symmetry

MDPI

Article

Intelligent Computing Collaboration for the Security of the Fog Internet of Things

Hong Zhao, Guowei Sun, Weiheng Li, Peiliang Zuo *, Zhaobin Li and Zhanzhen Wei

Department of Electronic and Communication Engineering, Beijing Institute of Electronic Science and Technology (BESTI), Beijing 100070, China

* Correspondence: zplzpl88@bupt.cn

Abstract: The application of fog Internet of Things (IoT) technology helps solve the problem of weak computing power faced by IoT terminals. Due to asymmetric differences in communication methods, sensing data offloading from IoT terminals to fog and cloud layers faces different security issues, and both processes should be protected through certain data transmission protection measures. To take advantage of the relative asymmetry between cloud, fog, and sensing layers, this paper considers using physical layer security technology and encryption technology to ensure the security of the sensing data unloading process. An efficient resource allocation method based on deep reinforcement learning is proposed to solve the problem of channel and power allocation in fog IoT scenarios, as well as the selection of unloading destinations. This problem, which is NP-hard, belongs to the attribute of mixed integer nonlinear programming. Meanwhile, the supporting parameters of the method, including state space, action space, and rewards, are all adaptively designed based on scene characteristics and optimization goals. The simulation and analysis show that the proposed method possesses good convergence characteristics. Compared to several heuristic methods, the proposed method reduces latency by at least 18.7% on the premise that the transmission of sensing data is securely protected.

Keywords: fog IoT; physical layer security; encryption; reinforcement learning; resource allocation

Citation: Zhao, H.; Sun, G.; Li, W.; Zuo, P.; Li, Z.; Wei, Z. Intelligent Computing Collaboration for the Security of the Fog Internet of Things. *Symmetry* **2023**, *15*, 974. https://doi.org/10.3390/sym15050974

Academic Editors: Qinghe Zheng, Guan Gui, Ruidan Su and Rui Yu

Received: 31 March 2023
Revised: 14 April 2023
Accepted: 19 April 2023
Published: 24 April 2023

1. Introduction

With the rapid development and progress of electronic and communication technology, the Internet of Things (IoT) has become an important application technology in today's society. It refers to the interconnection of everyday physical objects via the internet, enabling data exchange and communication between these objects. The key characteristics of IoT include: interconnectivity, things-related services, heterogeneity, enormous scale and low cost. IoT terminals can obtain various information about the surrounding environment through their own sensing capabilities, and they report sensing data to IoT servers through various communication methods; this working mechanism has made it widely used in industrial, household, environmental monitoring and other scenarios [1–4]. The benefits of IoT include improved automation and efficiency, optimized resource utilization, reduced costs, and enhanced customer experiences. However, IoT also faces challenges such as compatibility and interoperability, privacy and security, massive data management, and impact on jobs. IoT is transforming industries and society in profound ways [5–8].

Fog computing refers to a new computing architecture and network structure that provides an intermediate layer between the cloud and end devices for data storage and processing. Fog computing could improve efficiency and real-time performance, enabling new applications. It combines cloud resources with edge nodes to achieve low latency, real-time response, and flexible resource allocation [9–11]. The IoT, composed of low-cost sensing terminals, naturally faces the problem of unbalanced processing capacity in the network, while the proposal and application of cloud computing and edge computing

are expected to effectively solve this problem. Due to its powerful computing power and storage capabilities, the cloud layer can process perceptual data quickly and efficiently. In contrast, the comprehensive capability of fog layer devices is between cloud layer and sensing layer devices, but the auxiliary effects brought by its ability are more easily enjoyed by the sensing device, as it is physically closer to the sensing devices, and it can receive sensing data earlier than the cloud layer [12–14]. From this perspective, there is a significant asymmetry between the IoT devices, fog layer nodes, and cloud layer servers in terms of their own capabilities and potential utility.

Due to the weak computing, storage, and battery capacities of IoT sensing terminals, the fog IoT also faces many security issues, among which the issue of secure transmission of sensing data should be taken seriously, considering that it is the foundation for the effective work of the fog IoT. Ensuring the transmission security of sensing data is also of great significance for user privacy and data security. In particular, the wireless transmission process in the sensing layer, as well as the wired transmission process between the fog layer and the cloud layer, should receive corresponding security protection, to prevent the data from being eavesdropped, resulting in information disclosure and network attacks [15–17].

1.1. Related Work

Resource allocation in the fog IoT has received widespread attention from scholars [18–22]. In order to reduce the energy consumption of transmission and offloading strategies for the fog IoT computing system supported by non-orthogonal multiple access (NOMA), the authors in [18] decomposed the target problem into two sub-problems with different optimization variables and proposed a multi-node cooperative transmission and computing algorithm. X. Huang et al. [19] jointly optimized drone trajectory and transmission power and computed an offload radio in unmanned aerial vehicle (UAV)-assisted fog IoT network scenarios by using problem decomposition and parallel optimization. Considering the complexity of task offloading and resource allocation for execution on fog IoT nodes, an intelligent method based on deep reinforcement learning was proposed in [21] to achieve efficient problem solving. A two-tier task scheduling scheme and an uplink and downlink power allocation factor are jointly optimized by problem transformation and decomposition in [22] for reducing the data processing delay as well as improving fairness among different users in fog IoT scenarios.

Building a secure fog IoT environment, especially ensuring the secure transmission of sensing data, is of great significance [23–29]. A framework consisting of two components, i.e., the security component and the trust management component, was proposed in [23] to address blackhole, sinkhole, sybil, and collusion challenges faced by fog IoT. O. A. Khashan [24] presented a hybrid proxy re-encryption scheme combining both lightweight symmetric and asymmetric encryption algorithms, which have been proven to ensure the safety of the data sharing process in fog IoT computing. In order to settle the inefficient problem of the fuzzy inference system that could be utilized to offer customized encryption algorithm decisions to the fog IoT users, a solution for determining the appropriate encryption scheme and the key was proposed in [25]. A secure data-sharing system based on key aggregate searchable encryption using blockchain was raised in [26], and the resistance of the proposed scheme against various attacks has been proven. J. Ren et al. [28] proposed a task offloading strategy with a centralized decision-making approach to facilitate the resource-constrained fog nodes for task offloading. To ensure the efficiency of federated learning (FL) while protecting the privacy of device input data in fog IoT, Y. Liu et al. [29] presented a secure aggregation protocol based on efficient additive secret sharing, the security and performance of which were analyzed.

It is also essential to conduct research on improving various components of the entire fog IoT. Formal methods [30,31] may include: the anomaly detection methodology; the data exchange mechanism; the analysis of collected data; the optimization of utilized resources; and the protection of private and sensitive data. Only when all these components are enhanced can the robustness and effectiveness of the system be further improved. With

these formal methods, the paper [32] presented a brief survey on the use of smartphone sensors to detect various anomalies in several fields, including environment, agriculture, healthcare and road/traffic conditions. Meanwhile, a cloud-based system architecture for fraud detection and customer profiling in the banking domain was proposed in [33] to address cyberattacks and financial frauds by using systematic risk assessment.

1.2. Contributions of This Paper

Table 1 summarizes the existing security protection methods for fog IoT. As can be seen, the existing research has mostly focused on efficient authentication, lightweight encryption algorithms, data privacy protection, modular security checks for the overall system and other issues in fog-based IoT scenarios. Less attention has been paid to the offloading and computing issues based on the secure transmission of perception data in the IoT system oriented to the cloud–fog–terminal architecture. This is an area that needs to be further enriched in the current work. Specifically, it is of research significance to leverage the asymmetric advantages of fog and cloud layers on sensing layer devices and to achieve security protection for sensing data transmission. Although current methods such as artificial intelligence, machine learning, and federated learning have been helpful in improving work execution efficiency, only federated learning can achieve efficient joint training of models under secure conditions, while artificial intelligence and machine learning do not inherently possess security capabilities. These methods cannot be directly applied to the scenarios considered in this paper. Artificial intelligence methods, physical layer security technologies, encryption algorithms, etc., need to be deeply integrated with fog IoT scenarios to provide security protection for the transmission and processing of sensing data. This paper investigates resource allocation issues in the fog IoT and aims to minimize the latency of sensing data processing and transmission on the premise of the security of data transmission. The key contributions of this paper are threefold:

- We construct a comprehensive resource decision-making model in which the sensing data are transmitted to the fog layer under the physical layer security protection, and the security of unloading to the cloud layer is ensured through encryption.
- To settle the NP-hard resource planning problem, we propose an intelligent method on the basis of deep reinforcement learning to realize the rapid allocation of transmission power, channels and of deciding whether to process the data in the cloud.
- Through generating a large number of snapshots corresponding to the scenario, we train the proposed model and verify the performance of the proposed method through the test set.

Table 1. Summary of existing research on security protection for the fog IoT.

