Cognitive Balance for Fog Computing Resource in Internet of Things: An Edge Learning Approach

# Cognitive Balance for Fog Computing Resource in Internet of Things: An Edge Learning Approach

Siyi Liao, *Student Member, IEEE,* Jun Wu, *Member, IEEE,* Shahid Mumtaz, *Member, IEEE,* Jianhua Li, Rosario Morello, *Member, IEEE,* and Mohsen Guizani, *Fellow, IEEE*

**Abstract**—Currently, the highly dynamic fog computing resource requirements introduced by the diverse services of the Internet of Things (IoT) result in an imbalance between computing resource providers and consumers. However, current computing resource scheduling schemes cannot cognize the dynamic resources available and do not possess decision-making or management capabilities, which leads to inefficient use of computing resources and a decreased quality of service (QoS). Balancing computing resources cognitively at the IoT edge remains unresolved. In this paper, a cognition-centric fog computing resource balancing (CFCRB) scheme is proposed for edge intelligence-enabled IoT. First, we propose a cognitive balance architecture with a cognition plane, which includes service demand monitoring, policy processing and knowledge storage of cognitive fog resources. Second, we propose the fog functions structure with sensing, interaction and learning functionalities, realizing the knowledge-based proactive discovery and dynamic orchestration of resource sharing nodes. Finally, a distributed edge learning algorithm is proposed to construct knowledge of the balance between computing resource helpers and requesters in cognitive fogs, which is further proved with mathematics. The simulation results indicate the efficiency of the proposed scheme.

**Index Terms**—Resource management, learning systems, distributed computing, cognitive science.

## 1 Introduction

As a promising technology, the IoT enables virtual or physical network objects to be globally interconnected and creates new applications, opportunities, and service modes [1]. We are expected to enjoy services from more than 50 billion IoT devices by 2020 [2]. The remarkable development of the IoT has witnessed an explosive increase in the number of IoT edge devices. The heterogeneity and complexity of traditional cloud-enabled IoT architecture has led to an increase in network delay, unsatisfying resource utilization and a low overall efficiency of the IoT [3] [4]. Therefore, the basic model of the IoT in smart cities is evolving from a centralized, cloud-enabled system to a more distributed fog computing-enabled IoT (F-IoT) [5]. F-IoT has the capability to continuously obtain information from a wide range of sensors and devices, including data collection, efficient transmission, and data preprocessing [6] [7]. In the near future, billions of devices are expected to connect to F-IoT. This development brings unprecedented load to the network while introducing new opportunities to make full use of the edge resources of F-IoT. The basic idea of fog computing has been applied in various fields since it was proposed, including IoT [8] [9]. F-IoT services require highly efficient computation and communication, which leads to a bottleneck in terms of edge resource management because of the increasing device access, complex applications and multiple services [10].

In an F-IoT system, the massive quantities of data and tasks produced by edge devices are managed by local fog nodes instead of being transmitted to a distant cloud server [11] [12]. This means that fog nodes will have to cope with changing computing resource requirements. Due to the high dynamics of computing tasks brought about by the complex services of the F-IoT, edge nodes often fall into a situation of tight computing resources. The lower efficiency and higher time delays caused by the contradiction between the relatively fixed computing resources and the dynamic of tasks in F-IoT require further attention. However, edge devices in the IoT are not always in a busy state, which means that their computing resource utilization is not maximized [13]. The computing resources of massive IoT devices need to be shared in a more effective way. Scheduling and integrating idle resources in the fog for nodes or tasks with tight resources can effectively improve the quality of service (QoS) and increase the utilization of computing resources. Therefore, determining how to balance changing needs with fixed computing resources of IoT edge devices remains an open issue. In addition, recent artificial intelligence (AI) applications in IoT are attracting more attention from both academia and industry. Edge learning in IoT has been regarded as a vast and notable area [14]. AI algorithms have become an important methodology in supporting large-

*This work was supported in part by the National Natural Science Foundation of China under Grant 61972255.(Corresponding author: Jun Wu.)*
*Siyi Liao, Jun Wu, and Jianhua Li are with Shanghai Key Laboratory of Integrated Administration Technologies for Information Security, School of Cyber Security, Shanghai Jiao Tong University, Shanghai 200240, China.(e-mail: junwuhn@sjtu.edu.cn)*
*Shahid Mumtaz is with Instituto de Telecomunicaes, Universidade de Aveiro Campus Universitrio de santiago, 3810-193 Aveiro, Portugal. ARIES Research center Universidad antonio de nebrija C/Pirineos, 55, E-28040, Madrid Spain.*
*Rosario Morello is with the Department of Information Engineering, Infrastructure and Sustainable Energy, University Mediterranea of Reggio Calabria, 89122 Reggio Calabria, Italy.*
*Mohsen Guizani is the Department of Computer Science and Engineering. Qatar University, Qatar. (e-mail: mguizani@ieee.org).*

scale IoT applications [2]. However, many existing mechanisms are insufficient to cognize the highly dynamic edge computing resource requirements and unevenly distributed computing resources.

