading, and other operations for final classification output.

We keep the total amount of data constant at 60,000 and distribute the data unevenly to 6, 10, 20, 30, 40, and 50 clients using the HRL algorithm for incentive allocation, and the experimental results are shown in Figure 3. We can see from the figure that the training effect is best when the number of clients is set to 6, so we set the number of clients to 6 in the subsequent experiments.



**Figure 3.** Histogram of accuracy with a different number of clients.

The experiment uses a federated learning framework which consists of a server and six clients. Three of the clients are set up as UAVs and three are set up as ground terminals. The UAVs have better communication capabilities, but weaker arithmetic power compared to the ground terminals. Therefore, the number of CPU cycles  $c_i$  used by the UAV to perform the calculation of one data sample is set to 15 cycles/bit, the ground terminal is set to  $c_i = 30$  cycles/bit, the communication time of the UAV is randomly distributed in the range of 10~15 s, and the communication time of the ground terminal is randomly distributed in the range of 15~20 s. We set  $\alpha_{UAV} = 2, \alpha_{GT} = 1, \beta_{UAV} = 1, \beta_{GT} = 2$ , and the effective capacitance factor is  $2 \times 10^{-28}$ . In practice, the amount of data owned by each UAV and ground terminal is also different, so six clients are set up in the experiments that are not independently and identically distributed and have datasets of different sizes. The Fmnist dataset is used for the experiments. We set up different clients with different data distributions, and the model needs to have the ability to handle data heterogeneity in order

to adapt to the data characteristics in different environments. At the same time, we set different clients with different amount of data in our experiments. This can help evaluate the generalisation ability of the model in the face of sparse data or uneven data volume. By performing federated learning on data from different clients, the model's ability to adapt in different environments can be evaluated. This is crucial for the deployment of the model in real scenarios, as data distributions in the real world are often non-independently and identically distributed.

We abbreviate the hierarchical reinforcement learning motivation algorithm proposed in this paper as HRL. We compare the proposed algorithm with the greedy algorithm, the baseline algorithm, and the Monte Carlo (MC) algorithm, and the remaining three algorithms are set up as shown below.

**Greedy Algorithm [33]:** the greedy algorithm selects an initial solution from all possible solutions to the problem, the algorithm makes locally optimal choices at each stage and expects to reach the global optimum through these locally optimal choices.

**Baseline Algorithm [28]:** a single-level DRL algorithm is designed to constrain both reward and time, i.e., to set the single-level reward to  $R = -(\alpha T_{globe} + \sum_{i \in F} S_i^k(F) \zeta_{i,k})$ , while the PPO algorithm is used to dynamically adjust the incentives between rounds.

**Monte Carlo algorithm [34]:** alternating iterations of Monte Carlo sampling and proximal policy optimisation. In each iteration, Monte Carlo sampling is performed to obtain new samples and then the proximal policy optimisation algorithm is used to adjust the parameters. Such an iterative process can help the algorithm gradually converge to the global optimal solution or the local optimal solution.

## 6.2. Experimental Results and Analyses

Under different incentive budgets of 40, 80, 120, 160, and 200, the experiments compare the number of training rounds, the time consumed per round, and the training accuracy and loss function of the Fmnist dataset under four incentive allocation mechanisms: Greedy, Baseline, MC, and HRL. Since the number of datasets owned by the UAV good ground terminal is not the same, we set the ratio of the number of datasets owned by each user as 5:6:7:4:8:3. The experimental results are described below.

As can be seen from the experimental results of the accuracy under different budgets for the Fmnist dataset in Figure 4, HRL is able to achieve the highest training accuracy under the same incentive budget. For example, at an incentive budget of 40, the trained model using the HRL incentive allocation mechanism already achieves 96.3% accuracy, whereas the model using the baseline and MC incentive allocation mechanisms only achieves 96% accuracy at an incentive budget of 80, and the greedy algorithm only achieves 95.7% accuracy at an incentive of 100.