Ref.	Application Area	Complexity	Potential Contribution
[23]	fog layer	moderate	trust management
[24]	fog/cloud layer	high	data security sharing
[25]	perceptual layer	moderate	encryption scheme selection
[26]	cloud layer	high	ciphertext retrieval
[28]	fog/cloud layer	moderate	task offloading
[29]	fog/cloud layer	moderate	secret sharing
[30,31]	ensemble	moderate	formal methods for detecting security anomalies
[32,33]	ensemble	moderate	anomaly detection
[34]	fog/perceptual layer	moderate	wireless secure transmission

This paper is organized as follows. Section 2 presents the system model and formulates the resource planning problem. Section 3 introduces the preliminary knowledge required

for this paper. Section 4 covers the proposed intelligent method. Section 5 provides the simulation results and performance analysis, and Section 6 gives the conclusions.

Notation: Throughout the paper, scalars are denoted by a nonboldface type, while vectors and matrices are denoted by a boldface type. $(\cdot)^T$ and $E\{\cdot\}$, respectively, signify matrix transpose and statistical expectation. Furthermore, ω_i represents the ith entry of the vector $\boldsymbol{\omega}$.

2. System Model and Problem Formulation

2.1. System Model

Figure 1 illustrates the resource decision-making process for fog IoT scenarios, in which there are K fog nodes and one cloud server serving U sensor terminals or user equipment (UE) located at the sensing layer. Compared to cloud servers, the available computing resources of fog nodes are relatively limited, but sensing data offloading from edge nodes to cloud servers requires a large amount of communication bandwidth. Queuing and transmission delays may affect the speed of data processing and information feedback. Meanwhile, although the wired transmission process of data has significantly stronger anti-interference capabilities compared to wireless transmission, the transmission of data in plaintext may pose significant security risks to IoT systems and devices. For example, the data transmission process may be intercepted, tampered with, or hidden by attackers, which may lead to attacks against IoT systems, affecting their normal operation.

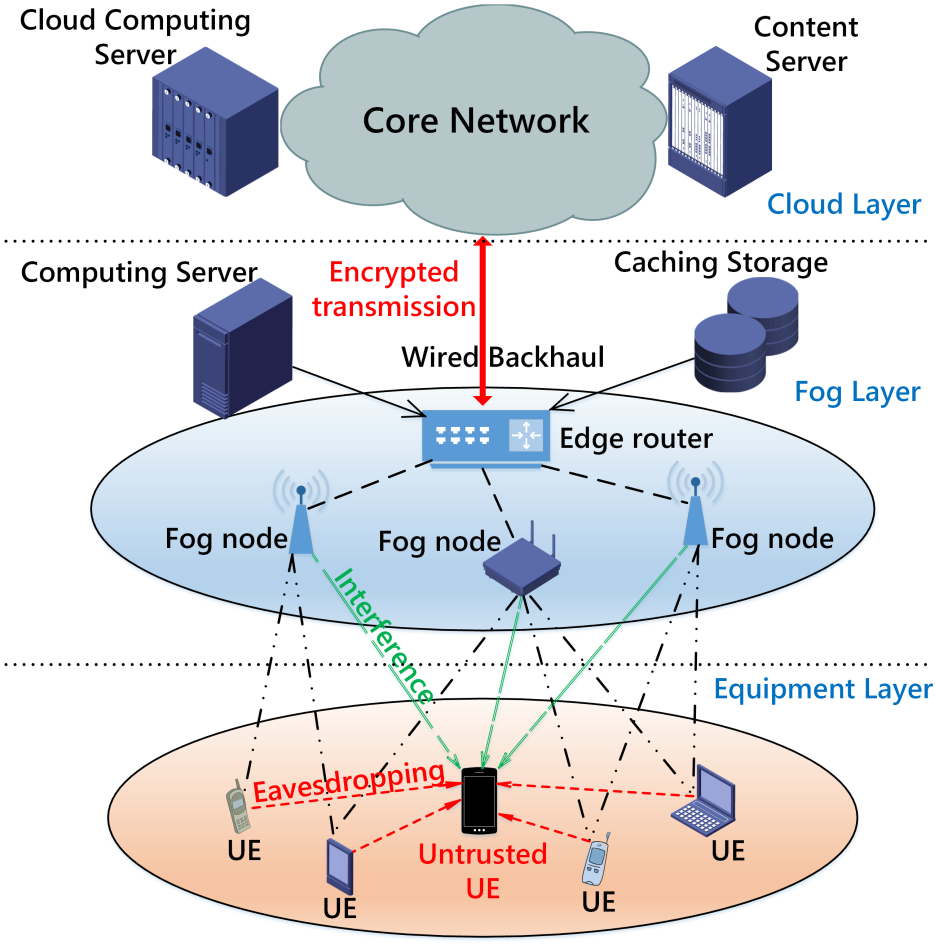

Figure 1. Schematic diagram of resource decision making for fog IoT scenarios.

This paper set that the wireless transmission from terminals to fog layer nodes is protected by physical layer security technology, and the wired transmission among the edge router and the cloud is protected by encryption technology. Specifically, considering

the weak computing power and battery life of sensor terminals, this paper relies on fog nodes to transmit artificial noise to ensure the secrecy capacity of the target channel (for more details, please refer to our previous paper [34]). We also assume that each fog node has the ability to eliminate self interference using co-time co-frequency full duplex technology, based on which fog nodes can effectively receive signals sent by sensing terminals (i.e., UEs). Furthermore, this paper sets that the bidirectional data transmission between the edge router node and the cloud must be protected through an encryption method. In this case, the offloading process of perception data in cloud–edge–terminal fog IoT scenarios is protected during transmission. We assume that the encryption and decryption rates of the edge router are E_R and D_R, respectively. To simplify the system model, it is assumed that the data transmission time between the fog node and the edge router is negligible and has undergone corresponding encryption protection. Set the encryption and decryption processing speeds of the cloud server to be E_C and D_C. It is assumed that the signal propagation time between the cloud end and the edge end is negligible, and the data transmission rate between the two is fixed to r. Set the processing speeds of fog node k and cloud server for sensing data to τ_k and τ_C, respectively. This paper considers the issue of computing and offloading data sensed by IoT devices and focuses on considering the duration between the data that are processed from the fog node or cloud node and ultimately fed back to the fog node. Set the proportion of the feedback information obtained after processing the data at the fog node and cloud node in the input data amount (i.e., the amount of data before processing) as β.

2.2. Problem Formulation

If the data from the terminal u, $u = 1, 2, \ldots, U$ is processed at the fog node k, $k = 1, 2, \ldots, K$, the corresponding time required to obtain feedback information can be expressed as:

$$T_{u,k}^{f} = \frac{Z_u \times \alpha_{u,k}}{\tau_k}, \tag{1}$$

where Z_u denotes the amount of data sensed by terminal u, $\alpha_{u,k} = 0$ or 1 is the indicator function. When $\alpha_{u,k} = 1$, it represents that the sensing data of node u is processed at fog node k. Otherwise, the data will be processed in the cloud. Considering the presence of multiple terminal sensing data aggregated at the fog node, the average data processing delay of terminal u at fog node k can be expressed as:

$$\bar{T}_{u,k}^{f} = \begin{cases} \frac{1}{\sum_{u=1}^{U} \lambda_{u,k}} \times \sum_{u=1}^{U} \frac{\lambda_{u,k} \times Z_u \times \alpha_{u,k}}{\tau_k}, if \sum_{u=1}^{U} \lambda_{u,k} \neq 0 \\ 0, if \sum_{u=1}^{U} \lambda_{u,k} = 0 \end{cases}, \tag{2}$$

where $\lambda_{u,k}$ is an indicator function for terminal u to report sensing data to fog node k. If the sensing data are selected for processing on the cloud server, the corresponding total delay can be expressed as:

$$\begin{aligned} T_{u,k}^{C} = & \frac{Z_u \times (1 - \alpha_{u,k})}{\tau_C} + \frac{Z_u \times (1 - \alpha_{u,k})}{r} + \frac{\beta \times Z_u \times (1 - \alpha_{u,k})}{r} + \frac{Z_u \times (1 - \alpha_{u,k})}{E_R} + \\ & \frac{Z_u \times (1 - \alpha_{u,k})}{D_C} + \frac{\beta \times Z_u \times (1 - \alpha_{u,k})}{E_C} + \frac{\beta \times Z_u \times (1 - \alpha_{u,k})}{D_C}, \end{aligned} \tag{3}$$

where the first item in the right formula represents the delay of processing the sensing data of node u on the cloud server; the second and third items represent the transmission delay and reverse transmission delay of sensing data from the fog node (edge router) to the cloud server, respectively. The fourth and fifth items represent the encryption delay of sensing data in the edge router and the decryption delay in the cloud server, respectively. The sixth and seventh items denote the encryption delay at the cloud server and the decryption

delay at the edge router after the data being processed by the cloud server, respectively. Equation (3) can be further simplified as:

$$T_{u,k}^{C} = Z_u \times (1 - \alpha_{u,k}) \times \left(\frac{1}{\tau_C} + \frac{1+\beta}{r} + \frac{1}{E_R} + \frac{1}{D_C} + \frac{\beta}{E_C} + \frac{\beta}{D_C} \right). \tag{4}$$