To cope with these challenges, we propose a cognition-centric fog computing resource balancing (CFCRB) scheme. The core concept of cognitive functionalities is based on the brain-empowered system operating paradigm [15] [16]. A system with cognitive functionalities can quickly perceive the available dynamic resources and possesses certain decision-making and management capabilities. Based on this concept, we introduce sensing, interaction and learning capabilities in the CFCEB scheme. These three capabilities are fully reflected in the proposed architecture, resource node discovery algorithm, and distributed reinforcement learning algorithm. Moreover, we have introduced a low-cost reinforcement learning method using multiagent consistency theory to efficiently schedule resources and implement cognitive balancing.

The contributions of our work are summarized as follows.

- We proposed a novel cognition-centric edge computing resource balancing scheme to balance changing computing resource needs with idle resources. We have detailed the cognition of the IoT in the layered fog node architecture, including cognition-centric sensing, interacting and learning.
- A cognitive balance architecture is proposed for edge intelligence-enabled IoT. Based on the layered traditional fog computing architecture, the cognition plane is introduced. The advantages of the various proposed functions are demonstrated.
- We formulated a model to describe the cognitive resources balancing in IoT. A knowledge-based proactive discovery mode of fog resource node is detailed. Considering the resource limitations of the IoT edge, we proposed a low-cost edge learning algorithm for the proposed CFCRB scheme based on Q-learning and consistency theory.

The remainder of this paper is organized as follows. The related work is given in Section 2, and the strengths of the proposed scheme are described. Our system model and architecture of the proposed scheme are presented in Section 3. Both the basic implementation of the scheme and proposed algorithm for cognitive resource balancing are provided in Section 4. The simulation results are shown in Section 5 to estimate the performance of the scheme. Final conclusions are drawn in Section 6.

## 2 Related Work

As a promising technology, the IoT can globally connect and allow communication among virtual and physical objects [1]. Better, faster and safer service modes have been studied in recent studies, including IoT resource management, AI-enabled IoT, etc. [17]- [23]. As an emerging and promising technology, fog computing acts as an extension of the traditional cloud. Therefore, the resource utilization and data processing capability of F-IoT have been greatly improved [24]- [25]. Data analyzing, managing, and processing services are being deployed at the edge of the IoT, leading the IoT to evolve from a centralized cloud-enabled IoT to a distributed system [26]- [27]. The authors of [28] focus on joint user association and IoT resource allocation. A double-matching strategy is studied for the fog resource allocation based on cost efficiency [29]. In [13], an advanced resource management scheme for IoT is proposed by allowing resource overbooking.

Moreover, efforts to introduce artificial intelligence into IoT has drawn increasing attention. The development of IoT end devices and related platforms provides prerequisites for edge learning tasks [30]. The authors in [31] enable deep learning (DL) for IoT in the environment of edge computing. To optimize the performance of IoT deep learning applications with edge computing, a novel computation offloading strategy is designed. To better the performance of software-defined network (SDN) technology in mobile edge computing (MEC), a big data-based deep reinforcement learning (DRL) approach is proposed in [32]. However, few methods have fully considered the limited resources of the IoT edge as well as the dynamic need for computing resource and uncertain utility. Therefore, further cognition in the method is needed to improve F-IoT.

The basic idea of cognition comes from cognitive radio, that is, the maximum use of limited spectrum resources [33]. The mainstream cognitive research mainly focuses on smarter applications and collaborative management [34] [35]. The authors of [36] proposed a distributed Q-learning-based online optimization algorithm to alleviate resource requirements in the smart grid. Compared with those of the existing research, the advantages of our work are mainly reflected in the following aspects. First, the IoT is enabled with edge intelligence because of the cognition introduced into the fog computing architecture. Second, three main aspects of cognition and cognition-centric resource sharing are defined in detail. Third, distributed Q-learning is further improved and collaborates with the node discovery algorithm to adapt to the IoT edge.

## 3 Basic Architecture

### 3.1 An Overview of the Basic Scene

Fig. 1 shows the cognitive balance of fog computing resources in smart cities. IoT devices are widely distributed in smart cities and collaborate to perform various functions. Due to the characteristics of different IoT devices and service modes, the cognition of the fog nodes enables the efficient use of computing resources. Under the scheduling of the fog node, edge devices can self-organize into virtual regions for cognitive balance of computing resources. When a computing task generated by a device requires far more resources than its own processing capability, the device will send a request message to the fog server to request idle computing resources from nearby devices. A device that accepts a request for a computing resource in its vicinity contributes its idle computing resources to the resource pool of the requester. Due to the variety and dynamic generation of computing tasks, the cognition of the network edge enables edge resources to be better self-organized. As a result, a