Similarly, as shown in Figure 5, the loss function value of the model under the HRL algorithm is always lower than that of the baseline, MC, and greedy algorithms under the same incentive budget. From Figure 6, it can be seen that the average training time per round is highest for the greedy algorithm and similar for the baseline and MC algorithms under different training budgets. The average training time for the HRL algorithm is slightly higher than baseline and MC algorithms at budgets of 20, 40, 60, and 80, and is lowest among the four incentive algorithms at an incentive budget of 100.

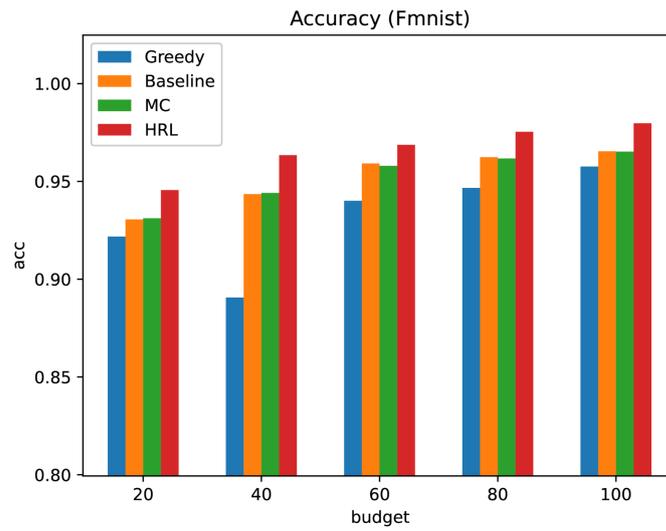


Figure 4. Test accuracy for training.

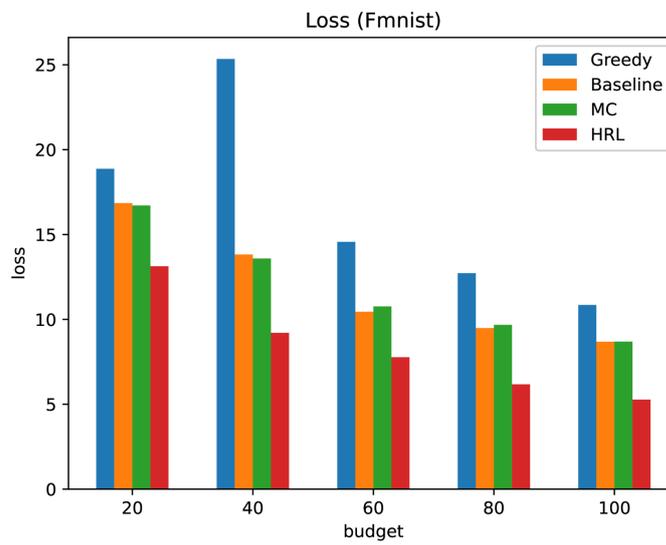


Figure 5. Test loss for training.

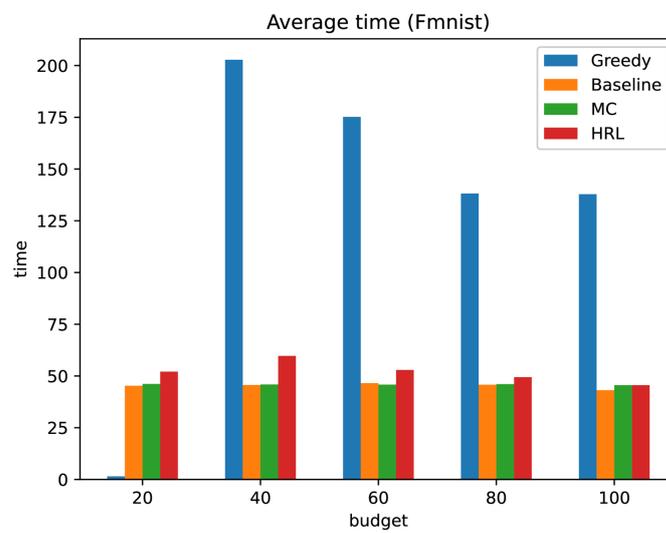
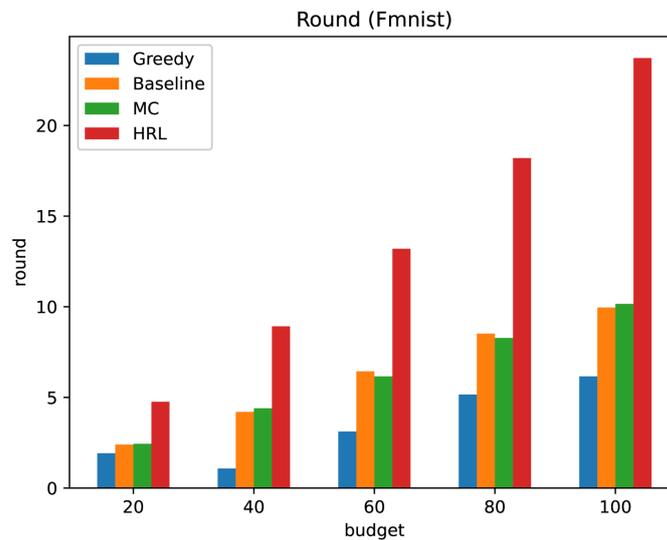