Considering the existence of multiple terminal sensing data aggregated in the cloud server, the delay of processing the data from the terminal u in the cloud service and returning information to the edge router can be expressed as:

$$\bar{T}_{u,k}^{C} = \begin{cases} \frac{\sum_{u=1}^{U}\sum_{k=1}^{K} T_{u,k}^{C}}{\sum_{u=1}^{U}\sum_{k=1}^{K}(1-\alpha_{u,k})}, if \sum_{u=1}^{U}\sum_{k=1}^{K}(1-\alpha_{u,k}) \neq 0 \\ 0, if \sum_{u=1}^{U}\sum_{k=1}^{K}(1-\alpha_{u,k}) = 0 \end{cases}. \tag{5}$$

Based on the premise of secure transmission at the sensing layer of the IoT in [34], and with the goal of minimizing the latency of sensing data processing, the security resource planning issues at the cloud edge of fog IoT can be summarized as follows:

$$\min_{\lambda_{u,k}, p_{k,u}, \alpha_{u,k}} \sum_{k=1}^{K}\sum_{u=1}^{U} \lambda_{u,k}(T_{u,k} + \Phi_{u,k}) \tag{6a}$$

$$\text{s.t.} \quad T_{u,k}(t) = \frac{Z_u}{C_{\text{sec}}^{u}} \tag{6b}$$

$$\Phi_{u,k} = \bar{T}_{u,k}^{f} + \bar{T}_{u,k}^{C}, \forall u, k \tag{6c}$$

$$C_{\text{sec}}^{u} = B\log_2(1 + \eta_{u,k}) - B\log_2(1 + \eta_{u,e}) \tag{6d}$$

$$\sum_{k=1}^{K} \lambda_{u,k} = 1, \lambda_{u,k} = 0, 1 \quad \forall u, k \tag{6e}$$

$$\sum_{u=1}^{U} p_{k,u} \leq P_k, \forall k, \tag{6f}$$

where C_{sec}^{u} denotes the secrecy capacity of data sent by device u. B, $\eta_{u,k}$ and $\eta_{u,e}$ respectively represent the channel bandwidth, the signal to noise ratio (SNR) of the target channel (i.e., the channel between device u and fog node k), and the SNR of the eavesdropping channel (i.e., the channel between device u and the untrusted node). P_k and $p_{k,u}$ respectively denote the total available transmission power of the fog node k and the artificial noise transmission power allocated to protect (for more details on the physical layer security approach, please refer to [34]). the device u.

We could observe that the complexity of problem (6) is further improved compared to the planning channel and power resources of the fog nodes. It belongs to the NP-hard problem of mixed integer nonlinear programming. It is difficult to directly obtain the optimal solution through mathematical programming methods. In this case, this paper explores the use of artificial intelligence methods combined with heuristic methods to efficiently solve the problem in Section 4.

3. Preliminaries

3.1. Reinforcement Learning

Reinforcement learning aims to obtain model parameters for using high reward actions in different environmental states through certain interactions with the environment [35–40]. Reinforcement learning can be described by the Markov Decision Process (MDP), in which an agent takes decision-making actions in a given environment, driving the state of the environment to change with a certain probability, and it receives corresponding rewards. Specifically, if the environmental state at time t is represented by s_t, the agent takes action

a_t, it would receive a numerical reward r_t, and then the environment enters the state of the next moment s_{t+1}. Mathematically, the change (learning) process of reinforcement learning can be described by quadruple $\langle \boldsymbol{S}, \boldsymbol{A}, \boldsymbol{P}, \boldsymbol{R} \rangle$, in which $\boldsymbol{S}, \boldsymbol{A}, \boldsymbol{P}, \boldsymbol{R}$ respectively denote the state space, action space, reward set, as well as the state transition probability matrix.

The agent's decision-making strategy $\pi_t(s, a)$, which indicates the action selection recommendation with state s at time t, could be updated when the experience sequence $\{(s_t, a_t, r_t, s_{t+1}), \ldots\}$ is acquired during the process of the agent's continuous exploratory learning. The goal of reinforcement learning is to enable the agent to obtain a series of maximum cumulative rewards in future decisions. Mathematically, we can express it as

$$R_t = \sum_{l=0}^{\infty} \vartheta^l r_{t+l}, \tag{7}$$

where $\vartheta \in [0, 1]$ represents the discount ratio.

The Q-learning method is widely adopted in current reinforcement learning methods [39,40], in which the Q-value of the model can be updated through the interaction with the environment. The Q-value reflects the utility value of adopting action a under the condition of policy π in environmental state s. The process can be written as

$$Q^{\pi}(s, a) = E[R_t \mid s_t = s, a_t = a]. \tag{8}$$

Denote $Q^*(s, a) = \max_{\pi} Q^{\pi}(s, a)$ as the optimal action value function, where we can obtain the following formula by using, according to the Bellman optimality equation,

$$Q^*(s, a) = E\left[r_{t+1} + \vartheta \max_{a'} Q^*\left(s', a'\right) \mid s_t = s, a_t = a\right], \tag{9}$$

where s' represents the new state after the action a was selected and performed.

Q-learning aims to continuously update the optimal action value function by utilizing the experience sequence acquired through the interaction between the agent and environment. Denote the estimated Q-value in the iterative process as $q(s_t, a_t)$, where we could use the following formula to express the update process of Q-learning

$$q(s_t, a_t) \leftarrow q(s_t, a_t) + \xi(r_t + \vartheta \max_{a'} q(s_{t+1}, a_{t+1}) - q(s_t, a_t)), \tag{10}$$

where $\xi \in [0, 1]$ denotes the learning rate.

To quickly obtain the Q-value and the policy relationship, the balance between exploration and exploitation should be handled by the agent. The exploitation process represents the use of learned decision-making strategies by the agent, i.e., the selection of actions based entirely on the output of the model, while the exploration process denotes that the agent selects actions in a certain random manner to explore the content of the decision. It should be added that an agent should not only bias toward a certain process to avoid situations where model training is not in place or model decisions are excessively random. The ε-greedy algorithm is a widely utilized balancing method that adopts the following probability mechanism [35–40]

$$a = \begin{cases} \arg\max_a q(s, a), & \text{with probability } 1-\varepsilon \\ \text{a random action}, & \text{with probability } \varepsilon \end{cases}. \tag{11}$$

3.2. Deep Reinforcement Learning

Under normal circumstances, reinforcement learning can efficiently train the decision model to guide the agent in selecting reasonable actions. Unfortunately, reinforcement learning will lead to a decrease in training efficiency and decision-making efficiency due to the large database corresponding to the Q-table, when the number of elements in state space or action space increases significantly. In this case, the deep reinforcement learning (DRL), which combines both reinforcement learning and deep learning, was proposed [41]. Unlike directly recording the correspondence between state elements and action elements,

DRL aims to rely on deep neural networks (DNNs) to achieve mapping between states and actions, i.e., using the neural network to intelligently estimate the action value function. Due to the ability of DNNs to fit complex nonlinear relationships, DRL can generally handle the problem with a large amount of elements brought about by large state space and action space well.

The alternative process of DRL to reinforcement learning can be written as $Q^*(s,a) \approx Q(s,a|\boldsymbol{\theta})$. The input and output of the DNNs in the DRL are respectively a state s and the Q-value of each action among the action space. $q(s,a|\boldsymbol{\theta})$ denotes the output of the DNNs, and it is determined only by the weights $\boldsymbol{\theta}$ of the DNNs. To update $\boldsymbol{\theta}$, the back propagation of the learning process is usually adopted. Similar to Equation (9), the following equation should be adopted to acquire the target in the Actor–Critic learning model with dual DNN architecture

$$L(\boldsymbol{\theta},\boldsymbol{\theta}') = \mathrm{E}\left[\left(r(s,a)+\vartheta \max_{a'} Q(s',a' \mid \boldsymbol{\theta}') - Q(s,a \mid \boldsymbol{\theta})\right)^2\right], \tag{12}$$

where $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ respectively represent the weights of the main network and the target network.

4. Proposed Intelligent Method

By analyzing problem (6), it can be concluded that the solution involves selecting values for the three parameters $\lambda_{u,k}, p_{k,u}, \alpha_{u,k}, \forall u,k$ in order to minimize the average transmission and processing delays corresponding to the target (6a). If the ergodic method is used to obtain the parameters, it can be calculated that the number of parameters to be estimated is $K^U \times M^K \times 2^U$, where M denotes the number of optional power allocation methods. This requires significantly high computational power for the traversal process and brings significantly large delays to the decision-making process. Obviously, it is not suitable for fog IoT scenarios that require high timeliness in resource decision making. This paper considers using a deep reinforcement learning method that has advantages in decision-making efficiency to solve the problem. The method constructs the decision-making environment faced by the executive into a mathematical vector, and forms a mapping relationship with possible decision-making results. Through training based on a large number of sample data, an automatic mathematical model with good decision-making performance is obtained.