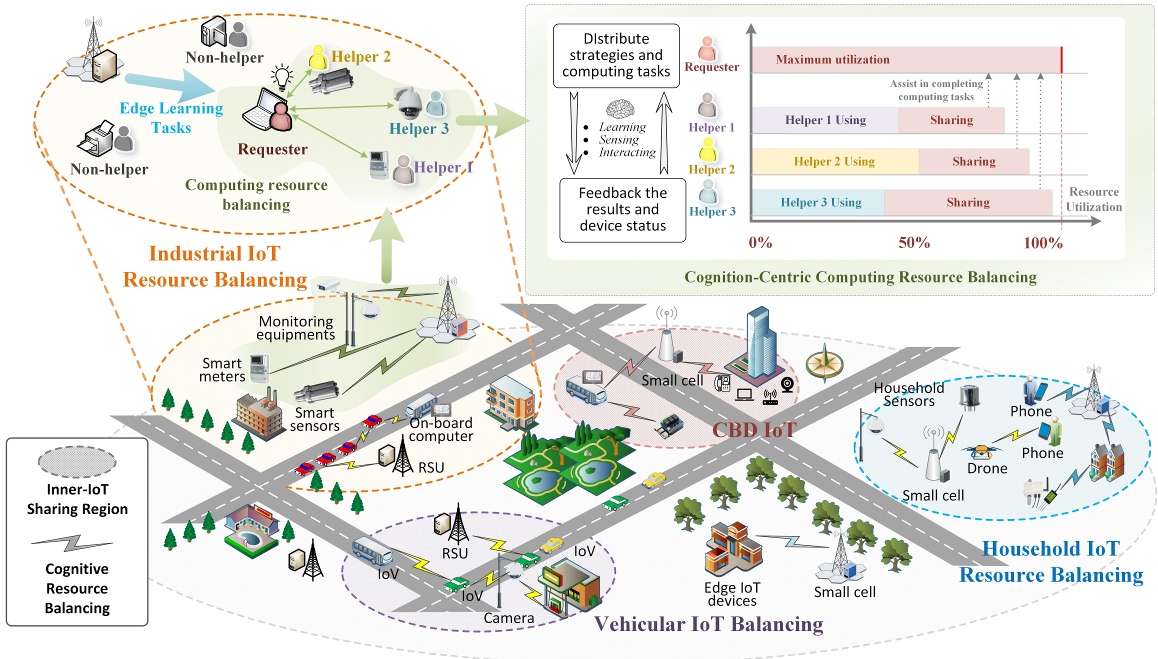



Fig. 1: The Basic Scene of Resource Cognitive Balance in Internet of Things.

vast number of IoT edge devices are able to realize dynamic clustering and share their resources cognitively.

### 3.2 Key Aspects of the Proposed Cognition-centric Fog Computing Architecture

The basic idea of the CFCRB scheme is to enable the fog nodes to have the ability to sense, interact, and learn, as well as to interact with the surrounding environment to perceive and utilize the available resources. The three key aspects of the cognition of the proposed CFCRB scheme are introduced as follows.

**Sensing:**The sensing capability of the CFCRB scheme lies in the perception of the underlying resources by the fog nodes. With this enhancement, a large quantity of edge sensors and sensing devices will be deployed at the edge of the IoT, monitoring the indicators and parameters of the edge objects. At the same time, the sensing ability

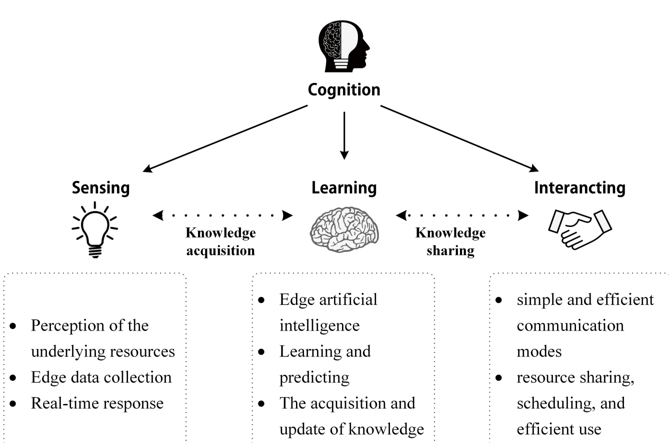



Fig. 2: Key Aspects of the Proposed Cognition-centric Fog Computing Architecture.

of the IoT edge provides data acquisition for fog-enabled artificial intelligence, gathering a real-time data report for upper-level fog nodes. Therefore, the CFCRB scheme will be helpful in perceiving key indicators for the cognitive fog nodes (CFNs) and will quickly discover potential resource nodes.

**Interacting:** IoT devices require low-cost and efficient communication modes to ensure efficient collaboration of the whole IoT system. The interaction ability of the CFCRB scheme in IoT lies in timely information exchange and task coordination. Several resource nodes can cooperate to complete the computational tasks of requesters to alleviate temporary resource constraints. An efficient interaction mode is designed in the CFCRB scheme for resource sharing, scheduling, and efficient use.

**Learning:** The learning ability at the IoT edge is the core of the CFCRB scheme in IoT. Various edge AI platforms can be embedded into smart devices to complete customized services. Through the deployment of edge learning algorithms and the obtained device status, the proposed CFCRB scheme can quickly find a cooperation strategy for resource nodes and respond to the demand for computing resources to reduce the delay of task processing. Massive data generated by the IoT edge provide natural conditions for edge learning. In our architecture, a distributed reinforcement learning method for finding the optimal strategy through interaction with the environment is proposed.