Figure 6. Average time per round.

As shown in Figure 7, by analysing the number of training rounds for various incentive allocation mechanisms under the same incentive budget, we can see that the HRL algorithm always has more training rounds compared to the other three compared algorithms under the same budget. The more the number of training rounds, the higher the accuracy of the trained model when the model has not reached full convergence. Thus, by using the HRL incentive allocation mechanism, higher training accuracy and smaller training loss can be achieved at the cost of a small increase in training time.

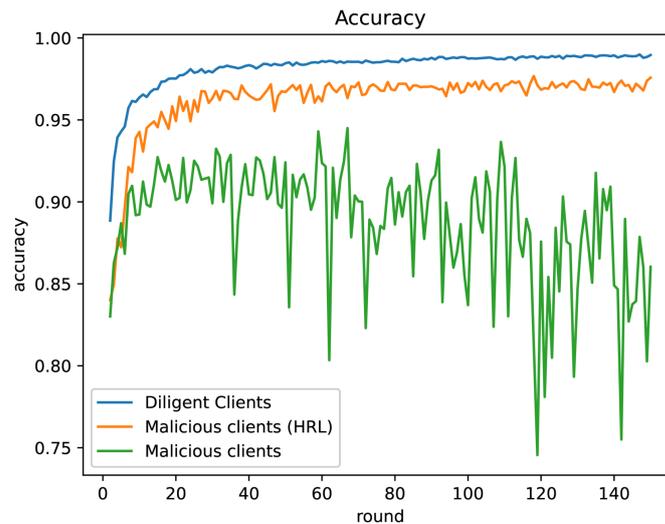


**Figure 7.** Number of training rounds.

Since federated training at UAVs and ground terminals has more flexible scheduling of task release and data collection, the requirement for latency can be reduced by reasonable planning and scheduling. However, model training in space–terrestrial integrated networks mostly requires higher accuracy to ensure the reliability of model training. Therefore, the HRL model is suitable for UAV-assisted air-ground integrated networks because the algorithm saves training costs, ensures federated learning security, and improves the economic efficiency of model training. The HRL incentive mechanism, besides being applied to UAV-assisted space–terrestrial integrated networks, has a better prospect in many fields, such as automotive networking [35], smart factories, and schools.

In order to validate the HRL model’s ability to detect malicious users, we partially inject the flip-flop label attack on the clients participating in federated learning. We still design a central server and six clients to participate in federated learning, three of which are UAV clients and three of which are ground terminals. We inject 50% of the flip label attack to one of the UAV clients and two of the ground terminals, with the remaining clients left as normal. Figure 8 shows three curves for the Fmnist dataset, under which all six clients are diligent clients (blue curve), containing three malicious clients but using the HRL algorithm (red curve), and containing three malicious clients but no malicious client detection algorithm (green curve).

As can be seen in Figure 8, the accuracy of the HRL algorithm is slightly reduced in the presence of a flip-flop tag attack. However, compared to the absence of defence mechanism, HRL not only improves the model accuracy, but also reduces the oscillations and increases the stability of the model.