4.1. Preliminary Exploration of the Method

Intuitively, the parameters of DRL can be extended and supplemented based on the method proposed in our previous work [34] to match the problem scenario. In this case, considering that in the cloud–edge–end resource decision-making process, new environmental factors mainly include the encryption and decryption rate of the edge router, the encryption and decryption processing speed of the cloud server, the processing speed of the fog node k and the cloud server for sensing data, the data transmission rate, and the proportion of feedback information to the input data, the corresponding state of the environment can be set as

$$s = \{\underbrace{Z_u}_{U}, \underbrace{C_{u,k}}_{U\times K}, \underbrace{p_u}_{U}, \underbrace{g_{u,e}}_{U}, \underbrace{g_{k,e}}_{K}, E_R, D_R, E_C, D_C, r, \tau_R, \tau_C, \beta\}_{1\times(U\times K+3U+K+8)}. \tag{13}$$

By referring to the optimization objective of problem (6), the action space for security resource decisions can be set as follows:

$$a = \{\underbrace{\lambda_{u,k}}_{U\times K}, \underbrace{\tau_m}_{M}, \underbrace{\alpha_{u,k}}_{U}\}_{1\times(U\times K+M+U)}. \tag{14}$$

From Equations (13) and (14), it can be concluded that the current state space size and action space size are respectively $Q_{Z_u}^U Q_{C_{u,k}}^{UK} Q_{p_u}^U Q_{g_{u,e}}^U Q_{g_{k,e}}^K Q_{E_R} Q_{D_R} Q_{E_C} Q_{D_C} Q_r Q_{\tau_R} Q_{\tau_C} Q_\beta$ and $2^U \times K^U M^K$, where $Q_{(\cdot)}$ denotes the quantization progression of parameter $(\cdot)$. The two

spatial sizes have significantly exceeded the corresponding sizes in [34], which means that the DRL method faces more complex decision-making scenarios with more state action pairs than the proposed method in [34], which in turn can lead to greater training delays, and this is detrimental to the timely updating and adjustment of models. In the next subsection, this paper proposes a lightweight intelligent decision-making method to reduce the complexity of model training and updating.

4.2. Lightweight Decision-Making Method

Compared to constructing all environmental impact factors in the cloud–edge–end decision scenarios of the fog IoT into a state space, this paper observes that there is a low coupling characteristic between the sensing layer decision scenario and the cloud–edge decision scenario. In other words, it is possible to achieve separation between how to securely and quickly transmit terminal sensing data to the fog node and determining whether the sensing data are processed in the cloud or in the fog layer. This means that the decision of the latter can be made after the decision of the former, rather than linking all the influencing factors together. In view of this, this paper considers constructing a DRL model separately for the latter decision scenario. Compared to the decision model in the previous subsection, this method reduces the computational complexity from a multiplicative relationship to an additive relationship. Mathematically, the spatial size of the state and action pairs of the decision model in the previous subsection is

$$2^U \times K^U M^K Q_{Z_u}^U Q_{C_{u,k}}^{UK} Q_{p_u}^U Q_{g_{u,e}}^U Q_{g_{k,e}}^K Q_{E_R} Q_{D_R} Q_{E_C} Q_{D_C} Q_r Q_{\tau_R} Q_{\tau_C} Q_\beta. \tag{15}$$

The overall spatial size of the state and action pairs for the new lightweight method proposed in this paper is at most

$$Q_{Z_u}^U Q_{C_{u,k}}^{UK} Q_{p_u}^U Q_{g_{u,e}}^U Q_{g_{k,e}}^K K^U M^K + 2^U \times Q_{Z_u}^U Q_{E_R} Q_{D_R} Q_{E_C} Q_{D_C} Q_r Q_{\tau_R} Q_{\tau_C} Q_\beta. \tag{16}$$

It can be seen that the decision space size of the new method has been significantly reduced, i.e., the training and updating efficiency of the decision model has been significantly improved. The details of the new method are described below.

State Space

Based on the safe and rapid transmission of terminal sensing data to fog nodes, that is, using the algorithm proposed in [34] as the decision-making premise, it can be determined that the sensing data of each terminal currently uses the corresponding fog node as the starting point for data transmission, and whether the decision-making data are processed in the local fog layer or transmitted to the cloud server for processing and transmitting back. The current relevant impact factors are constructed into an environmental state, which can be expressed as

$$s = \{\underbrace{Z_u}_{U}, E_R, D_R, E_C, D_C, r, \tau_R, \tau_C, \beta\}_{1\times(U+8)}. \tag{17}$$

4.3. Action Space

From the objective function of problem (6), it can be observed that with the operation of the algorithm proposed in [34], the relevant parameters to be solved (i.e., $\lambda_{u,k}, p_{k,u}, \forall u, k$) have obtained results, while the unknown parameters are mainly $\alpha_{u,k}, \forall u, k$. Therefore, the action of the deep reinforcement learning model should be set to

$$a = \{\underbrace{\alpha_{u,k}}_{U}\}_{1\times(U)}. \tag{18}$$

4.4. Action Reward

The goal of problem (6) is to minimize the processing of sensing data and feedback delay by discarding the parameters that have been optimized through the algorithm

proposed in [34] (i.e., $T_{u,k}$). The parameters to be optimized are $\Phi_{u,k}, \forall u, k$. This paper sets the reward of the proposed method as

$$r_w = -\sum_{k=1}^{K}\sum_{u=1}^{U}\Phi_{u,k}. \tag{19}$$

This means that the lower the average processing and return latency of the sensing data at each terminal, the greater the corresponding reward value, which in turn can guide the method to adjust to low latency decision parameters to improve the decision-making effect.

4.5. Method Summary

We summarize the proposed Secure Lightweight Resource Allocation method of the Fog IoT (S-LFRA) method in Algorithm 1 with the process diagram in Figure 2. The method relies on the decay process of the ε-greedy algorithm to execute the training process of the DNN model. By randomly obtaining action values, a state action pair is obtained, and a quad sample is formed with the corresponding rewards and the next state. It is added to the training memory for use. By randomly obtaining batch samples from the memory, the parameters of the main network are updated using the Adam algorithm [35,36], and they timely update the main network parameters to the target network to ensure a smooth training and learning process. Through the implementation of Algorithm 1, the final output for the three types of parameters $\alpha_{u,k}, \lambda_{u,k}, \tau_m, \forall u, k, m$ can be obtained, which can then be intuitively converted to the set of actions that the edge router needs to perform.

Algorithm 1 The proposed S-LFRA method

Input: $E_R, D_R, E_C, D_C, r, \tau_R, \tau_C, \beta, Z_u, \forall u$.
Output: $\alpha_{u,k}$, $\lambda_{u,k}$, and $\tau_m, \forall u, k, m$.

Initialize replay memory $\mathbf{\Gamma}$, random parameters of models $\boldsymbol{\theta} = \boldsymbol{\theta'}$, $\varepsilon, \varepsilon_{\text{delay}}, \varepsilon_{\min}$, training rounds I, learning threshold ϖ, update frequency of the target network F, and $j = 0$.

Run Algorithm 1 in [34], and obtain the output $\lambda_{u,k}$ as well as $\tau_m, \forall u, k, m$.
While $\varepsilon > \varepsilon_{\min}$ **Do**
  $\varepsilon \leftarrow \varepsilon \times \varepsilon_{\text{delay}}$
  For $i \leftarrow 1, \ldots, I$ **Do**
    Take a random value from 0 to 1, and execute Equation (11);
    Execute the selection of action a, calculate the reward received, and store the quad (s, a, r_w, s') into $\mathbf{\Gamma}$;
    If $i > \varpi$ **Do**
      Take a batch of experience samples randomly from $\mathbf{\Gamma}$;
      Calculate $L(\boldsymbol{\theta}, \boldsymbol{\theta'})$ using Equation (12);
      Let $j = j + 1$;
    End If
    If j **mod** F **= 0 Do**
      $\theta' \leftarrow \theta, s \leftarrow s'$;
      Let $j = j + 1$;
    End If
  End For
End While