### 3.3 Cognition-centric Fog Computing Architecture

Fig. 3 shows the proposed CFCRB architecture based on the traditional fog computing paradigm [37]. Along with the availability for massive data processing provided by the cloud, the IoT cloud acts as the centralized control and service center for the fog nodes. The information provided by various IoT devices can be collected, analyzed, and

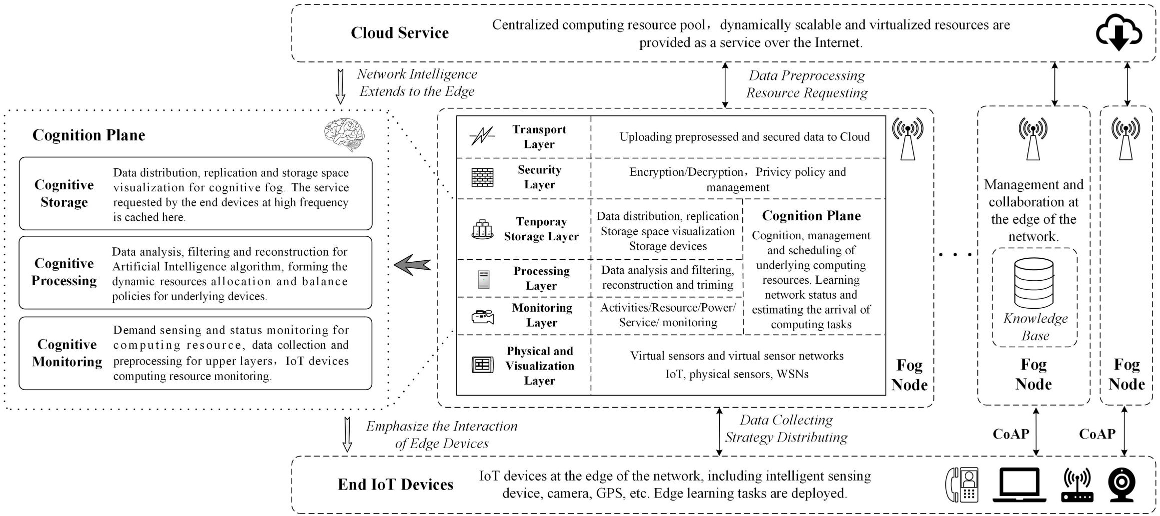



Fig. 3: The Architecture of Cognition-centric Fog Computing.

processed at the IoT edge by the cognitive fog node to optimize task scheduling, dynamic parameter adjustment, real-time status reporting, etc. Due to powerful computing and storage capabilities, the cloud collects the preprocessed data of the fog node and distributes the global computing resource sharing strategy.

The IoT devices are geographically dispersed and numerous. The computing resources of IoT edge devices are relatively weak compared to those of cloud servers. However, through device collaboration and resource balancing, the fog computing resource pool affords the possibility for edge learning. The widely distributed fog nodes in F-IoT can provide quality and timely services at the edge of the IoT. As the managers of their surrounding devices, the fog node is deployed between the underlying device and the cloud, and plays a role in edge device management, data collection and strategy distribution. Fog node monitoring, processing, and storage functions enable the IoT devices to share cognitive resources. The main functional modules of the cognitive plane are as follows.

**Cognitive Monitoring:** The main function of the cognitive monitoring layer is to monitor the real-time resource status of the IoT devices. This layer perceives the need for computing resources and the state of devices. Data are collected for upper layer preprocessing and learning.

**Cognitive Processing:** The specific edge AI algorithm is deployed at this layer, after obtaining resource demand data and device availability status. The data are analyzed, filtered, and reconstructed to form a dynamic resource allocation strategy for the underlying device.

**Cognitive Storage:** According to the results obtained by the algorithm, the final resource balancing strategy is formed here. This layer implements policy distribution of the fog, replication and storage virtualization. The models of high-frequency service requests of the IoT devices are cached here.

### 3.4 Proposed IoT Resource Balancing Model

To ensure the utilization of computing resources and the completion quality of incoming tasks, fog nodes tend to provide IoT services locally and ask for assistance from nearby helpers, rather than work with fewer remote helpers. As shown in Fig. 4, the balancing of computing resources can be divided into the following stages.

**Stage 1: Resource Searching:** The computing resources of edge devices first process their own tasks and estimate the computing resources needed to meet the latency requirements. The resource requester sends a request message to the fog server first. The fog node proactively discovers potential resource nodes based on the information in the existing knowledge base. The devices with the idle computing resource choose whether to accept the request. If the node is found to be busy, the request is rejected. Otherwise, the resource-available information is returned.

**Stage 2: Resource Balancing:** Next, the fog node feeds back helper information for all computing resources to the computing resource requester. The requester and the helper establish a D2D connection to realize resource balancing. During this process, the requester can issue a calculation task to the helpers and wait for the calculation results. In this architecture, the edge learning algorithm is deployed at the edge of the IoT to form an optimal resource sharing strategy of fog computing resources.