**Figure 8.** Comparison of accuracy under the Fmnist dataset containing malicious clients.

## 7. Conclusions

In this paper, we present a federated learning-based model of hierarchical reinforcement learning incentives to protect the data privacy of UAVs and ground terminals. The model aims to incentivise participants to actively participate in joint training. This incentive mechanism is capable of dynamically adjusting the rewards to motivate the parties to collaborate according to the stated goals while maximising the protection of their local data. By conducting experiments on the Fmnist dataset, we observe that using our proposed hierarchical reinforcement learning incentive mechanism slightly increases the training time. However, for the same budget, our approach achieves higher training accuracy. This shows that our model is able to improve the model performance under a certain incentive budget. Meanwhile, we set up a malicious attack identification mechanism to better secure the model training. This mechanism detects and filters out potential malicious attacks to ensure that the data and models of the participants are not compromised. By comparing our model with greedy, baseline, and Monte Carlo algorithms, we demonstrate its advantages in saving incentive budget, improving training accuracy and guaranteeing model security.

**Author Contributions:** Conceptualization, H.Z.; Methodology, C.Z.; Software, C.Z.; Validation, C.Z.; Formal analysis, C.Z.; Investigation, M.S.; Resources, C.Z.; Data curation, K.C.; Writing—original draft, C.Z.; Writing—review & editing, H.Z.; Visualization, C.Z.; Supervision, H.Z.; Project administration, T.Z. and C.B.; Funding acquisition, H.Z. 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 under Grant 62371250, Science and Technology Innovation 2030—Major Project under Grant 2021ZD0140405, the Natural Science Foundation on Frontier Leading Technology Basic Research Project of Jiangsu under Grant BK20212001, and the Jiangsu Natural Science Foundation for Distinguished Young Scholars under Grant BK20220054.

**Data Availability Statement:** Derived data supporting the findings of this study are available from the corresponding author on request.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Feng, B.; Zhou, H.; Zhang, H.; Li, G.; Li, H.; Yu, S.; Chao, H.C. HetNet: A Flexible Architecture for Heterogeneous Satellite-Terrestrial Networks. *IEEE Netw.* **2017**, *31*, 86–92. [\[CrossRef\]](#)
2. Liu, M.; Yang, J.; Gui, G. DSF-NOMA: UAV-Assisted Emergency Communication Technology in a Heterogeneous Internet of Things. *IEEE Internet Things J.* **2019**, *6*, 5508–5519. [\[CrossRef\]](#)
3. Feng, W.; Tang, J.; Zhao, N.; Fu, Y.; Zhang, X.; Cumanan, K.; Wong, K.K. NOMA-based UAV-aided networks for emergency communications. *China Commun.* **2020**, *17*, 54–66. [\[CrossRef\]](#)