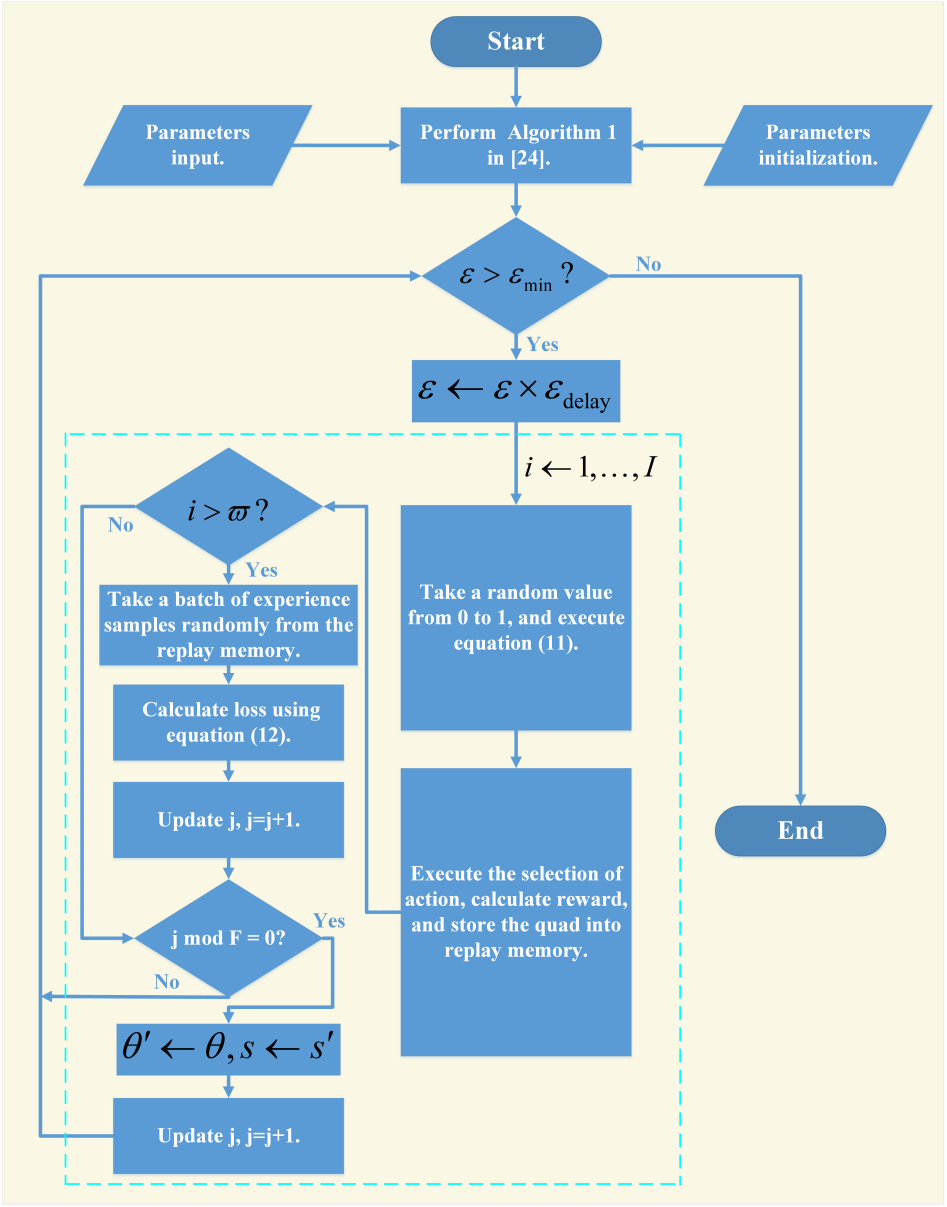

Figure 2. Process diagram of Algorithm 1.

To avoid the problem of degradation of traditional network structures, the Residual Network (ResNet) [42] is adopted in the proposed S-LFRA method. Specifically, ResNets employ skip connections to overcome the issues of degradation and difficulty in training deep neural networks. By skipping some layers during training, the gradients can flow directly through the shortcut paths, mitigating the gradient vanishing/exploding effect. This enables the training of very deep networks beyond 100 layers. The proposal of ResNet is a milestone that unleashes the depth potential of neural networks and spearheads a new wave of deep learning revolution. Furthermore, the Relu activation function is employed in the training process of the ResNet. In order to avoid violent fluctuations in the process of training, the dual neural network is adopted in the method; the replay memory and small batch learning approaches are also utilized.

The advantages of Algorithm 1 are mainly reflected in efficiency and forward compatibility. The former ensures that the proposed model can quickly make decisions with high rewards, and the latter ensures that the trained model has good interoperability. Its disadvantages are mainly that the model is required to be updated and trained, and the training complexity is high. Fortunately, this disadvantage can be overcome by selecting

high-performance processing nodes and pre-computing. The complexity of the proposed method mainly gathers at the training stage, as a lot of differentiation and other operations are required at this stage, while the complexity is quite low in the testing or application stages, since only relatively simple forward linear and nonlinear calculations are needed in these stages. This further ensures the applicability of the method, as the calculations can be performed in advance on devices or networks with strong comprehensive capabilities. Upon applying the trained model to practical scenarios, one merely needs to input a state vector of identical form to that used in training. The model can then promptly generate an action vector directing resource allocation. The proposed method possesses forward compatibility. Specifically, when the scenario nodes mapped to the trained model are complex, the model remains applicable to more simple fog-based IoT scenarios. For simpler scenarios, it can be achieved through setting corresponding prior values to zero or infinity. This indicates that the proposed method has a degree of interoperability.

5. Simulation and Performance Analysis

This paper adopts the Keras platform [43,44] to perform the simulation. Keras is an open source deep learning library written in Python. It provides a high-level API for building and training deep learning models. Keras makes it easy to start with deep learning and provides useful abstractions that can be used in Theano, TensorFlow, and CNTK. With its simple and consistent interface, Keras accelerates experiments in deep learning and machine learning. The settings of parameters related to the proposed method are summarized in Table 2. The encryption and decryption rate of the edge router, the encryption and decryption processing speed of the cloud server, the data transmission rate between the cloud and the edge, the processing speed of the sensing data by the fog node and the cloud server, and the proportion of the feedback information obtained after processing by the fog node and the cloud node in the input data volume are randomly generated according to the data range shown in the table. The basic settings of parameters are consistent with the simulation in [34]. This paper verifies the performance of methods by generating multiple samples. Specifically, each sample corresponds to a scenario snapshot, which includes the node state, sensing data, and other variables of the scenario. The state vector is a mathematical description of the snapshot. With a large number of snapshots, we can quickly and effectively train and test the decision model of the proposed method.

Table 2. Parameters in the simulation.

Hyper-Parameter	Value
ϑ, ε, ξ, ϖ	0.9, 0.6, 0.01, 7
Decay rate of ε, ε_{decay}	0.99
The minimal value of ε, $\varepsilon_{\min}$	0.001
Experience-replay memory capacity $\mathbf{\Gamma}$	2000
Update frequency F of the target network	500
Experience–replay minibatch size	32
U, K, M	5, 2, 8
δ for training/test set	5000/1000
Channel bandwidth B	1 MHz
$p_u, \forall u \in \boldsymbol{U}$	−5∼10 dBm
$Z_u, \forall u \in \boldsymbol{U}$	10∼50 Mbit

Table 2. *Cont.*

Hyper-Parameter	Value
$P_k, \forall k \in \boldsymbol{K}$	20∼30 dBm
$g_{k,e}, g_{u,e}, g_{u,k}, \forall k \in \boldsymbol{K}, \forall u \in \boldsymbol{U}$	−80∼−120 dB
N_{train}, N_{test}	2000, 500
Number of layers of the used RESNET	8
Number of neurons in each layer of RESNET	64
E_R, D_R	2∼5, 3∼8 Mbit/s
E_C, D_C	10∼20, 15∼40 Mbit/s
r, β	3∼10 Mbit/s, 0.1∼0.5
τ_k, τ_C	5∼15, 20∼60 Mbit/s

Figure 3 shows the verification process for the method proposed in this paper, where a large number of snapshots are generated to form training sets and test sets to update and verify the performance of the eight-layer ResNet model parameters. Meanwhile, to demonstrate the performance of the proposed method, the following methods are adopted.

- LFRA: This method corresponds to Method S-LFRA that does not perform Step 4.
- SIRA: The method proposed in [34].
- FCRA: A variation of SIRA. This method finds the optimal power allocation through traversal operation on the basis of using a randomly specified connection relationship between sensing devices and fog nodes.
- APRA: A variation of SIRA. This method finds the optimal connection matching relationship through traversal operation on the basis of using uniform power allocation mode.
- RanF: This method randomly determines whether the sensing data are processed in the cloud or in the fog layer.
- MidV: This method takes the average value of all the sensing data to be processed as the threshold value. Data exceeding this threshold value are processed in the cloud; otherwise, it is processed in the fog layer.
- S-LFRA-O: This method obtains the optimal solution for selecting whether to process in the cloud layer through traversal. It has a high complexity and is not practical in application.

Firstly, considering that the operational complexity of the proposed method will greatly affect the decision-making efficiency of the edge router in fog IoT scenarios, we specifically compare the convergence speed of the proposed LFRA method with the previously proposed SIRA method, to ensure that the proposed LFRA method does not significantly increase computational complexity and runtime latency. For ease of observation, the simulation only shows the performance of the method for one snapshot, and the performance results are shown in Figure 4. Although the convergence values of the two methods are not comparable, it can be seen from the figure that the two methods have significantly different convergence performances. The LFRA method has a faster convergence speed than the SIRA method, which is due to the smaller state and action space compared to the SIRA method. This also fully demonstrates that the proposed LFRA method has lower complexity compared to the SIRA method and preliminarily proves that the proposed method has strong application deployment feasibility.