**Stage 3: Resource Releasing:** Finally, when the computing resource requester no longer needs the support of the helpers, or the agreed service time ends, the helpers' computing resources are released. Each device then reports the current status to the fog server. The new knowledge formed by this service is integrated into the knowledge base. Due to the variety of computing tasks and network environments, the service between helpers and requesters is not fixed. The helpers only share their resource for a period of time in the near future.

If the helper's computing resources are subsequently busy, the helper will no longer immediately serve the requester. Sometimes, the helper itself may become a computing resource requester to issue a resource sharing request. In addition, resource-stressed nonhelpers will report to the server after their resources become available, and they may be transformed from nonhelpers to helpers. Correspondingly, if a nonhelper's computing resources cannot be met for a

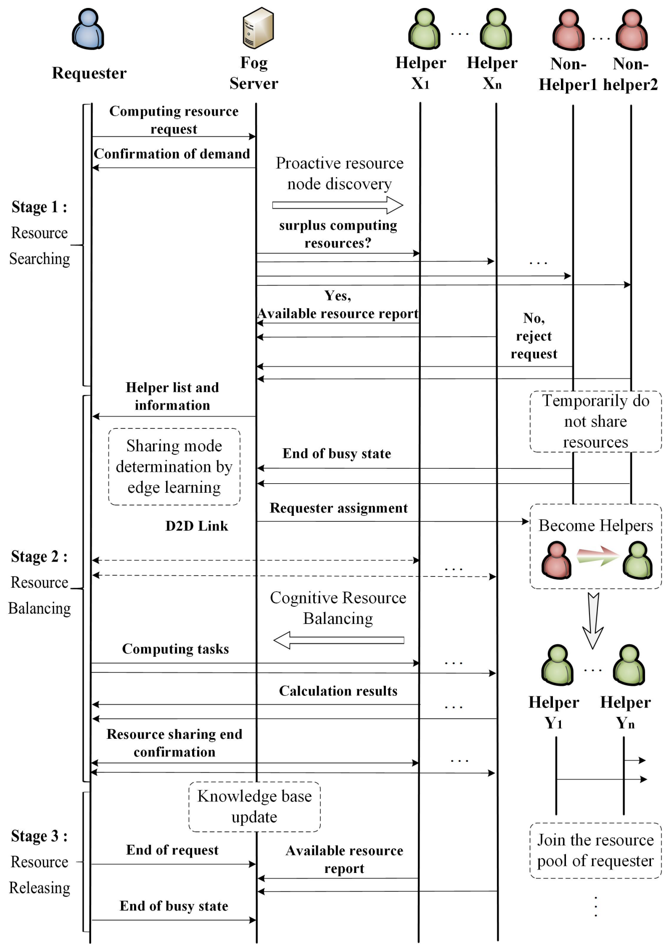



Fig. 4: The system model of the proposed CFCRB scheme.

long time, it may also become a resource requester.

## 4 Implementation of Cognition-Centric Fog Computing Resource Balancing

Due to the dynamic network environment and heterogeneity of fog nodes, IoT edge devices are required to dynamically and flexibly allocate computing resources for different incoming computing tasks. In this section, we fully consider the cognitive resource sharing scenario at the edge of the IoT, and a mathematical model is established to describe the resource sharing problem.

### 4.1 Knowledge-based Proactive Discovery of Helpers

In the CFCRB scheme in this paper, to improve the ability of fog nodes to sense idle resources, we have developed a knowledge-based, proactive discovery algorithm for the CFCRB scheme. Its main purpose is to increase the speed at which IoT devices can be used for resource sharing in order to quickly respond to the needs of resource requesters. If a resource requesting node or a fog node randomly requests helpers one by one, a large amount of communication resources will be wasted and task processing will be delayed. Therefore, we fully use the knowledge base formed by the cognitive fog nodes for efficient sensing of the potential resources.

TABLE 1: Main symbols and explanations

| *Symbols* | *Explains* |
|---|---|
| $T$ | Time of cognitive resource balancing |
| $N$ | Number of nodes for balancing |
| $R_{i,t}$ | Occupied computing resources |
| $U_i$ | Comprehensive utility function |
| $D$ | Total resource requirements |
| $\gamma$ | discount factor, $\gamma \in (0, 1]$ |
| $\mathcal{D}, \mathcal{R}$ | Collection of all possible needs and resource |
| $\mathcal{I}_c$ | Computing resource balancing strategy |
| $\tilde{D}_{i,t}$ | Estimated average requirement |
| $\epsilon_i$ | The step of unit $i$ |
| $\alpha$ | Local estimate of the demand error |
| $\hat{R}_{i,t}$ | The resource output estimated by the $i-th$ other unit |
| $\tilde{J}(\tilde{D}_{i,t}, \hat{R}_{i,t})$ | Local action value function |
| $\pi_i(\tilde{D}_{i,t}, \hat{R}'_{i,t})$ | Local operation strategy |
| $\bar{c}_{i,t}$ | The sum of the utility of all the units |
| $V(T, \gamma, R_{\mathcal{I},[1:T]})$ | The total balance utility |
| $a_i, b_i, c_i, e_i, f_i$ | Parameters for utility function |
| $\mu_i$ | The utility function adjustment factor |
| $\eta_{i,t}$ | The resource utilization |