4. Zhao, H.; Liu, K.; Liu, M.; Garg, S.; Alrashoud, M. Intelligent Beamforming for UAV Assisted IIoT Based on Hypergraph Inspired Explainable Deep Learning. *IEEE Trans. Consum. Electron.* **2023**. [[CrossRef](#)]
5. Chen, Y.; Liu, X.; Zhao, N.; Ding, Z. Using Multiple UAVs as Relays for Reliable Communications. In Proceedings of the 2018 IEEE 87th Vehicular Technology Conference (VTC Spring), Porto, Portugal, 3–6 June 2018; pp. 1–5. [[CrossRef](#)]
6. Jiang, H.; Xiong, B.; Zhang, H.; Basar, E. Physics-Based 3D End-to-End Modeling for Double-RIS Assisted Non-Stationary UAV-to-Ground Communication Channels. *IEEE Trans. Commun.* **2023**, *71*, 4247–4261. [[CrossRef](#)]
7. Hanyu, A.; Kawamoto, Y.; Kato, N. On Improving Flight Energy Efficiency in Simultaneous Transmission and Reception of Relay Using UAVs. In Proceedings of the 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC), Tangier, Morocco, 24–28 June 2019; pp. 967–972. [[CrossRef](#)]
8. Chen, J.; Ding, R.; Wu, W.; Liu, J.; Gao, F.; Shen, X.S. Multi-Agent Learning Based Packet Routing in Multi-Hop UAV Relay Network. In Proceedings of the ICC 2022–IEEE International Conference on Communications, Seoul, Republic of Korea, 16–20 May 2022; pp. 1–6. [[CrossRef](#)]
9. Wang, H.F.; Huang, C.S.; Wang, L.C. RIS-assisted UAV Networks: Deployment Optimization with Reinforcement-Learning-Based Federated Learning. In Proceedings of the 2021 30th Wireless and Optical Communications Conference (WOCC), Taipei, Taiwan, 7–8 October 2021; pp. 257–262. [[CrossRef](#)]
10. Gui, G.; Liu, M.; Tang, F.; Kato, N.; Adachi, F. 6G: Opening New Horizons for Integration of Comfort, Security, and Intelligence. *IEEE Wirel. Commun.* **2020**, *27*, 126–132. [[CrossRef](#)]
11. Qu, Y.; Dai, H.; Wang, H.; Dong, C.; Wu, F.; Guo, S.; Wu, Q. Service Provisioning for UAV-Enabled Mobile Edge Computing. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 3287–3305. [[CrossRef](#)]
12. Jiang, H.; Zhang, Z.; Wang, C.X.; Zhang, J.; Dang, J.; Wu, L.; Zhang, H. A Novel 3D UAV Channel Model for A2G Communication Environments Using AoD and AoA Estimation Algorithms. *IEEE Trans. Commun.* **2020**, *68*, 7232–7246. [[CrossRef](#)]
13. Jiang, H.; Zhang, Z.; Gui, G. Three-Dimensional Non-Stationary Wideband Geometry-Based UAV Channel Model for A2G Communication Environments. *IEEE Access* **2019**, *7*, 26116–26122. [[CrossRef](#)]
14. Masood, A.; Nguyen, T.V.; Truong, T.P.; Cho, S. Content Caching in HAP-Assisted Multi-UAV Networks Using Hierarchical Federated Learning. In Proceedings of the 2021 International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Republic of Korea, 20–22 October 2021; pp. 1160–1162. [[CrossRef](#)]
15. Yang, Z.; Chen, M.; Saad, W.; Hong, C.S.; Shikh-Bahaei, M. Energy Efficient Federated Learning Over Wireless Communication Networks. *IEEE Trans. Wirel. Commun.* **2021**, *20*, 1935–1949. [[CrossRef](#)]
16. Jing, Y.; Dong, C.; Qu, Y.; Zhou, F. Air-Ground Integrated Federated Learning: An Experimental Implementation. In Proceedings of the 2021 International Conference on Space-Air-Ground Computing (SAGC), Huizhou, China, 23–25 October 2021; pp. 161–162. [[CrossRef](#)]
17. Xu, M.; Wu, Y.; Zhang, H.; Yuan, L.; Wan, Y.; Zhou, F.; Wu, Q. GAN-Enabled Robust Backdoor Attack for UAV Recognition. In Proceedings of the 2022 7th International Conference on Communication, Image and Signal Processing (CCISP), Chengdu, China, 18–20 November 2022; pp. 474–478. [[CrossRef](#)]
18. Liu, J.; Shi, Y.; Fadlullah, Z.M.; Kato, N. Space-Air-Ground Integrated Network: A Survey. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 2714–2741. [[CrossRef](#)]
19. Wu, H.; Chen, J.; Zhou, C.; Li, J.; Shen, X. Learning-Based Joint Resource Slicing and Scheduling in Space-Terrestrial Integrated Vehicular Networks. *J. Commun. Inf. Netw.* **2021**, *6*, 208–223. [[CrossRef](#)]
20. Hao, L.; Ren, P.; Du, Q. Cooperative Regional Caching and Distribution in Space-Terrestrial Integrated Networks. In Proceedings of the 2021 IEEE/CIC International Conference on Communications in China (ICCC), Xiamen, China, 28–30 July, 2021; pp. 1042–1047. [[CrossRef](#)]
21. Yan, S.; Cao, X.; Liu, Z.; Liu, X. Interference management in 6G space and terrestrial integrated networks: Challenges and approaches. *Intell. Conver. Netw.* **2020**, *1*, 271–280. [[CrossRef](#)]
22. Arani, A.H.; Hu, P.; Zhu, Y. Fairness-Aware Link Optimization for Space-Terrestrial Integrated Networks: A Reinforcement Learning Framework. *IEEE Access* **2021**, *9*, 77624–77636. [[CrossRef](#)]
23. Mozaffari, M.; Saad, W.; Bennis, M.; Nam, Y.H.; Debbah, M. A Tutorial on UAVs for Wireless Networks: Applications, Challenges, and Open Problems. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2334–2360. [[CrossRef](#)]
24. Shahbazi, A.; Donevski, I.; Nielsen, J.J.; Di Renzo, M. Federated Reinforcement Learning UAV Trajectory Design for Fast Localization of Ground Users. In Proceedings of the 2022 30th European Signal Processing Conference (EUSIPCO), Belgrade, Serbia, 29 August–2 September 2022; pp. 663–666. [[CrossRef](#)]
25. Khelf, R.; Driouch, E.; Ajib, W. On the Optimization of UAV-Assisted Wireless Networks for Hierarchical Federated Learning. In Proceedings of the 2023 IEEE 34th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), Toronto, ON, Canada, 5–8 September 2023; pp. 1–6. [[CrossRef](#)]
26. Sharma, I.; Sharma, A.; Gupta, S.K. Asynchronous and Synchronous Federated Learning-based UAVs. In Proceedings of the 2023 Third International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP), Bangkok, Thailand, 18–20 January 2023; pp. 105–109. [[CrossRef](#)]
27. Pandey, S.R.; Tran, N.H.; Bennis, M.; Tun, Y.K.; Manzoor, A.; Hong, C.S. A Crowdsourcing Framework for On-Device Federated Learning. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 3241–3256. [[CrossRef](#)]