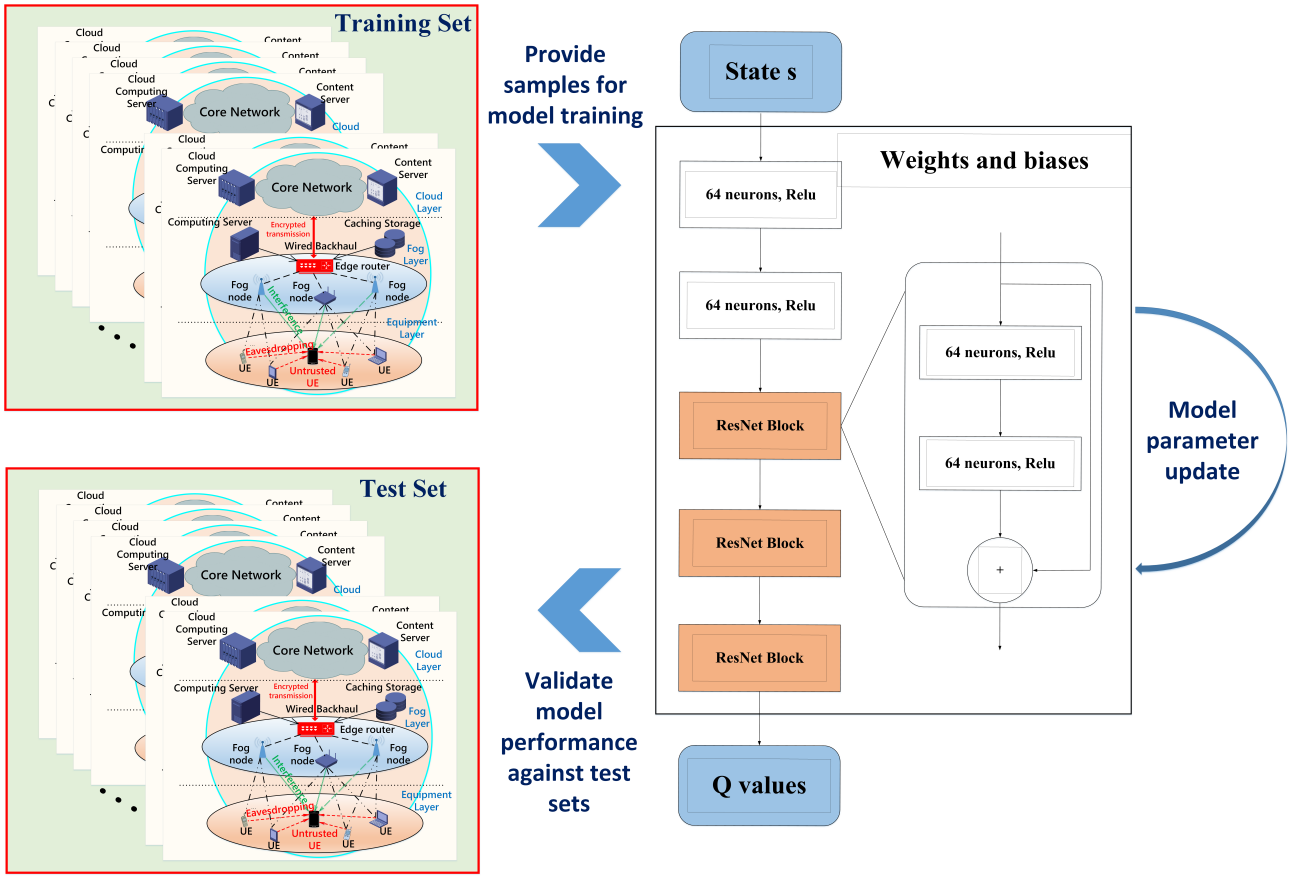

Figure 3. Schematic diagram of the simulation process. It relies on a large number of generated snapshots to compose training sets and test sets to update and verify the performance of ResNet model parameters.

Next, this paper verifies the performance of the proposed S-LFRA algorithm and selects three comparison methods, namely, the S-LFRA-O method, the SIRA+RanF method, and the SIRA+MidV method. The S-LFRA-O method requires calculation of all possible combinations, which is very complex and not feasible in reality. The simulation results are shown in Figure 5. As can be seen from the figure, method S-LFRA is significantly superior to the SIRA+RanF method and the SIRA+MidV method in terms of delay (reward) performance. Overall, the performance of the SIRA+RanF method is relatively unstable, and the performance of the S-LFRA method is basically equivalent to that of the S-LFRA-O method, with only a slightly lower reward on individual snapshots. Considering the poor feasibility of the S-LFRA-O method in reality, this indicates the rationality and performance advantages of the proposed S-LFRA method. Among them, LFRA-O is a variant of LFRA. This method finds the optimal action through traversal operations, which takes a long time and can reflect performance boundaries. RanF is a random method in which the system randomly selects whether the sensing data are offloaded to cloud computing or processed in fog. MidV is a median allocation method that takes the average of all data volumes, uploads data that are greater than the average for cloud processing, and processes data that are less than the average in the fog layer. From the simulation results, it can be seen that LFRA and LFRA-O almost completely coincide, and in practical applications, LFRA-O, a traversal operation method, is not applicable. By relying on 2000 snapshots for testing, this paper finds that the proposed method S-LFRA has an average increase in latency (reward) values by more than 37.3% compared to methods SIRA+RanF and SIRA+MidV. Therefore, the results fully reflect the advantages of the LFRA method proposed in this paper.

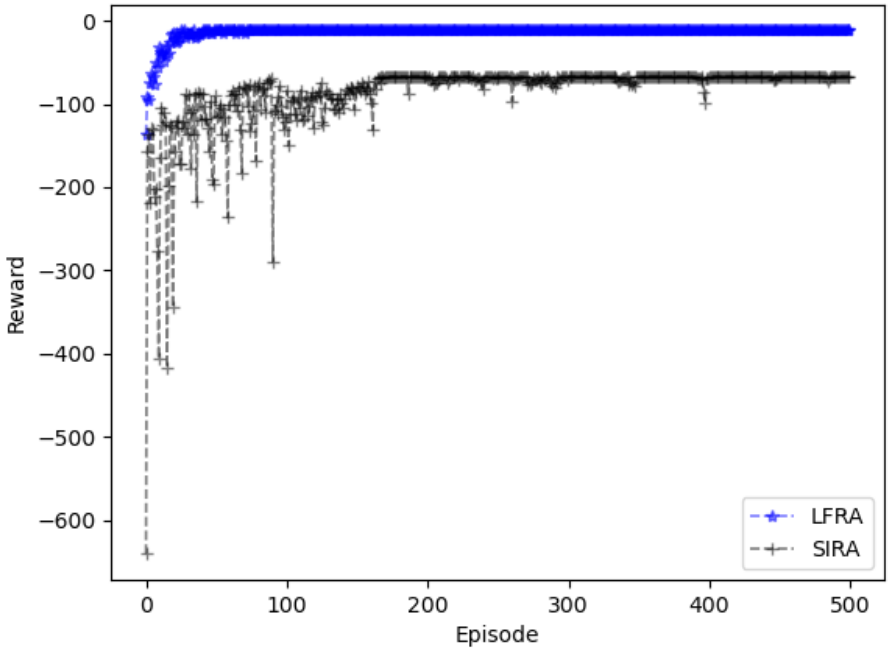

Figure 4. Convergence performance comparison of proposed methods.

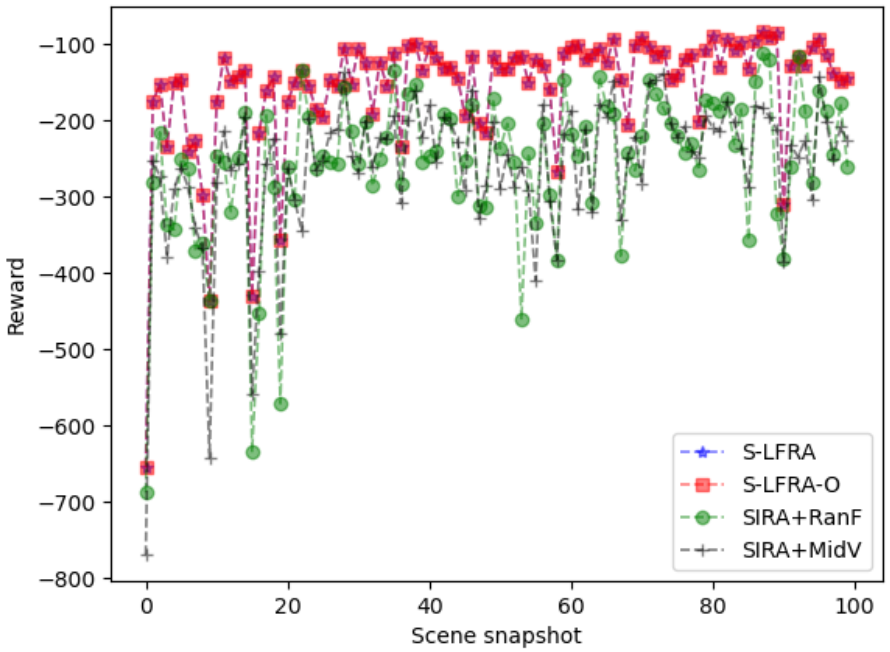

Figure 5. Performance results of S-LFRA method and comparative methods on random snapshots.

Finally, this paper utilizes multiple sensing layer decision-making methods, which are combined with the cloud–edge–end LFRA method, to simulate on a test set to demonstrate the rationality of the proposed method. The methods used for the sensing layer include SIRA, APRA, FCRA, and SIRA-R. These methods are described in [34]. The simulation results are shown in Figure 6. In order to more clearly observe the differences between the four methods, part of the data in Figure 6 are zoomed in and are displayed in Figure 7. Combining the two figures, it can be clearly seen that the effect of combining SIRA with LFRA is significantly better than that of combining other methods with LFRA. Through experiments on an additional 2000 samples (snapshots), this paper observes that the proposed method S-LFRA yields an average of at least 18.7% higher reward compared to these other three methods. This fully demonstrates the rationality and advantages of this

chapter's rapid allocation of sensing layer channel and power resources and cloud side resource planning decisions in the fog IoT, and it significantly reduces latency overall.