The current resource idle state of a node cannot be a sufficient reason to become a better helper. Therefore, this algorithm first makes a request to the knowledge base to determine if a node is suitable as a helper based on various parameters, including communication delay, available resources in a previous period of time, communication cost, cost of using the resource, etc. For all helpers participating in cognitive resource balancing, we construct the parameter matrix $X$ according to the following table. However, the values in the above parameter matrix are the absolute values of resources, not relative values used for resource evaluation. Therefore, we normalize the matrix.

$$A_{m\times n} = (a_{ij})_{m\times n} \quad a_{ij} = \begin{cases} \text{H: } x_{ij}/\sqrt{\sum_{i=1}^{n} x_{ij}^2} \\ \text{L: } x'_{ij}/\sqrt{\sum_{i=1}^{n} {x_{ij}'}^2} \end{cases} \tag{1}$$

TABLE 2: Building matrix $X_{m\times n}$ based on the current resources

| $Index$ | $e_1$ | $e_2$ | $\cdots$ | $\cdots$ | $e_n$ |
|---|---|---|---|---|---|
| $Helper\ 1$ | $x_{11}$ | $x_{12}$ | $\cdots$ | $\cdots$ | $x_{1n}$ |
| $Helper\ 2$ | $x_{21}$ | $x_{22}$ | $\cdots$ | $\cdots$ | $x_{2n}$ |
| $\cdots$ | $\cdots$ | $\cdots$ | $\cdots$ | $\cdots$ | $\cdots$ |
| $Helper\ m$ | $x_{m1}$ | $x_{m2}$ | $\cdots$ | $\cdots$ | $x_{mn}$ |

**Algorithm 1** Helper discovery and selection algorithm

**Input:** Node resource matrix $Xi_{m\times n}$ of different helpers, Corresponding weight of resources $\omega_1, \cdots, \omega_n$.
**Output:** Helpers $\{h_1^{'}, \cdots h_k^{'}\}$.

```
for i = 1, ..., X do
    Request computing resources
    Get the current resource status
    Set the value of δ_{i,j}
end for
Count the various resource parameters of each helper
for i = 1, ..., H do
    Initialization of matrix Xi_{m×n}
    Calculate Z*_m using Eq.(1) to Eq.(4)
    Sort Z*_m
end for
Set time window j in Eq.(5)
Calculate the probability of helpers using Eq.(5)
Select helpers in probabilistic order according to resource requirements
```

For the above matrix, we take the best solution and the worst solution in its columns.

$$\begin{cases} A^{+} = (max(a_{i1}), max(a_{i1}), \cdots, max(a_{in})) \\ A^{-} = (min(a_{i1}), min(a_{i1}), \cdots, min(a_{in})) \end{cases} \tag{2}$$

In this way, we can calculate the distance between the various parameters of each helper as well as the optimal solution and the worst solution as follows.

$$\begin{cases} S_i^{+} = \sqrt{\sum_{i=1}^{n} \omega_i (a_{ij} - a_j^{+})^2} \\ S_i^{-} = \sqrt{\sum_{i=1}^{n} \omega_i (a_{ij} - a_j^{-})^2} \end{cases} \tag{3}$$

Therefore, based on the above distance, we can calculate the weight of each helper.

$$Z_i^{*} = \frac{s_i^{-}}{s_i^{+} + s_i^{-}} \tag{4}$$

The above $Z$ represents the relative situation between various aspects of both the helper's conditions and other helpers participating in resource balancing. A larger value indicates that the comprehensive situation of each aspect is closer to the optimal solution; otherwise, it is closer to the worst solution. Then, we calculate the cumulative value of the resource statistics of all nodes in the past $j$ times and use the value of the cumulative $Z$ value as the probability value of the node being elected as a helper.

$$P_{nodex} = \frac{\sum_i Z_i}{\sum_x \sum_i Z_i \cdot \delta_{i,j}} \qquad \begin{cases} \delta_{i,j} = 0 & x_i \in \mathbb{H} \\ \delta_{i,j} = 1 & x_i \notin \mathbb{H} \end{cases} \tag{5}$$

If node $x$ is a helper in the $j-th$ window, then the corresponding $\delta_{i,j} = 1$; otherwise, $\delta_{i,j} = 0$. In this way, we can select $N$ units for sharing computing resources.