28. 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](#)]
29. Kang, J.; Xiong, Z.; Niyato, D.; Xie, S.; Zhang, J. Incentive Mechanism for Reliable Federated Learning: A Joint Optimization Approach to Combining Reputation and Contract Theory. *IEEE Internet Things J.* **2019**, *6*, 10700–10714. [[CrossRef](#)]
30. Wang, G.; Dang, C.X.; Zhou, Z. Measure Contribution of Participants in Federated Learning. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019; pp. 2597–2604. [[CrossRef](#)]
31. Weng, J.; Weng, J.; Zhang, J.; Li, M.; Zhang, Y.; Luo, W. DeepChain: Auditable and Privacy-Preserving Deep Learning with Blockchain-Based Incentive. *IEEE Trans. Dependable Secur. Comput.* **2021**, *18*, 2438–2455. [[CrossRef](#)]
32. Le, T.H.T.; Tran, N.H.; Tun, Y.K.; Han, Z.; Hong, C.S. Auction based Incentive Design for Efficient Federated Learning in Cellular Wireless Networks. In Proceedings of the 2020 IEEE Wireless Communications and Networking Conference (WCNC), Seoul, Republic of Korea, 25–28 May 2020; pp. 1–6. [[CrossRef](#)]
33. Bhandarkar, A.B.; Jayaweera, S.K.; Lane, S.A. User Coverage Maximization for a UAV-mounted Base Station Using Reinforcement Learning and Greedy Methods. In Proceedings of the 2022 International Conference on Artificial Intelligence in Information and Communication (ICAIIIC), Jeju Island, Republic of Korea, 21–24 February 2022; pp. 351–356. [[CrossRef](#)]
34. Wang, J.H.; Luo, P.C.; Xiong, H.Q.; Zhang, B.W.; Peng, J.Y. Parallel Machine Workshop Scheduling Using the Integration of Proximal Policy Optimization Training and Monte Carlo Tree Search. In Proceedings of the 2020 Chinese Automation Congress (CAC), Shanghai, China, 6–8 November 2020; pp. 3277–3282.
35. Jiang, H.; Zhang, Z.; Wu, L.; Dang, J. Three-Dimensional Geometry-Based UAV-MIMO Channel Modeling for A2G Communication Environments. *IEEE Commun. Lett.* **2018**, *22*, 1438–1441. [[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.