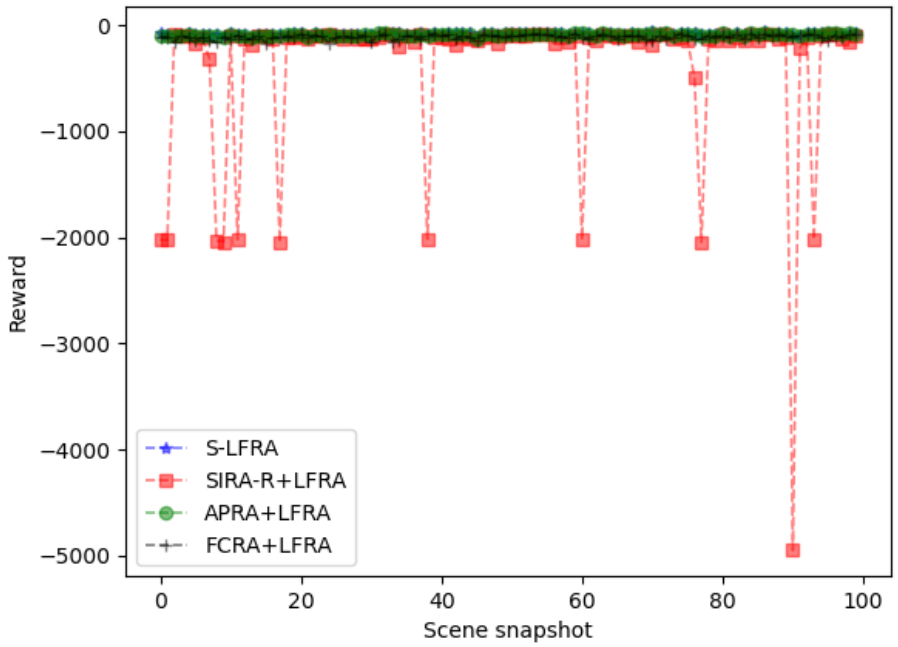

Figure 6. Performance comparison of LFRA combined with different methods.

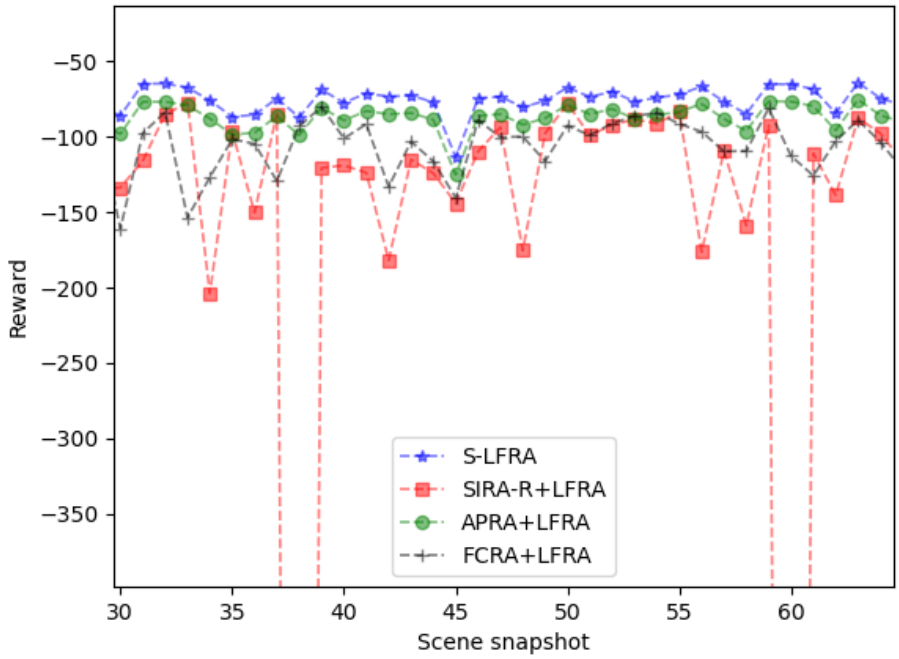

Figure 7. Enlarged display of some results in Figure 6.

6. Conclusions

This paper considered the issue of security resource decision making for the fog Internet of Things under the cloud–edge–end architecture. Through analyzing the complexity of the problem and solution methods, it is proposed to independently solve the computing and offloading problem of the sensing data of the IoT devices after implementing the resource decisions corresponding to the wireless security transmission of the sensing layer. The problem aims to minimize the latency of sensory data processing in fog IoT scenarios on the premise of secure transmission of sensory data at the cloud edge layer. A lightweight

intelligent decision-making method that matches the depth of the scene on the basis of deep reinforcement learning technology was proposed, where the key factors such as status, actions, rewards, and network parameters were reasonable designed. A large number of snapshot simulations have verified the proposed method, indicating that the convergence speed of the method is fast, that it has lower model training and update complexity, and that is has significantly better resource decision-making performance compared to multiple heuristic methods.

The future work can be explored in the following three directions: Firstly, given the existence of various encryption techniques between the fog computing layer and cloud layer, it is worthwhile to study how to achieve a balance between efficiency and security in data collaborative processing. Secondly, it is worth considering providing physical layer security protection for the downstream processes of the perception layer so as to protect the data security of IoT devices without consuming their limited computing resources. Finally, it is significant to consider how to achieve secure data transmission and fast processing under the scenario where there are multiple untrusted third parties with unknown prior information.

Author Contributions: Conceptualization, H.Z. and P.Z.; Investigation, H.Z. and G.S.; Methodology, P.Z. and W.L.; Project administration, P.Z. and Z.W.; Validation, G.S., W.L. and Z.L.; Writing—original draft, H.Z. and P.Z.; Writing—review and editing, W.L. and Z.W.. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Fundamental Research Funds for the Central Universities (no. 328202206), the Beijing Natural Science Foundation (no. L192002), and in part by the "Advanced and sophisticated" discipline construction project of universities in Beijing (no. 20210013Z0401), the China National Key R&D Program (no. 2020YF-B1808000).

Data Availability Statement: The data used in the simulation section of the paper is generated by sample generation for the scenario under consideration in this paper. Readers who need it can contact the corresponding author of this paper.

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

Abbreviations

The following abbreviations are used in this manuscript:

IoT	Internet of Things
DRL	deep reinforcement learning
UAV	unmanned aerial vehicle
NOMA	non-orthogonal multiple access
DQN	deep Q-Network
SNR	signal-to-noise ratio
MDP	signal-to-interference-noise ratio
ResNet	residual network
MDP	Markov decision process
DNN	deep neural network
FL	federal learning
QoS	quality of service
UE	user equipment

References

1. You, X.; Wang, C.X.; Huang, J.; Gao, X.; Zhang, Z.; Wang, M.; Huang, Y.; Zhang, C.; Jiang, Y.; Wang, Y.; et al. Towards 6G wireless communication networks: Vision, enabling technologies, and new paradigm shifts. *Sci. China Inf. Sci.* **2021**, *64*, 1–74. [CrossRef]
2. Zhang, Z.; Xiao, Y.; Xiao, M.; Ding, Z.; Lei, X.; Karagiannidis, G.K.; Fan, P. 6G Wireless Networks: Vision, Requirements, Architecture, and Key Technologies. *IEEE Veh. Technol. Mag.* **2019**, *14*, 28–41. [CrossRef]
3. Tran-Dang, H.; Krommenacker, N.; Charpentier, P.; Kim, D.-S. Toward the Internet of Things for Physical Internet: Perspectives and Challenges. *IEEE Internet Things J.* **2020**, *7*, 4711–4736. [CrossRef]