### 4.2 System Model of Resource Balancing

The basic problem we are studying is that multiple units in the IoT share their redundant resources for collaborative processing of computing tasks. It is assumed that there are a total of $N$ units in a region that can provide the surplus computing resources needed to collectively meet the total computing resource demand $D$. The main purpose of computing resource sharing is to find an economical scheduling method for a period of time $T$ under the condition of meeting various restrictions. The economical scheduling utility can be expressed as

$$\begin{aligned} U_{total} &= \sum_{t=1}^{T} \gamma^{t-1} U_i(R_{i,t}) \\ &= \sum_{t=1}^{T} \gamma^{t-1} (Rev(\sum_{i=1}^{N} R_{i,t}) - \sum_{i=1}^{N} Cst_i(R_{i,t})) \end{aligned} \tag{6}$$

The above expression represents the sum of all the costs of units and revenue of the requester during time $T$, where $R_{i,t}$ represents the computational resource output of the unit $i$ at time $t$. $\gamma$ is the discount factor, and $U_i(\cdot)$ is the utility function. $R_{i,t}$ represents the computing resources that unit $i$ has used at time $t$, and $U_i(R_{i,t})$ indicates the total utility at that moment. In particular, if the computing resource balancing issue is considered for a limited time and regardless of the influence of time factors, which are $T < \infty$ and $\gamma = 1$, then the objective function can also be written as $\sum_{t=1}^{T} Rev(\sum_{i=1}^{N} R_{i,t}) - \sum_{i=1}^{N} Cst_i(R_{i,t})$. If we consider the issue again indefinitely, where $T = \infty$, then we set usually $\gamma \in (0, 1)$. To make our CFCRB model more suitable for the IoT edge, and to provide a more in-depth study of the cognitive resource balancing issue, we made the following reasonable assumptions:

**(1)** Feasibility: Assume that $\mathcal{R}_i$ is the limited discrete set of feasible computing resource outputs of IoT device $i$ and meets the condition that $0 \in \mathcal{R}_i$. For any moment $t$, there is a computing resource balancing strategy for devices $\mathcal{I}_c$,

$$\sum_{i \in \mathcal{I}_c} R_{i.t} = D_t, R_{i,t} \in \mathcal{R}_i \setminus \{0\} \tag{7}$$

**(2)** Optimality: In the case where the feasibility conditions are established, meet $R_{i,t} \in R_i$ as the optimal solution $R_{i,t}^{*}$, and meet the condition $P_{i,t}^{*} \in \mathcal{R}_i$ for any moment $t$.

**(3)**Topology: Assume that the topology between agents is strongly connected and balanced.

Note that the current literature often assumes that the utility function of each unit is known. This assumption has some shortcomings when only incomplete information of the utility function is known. Based on existing research, the utility function of this paper is assumed to be static and nonconvex, and even its mathematical expression is unknown [38] [39]. Based on some related work, we modeled the function of income and cost in this paper as follows. In this paper, the cost function is specifically defined as $Cst_i(R_i) = a_i R_i^2 + b_i R_i + c_i + |e_i sin(f_i(\underline{R}_i - R_i))|$ based on some related research [40] [41]. The cost part of the utility function is related to each helper that provides resources. The overall cost is the sum of the cost of using each helper resource. Meanwhile, the revenue function is defined as $Rev(\sum_{i=1}^{N} R_i) = \theta ln(1 + \sum_{i=1}^{N} R_i)$. The revenue part of the utility function calculates the service satisfaction given to the resource requester by using the computing resources of the helpers, and this satisfaction is related to the total resources obtained. The more resources the resource requester obtains, the higher its satisfaction is. In general, the resource scheduling problem can usually be summarized as the problem of seeking the optimal value of a system's utility [42]- [44]. The utility function in this paper also follows the same rule.

### 4.3 Problem Formulation of Cognitive Balancing

Based on the assumptions and analysis, we can obtain the following optimization equations. For the objective function, $U_{total}$ indicates the sum of all the utilities of $N$ units in a limited period of time T. In addition, $\mu_i$ is the parameter used to change the overall level of the utility function of unit $U_i$. The discount factor $\gamma$ means that the closer the level is to the current state, the greater the impact it has on the current system. $\eta_{i,t}$ represents the resource utilization of the unit $U_i$ at the specific time $t$, that is, the proportion of used resources to the maximum available resources $R_{i,max}$.

$$\min: \; U_{total} = \sum_{t=1}^{T} \gamma^{t-1}(Rev(\sum_{i=1}^{N} R_{i,t}) - \sum_{i=1}^{N} \mu_i Cst_i(R_{i,t}))$$

$$s.t. \begin{cases} C_1: & \sum_{i=1}^{N} R_{i,t} \geq D_t, \qquad D_t \in \mathcal{D} \\ C_2: & R_{i,max} \cdot \eta_{i,t} = R_{i,t} \in \mathcal{R}_i, \\ & \quad i = 1, \cdots, N \qquad t = 1, \cdots, T \\ C_3: & \mu_i \in \mathcal{H}_i, \quad i = 1, \cdots, N \\ C_4: & \gamma \in (0, 1] \end{cases}$$

$C_1$ ensures that the computing resources shared by all units at every $t$ can meet the total computing needs $D_t$ in the region. $D_t$ is the computing resource demand in the period of time $t$. $\mathcal{D}$ is a collection of all possible computing resource needs, and $\mathcal{R}_i$ is the feasible collection of computing resource outputs of the unit $i$. Assume that $R_i = [0, \overline{R_i}]$. $C_2$ determines all the possible combinations of computing resources of the units and resource utilization, and $C_3$ determines the cardinality of the utility function. The computational cost model of each unit is relatively fixed. However, if the units own resources are tight, we increase $\mu_i$ to increase the cost level of using the node. Otherwise, we decrease $\mu_i$.