4. Nguyen, D.C.; Seneviratne, A.; Li, J.; Niyato, D.; Dobre, O.; Poor, H.V. 6G Internet of Things: A Comprehensive Survey. *IEEE Internet Things J.* **2022**, *9*, 359–383. [CrossRef]
5. Alsharif, M.H.; Jahid, A.; Kelechi, A.H.; Kannadasan, R. Green IoT: A Review and Future Research Directions. *Symmetry* **2023**, *15*, 757. [CrossRef]
6. Cui, T.; Yang, R.; Fang, C.; Yu, S. Deep Reinforcement Learning-Based Resource Allocation for Content Distribution in IoT-Edge-Cloud Computing Environments. *Symmetry* **2023**, *15*, 217. [CrossRef]
7. Kanellopoulos, D.; Sharma, V.K. Dynamic Load Balancing Techniques in the IoT: A Review. *Symmetry* **2022**, *14*, 2554. [CrossRef]
8. Abbas, G.; Mehmood, A.; Carsten, M.; Epiphaniou, G.; Lloret, J. Safety, Security and Privacy in Machine Learning Based Internet of Things. *J. Sens. Actuator Netw.* **2022**, *11*, 38. [CrossRef]
9. Wu, T.-Y.; Guo, X.; Chen, Y.-C.; Kumari, S.; Chen, C.-M. SGXAP: SGX-Based Authentication Protocol in IoV-Enabled Fog Computing. *Symmetry* **2022**, *14*, 1393. [CrossRef]
10. Alomari, A.; Subramaniam, S.K.; Samian, N.; Latip, R.; Zukarnain, Z. Resource Management in SDN-Based Cloud and SDN-Based Fog Computing: Taxonomy Study. *Symmetry* **2021**, *13*, 734. [CrossRef]
11. Bani-Bakr, A.; Hindia, M.N.; Dimyati, K.; Hanafi, E.; Tengku Mohmed Noor Izam, T.F. Multi-Objective Caching Optimization for Wireless Backhauled Fog Radio Access Network. *Symmetry* **2021**, *13*, 708. [CrossRef]
12. Aazam, M.; Islam, S.U.; Lone, S.T.; Abbas, A. Cloud of Things (CoT): Cloud-Fog-IoT Task Offloading for Sustainable Internet of Things. *IEEE Trans. Sustain. Comput.* **2022**, *7*, 87–98. [CrossRef]
13. Martinez, I.; Hafid, A.S.; Jarray, A. Design, Resource Management, and Evaluation of Fog Computing Systems: A Survey. *IEEE Internet Things J.* **2021**, *8*, 2494–2516. [CrossRef]
14. Tange, K.; De Donno, M.; Fafoutis, X.; Dragoni, N. A Systematic Survey of Industrial Internet of Things Security: Requirements and Fog Computing Opportunities. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 2489–2520. [CrossRef]
15. Hassija, V.; Chamola, V.; Saxena, V.; Jain, D.; Goyal, P.; Sikdar, B. A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures. *IEEE Access* **2019**, *7*, 82721–82743. [CrossRef]
16. Puthal, D.; Mohanty, S.P.; Bhavake, S.A.; Morgan, G.; Ranjan, R. Fog Computing Security Challenges and Future Directions [Energy and Security]. *IEEE Consum. Electron. Mag.* **2019**, *8*, 92–96. [CrossRef]
17. Wazid, M.; Das, A.K.; Shetty, S.; Rodrigues J.J.P.C.; Guizani, M. AISCM-FH: AI-Enabled Secure Communication Mechanism in Fog Computing-Based Healthcare. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 319–334. [CrossRef]
18. Feng, W.;Zhang, N.; Lin, S.; Li, S.; Wang, Z.; Ai, B.; Zhong, Z. Energy-Efficient Collaborative Offloading in NOMA-Enabled Fog Computing for Internet of Things. *IEEE Internet Things J.* **2022**, *9*, 13794–13807. [CrossRef]
19. Huang, X.; Yang, X.; Chen, Q.; Zhang, J. Task Offloading Optimization for UAV-Assisted Fog-Enabled Internet of Things Networks. *IEEE Internet Things J.* **2022**, *9*, 1082–1094. [CrossRef]
20. Jia, B.; Hu, H.; Zeng, Y.; Xu, T.; Yang, Y. Double-matching resource allocation strategy in fog computing networks based on cost efficiency. *J. Commun. Netw.* 2018, *20*, 237–246. [CrossRef]
21. Tran-Dang, H.; Bhardwaj, S.; Rahim, T.; Musaddiq, A.; Kim, D.-S. Reinforcement learning based resource management for fog computing environment: Literature review, challenges, and open issues. *J. Commun. Netw.*, **2022**, *24*, 83–98. [CrossRef]
22. Fei, Z.; Wang, Y.; Zhao, J.; Wang, X.; Jiao, L. Joint Computational and Wireless Resource Allocation in Multicell Collaborative Fog Computing Networks. *IEEE Trans. Wirel. Commun.* **2022**, *21*, 9155–9169. [CrossRef]
23. Junejo, A.K.; Komninos, N.; McCann, J.A. A Secure Integrated Framework for Fog-Assisted Internet-of-Things Systems. *IEEE Internet Things J.* **2021**, *8*, 6840–6852. [CrossRef]
24. Khashan, O.A. Hybrid Lightweight Proxy Re-Encryption Scheme for Secure Fog-to-Things Environment. *IEEE Access* **2020**, *8*, 66878–66887. [CrossRef]
25. Deb, P.K.; Mukherjee, A.; Misra, S. CEaaS: Constrained Encryption as a Service in Fog-Enabled IoT. *IEEE Internet Things J.* **2022**, *9*, 19803–19810. [CrossRef]
26. Oh, J.; Lee, J.; Kim, M.; Park, Y.; Park, K.; Noh, S. A Secure Data Sharing Based on Key Aggregate Searchable Encryption in Fog-Enabled IoT Environment. *IEEE Trans. Netw. Sci. Eng.* **2022**, *9*, 4468–4481. [CrossRef]
27. Wu, D.; Ansari, N. A Cooperative Computing Strategy for Blockchain-Secured Fog Computing. *IEEE Internet Things J.* **2020**, *7*, 6603–6609. [CrossRef]
28. Ren, J.; Li, J.; Liu, H.; Qin, T. Task offloading strategy with emergency handling and blockchain security in SDN-empowered and fog-assisted healthcare IoT. *Tsinghua Sci. Technol.* **2022**, *27*, 760–776. [CrossRef]
29. Liu, Y.; Dong, Y.; Wang, H.; Jiang, H.; Xu, Q. Distributed Fog Computing and Federated-Learning-Enabled Secure Aggregation for IoT Devices. *IEEE Internet Things J.* **2022**, *9*, 21025–21037. [CrossRef]
30. Krichen, M.; Lahami, M.; Cheikhrouhou, O.; Alroobaea, R.; Maâlej, A.J. Security Testing of Internet of Things for Smart City Applications: A Formal Approach. In *Smart Infrastructure and Applications*; EAI/Springer Innovations in Communication and Computing; Mehmood, R., See, S., Katib, I., Chlamtac, I., Eds.; Springer: Cham, Switzerland, 2020. [CrossRef]
31. Keerthi, K.; Roy, I.; Hazra, A.; Rebeiro, C. Formal Verification for Security in IoT Devices. In *Security and Fault Tolerance in Internet of Things*; Internet of Things; Chakraborty, R., Mathew, J., Vasilakos, A., Eds.; Springer: Cham, Switzerland, 2019. [CrossRef]
32. Krichen, M. Anomalies Detection Through Smartphone Sensors: A Review. *IEEE Sens. J.* **2021**, *21*, 7207–7217. [CrossRef]
33. Stojanović, B.; Božić, J. Robust Financial Fraud Alerting System Based in the Cloud Environment. *Sensors* **2022**, *22*, 9461. [CrossRef] [PubMed]

34. Zuo, P.; Sun, G.; Li, Z.; Guo, C.; Li, S.; Wei, Z. Towards Secure Transmission in Fog Internet of Things Using Intelligent Resource Allocation. *IEEE Sens. J.* **2023**, *to be published*.
35. Sutton R.S.; Barto A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 2018.
36. Zhao, X.; Yang, R.; Zhang, Y.; Yan, M.; Yue, L. Deep Reinforcement Learning for Intelligent Dual-UAV Reconnaissance Mission Planning. *Electronics* **2022**, *11*, 2031. [CrossRef]
37. Ud Din, A.F.; Mir, I.; Gul, F.; Mir, S.; Saeed, N.; Althobaiti, T.; Abbas, S.M.; Abualigah, L. Deep Reinforcement Learning for Integrated Non-Linear Control of Autonomous UAVs. *Processes* **2022**, *10*, 1307. [CrossRef]
38. Zhan, G.; Zhang, X.; Li, Z.; Xu, L.; Zhou, D.; Yang, Z. Multiple-UAV Reinforcement Learning Algorithm Based on Improved PPO in Ray Framework. *Drones* **2022**, *6*, 166. [CrossRef]
39. Zuo, P.; Wang, C.; Wei, Z.; Li, Z.; Zhao, H.; Jiang, H. Deep Reinforcement Learning Based Load Balancing Routing for LEO Satellite Network. In Proceedings of the 2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring), Helsinki, Finland, 19–22 June 2022; pp. 1–6.
40. Yu, Y.; Wang, T.; Liew, S.C. Deep-Reinforcement Learning Multiple Access for Heterogeneous Wireless Networks. *IEEE J. Sel. Areas Commun.* **2019**, *37*, 1277–1290. [CrossRef]
41. Lange, S.; Riedmiller, M. Deep auto-encoder neural networks in reinforcement learning. In Proceedings of the The 2010 International Joint Conference on Neural Networks (IJCNN), Barcelona, Spain, 18–23 July 2010; pp. 1–8.
42. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [CrossRef]
43. Keras: Deep Learning for Humans. Available online: https://keras.io/ (accessed on 26 February 2023).
44. Ziegler, J.L.; Arn, R.T.; Chambers, W. Modulation recognition with GNU radio, keras, and HackRF. In Proceedings of the 2017 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN), Baltimore, MD, USA, 6–9 March 2017; pp. 1–3. [CrossRef]