### 4.4 Distributed Q-Learning Algorithm in IoT

Suppose there are multiple decision agents responsible for the optimal operation of the production unit. Each agent $i$ has the ability to adjust the local power output $R_{i,t}$ and measure the local utility $U_i(R_{i,t})$. Unlike a centralized algorithm, where a decision agent can find the total power demand $D_t$, we assume that at time $t$, each agent can only find the local power demand $D_{i,t}$.

#### 4.4.1 *Discovery of average computing resource requirements*

We define $\tilde{D}_{i,t}$ as the estimated average resource requirement. For each agent, we apply the following average agreement:

$$\tilde{D}_{i,t}[n+1] = \tilde{D}_{i,t}[n] + \epsilon \sum_{j \in \mathcal{N}_i} a_{ij}(\tilde{D}_{j,t}[n] - \tilde{D}_{i,t}[n]) \tag{8}$$

where $\tilde{D}_{i,t}[0] = D_{i,t}$.

If there is a consistent step $\epsilon \in (0, 1/max_i l_{ii})$ and the given topology is strongly connected and balanced, then $\tilde{D}_{i,t}[n] \to \sum_{i \in \mathcal{I}} D_{i,t}/N$ when $n \to \infty$.

Each agent $i$ agrees on the average consistent step $\epsilon$ through the following iterative process:

$$\begin{cases} \epsilon_i[1] = \sum_{j \in \mathcal{N}_i} a_{ij}\epsilon_j[0], \\ \epsilon_i[2] = max_{j \in \mathcal{N}_i}\epsilon_j[1], \\ \qquad \cdots \qquad\qquad\qquad n \geq N-2, \\ \epsilon_i[n+1] = max_{j \in \mathcal{N}_i}\epsilon_j[n] \\ \epsilon = \frac{1}{\epsilon_i[n+1]} \end{cases} \tag{9}$$

where $\epsilon_i$ is the local step size that the agent $i$ will negotiate and the initial $\epsilon_i[0] = 1, \forall i \in \mathcal{I}$. We have

$$\begin{aligned} 1/\epsilon_i[n+1] = 1/max_{i \in \mathcal{I}} \sum_{j \in \mathcal{I}} a_{ij} \in (0, 1/max_i l_{ii}), \\ \forall i \in \mathcal{I}, \forall n \geq N-2 \end{aligned} \tag{10}$$

Thus, the entire system operated by the agent has the ability to improve the scalability and robustness of IoT edge devices.

#### 4.4.2 *Balance computing resource output and demand*

In a distributed algorithm, each unit can discover information about its neighbors. Let $\alpha$ be a local estimate of the resource demand error, and use an average consistent algorithm to estimate the global resource output $R_{i,t} \in [0, \overline{R}_i]$:

$$R_i[n+1] = R_i[n] + \epsilon \sum_{j \in N_i} a_{ij}(R_j[n] - R_i[n]) \tag{11}$$

where $R_i[0] = \tilde{D}_{i,t} - R_{i,t}$. In a strongly connected and balanced topology, when $n \to \infty$, we have

$$R_i[n] \to \alpha \triangleq \sum_{i=1}^{N} (\tilde{D_{i,t}} - R_{i,t})/N \tag{12}$$

If $\alpha > 0$, then

$$R_{i,t} \leftarrow R_{i,t} + sign(\alpha) min\{\alpha, \overline{R}_i - R_{i,t}\} \tag{13}$$

If $\alpha < 0$, then

$$R_{i,t} \leftarrow R_{i,t} + sign(\alpha) min\{|\alpha|, \overline{R}_i - R_{i,t}\} \tag{14}$$

until $\alpha = 0$.

#### 4.4.3 *Update local action value function*

Let $\hat{R}_{i,t}$ be the computing resource output of the other units estimated by the $i-th$ unit, $\hat{R}_{i,t} = (R_{i_1,t}, R_{i_2,t}, \cdots R_{i_{N-1},t}, R_{i_N,t})^T$.

Update the local action value function $\tilde{J}(\tilde{D}_{i,t}, \hat{R}_{i,t})$ according to the following formula:

$$\tilde{J}(\tilde{D}_{i,t}, \hat{R}_{i,t}) \leftarrow \overline{c}_{i,t} + \gamma min_{\hat{R}'_{i,t}} \tilde{J}(\tilde{D}_{i,t+1}, \hat{R}'_{i,t}) \tag{15}$$

where $\overline{c}_{i,t}$ denotes the sum of the utility for computing resource balance of all the units known by device $i$ at the time $t$, which is $\overline{c}_{i,t} = \sum_{j=1}^{N} C_j(R_{ij,t})$

**Algorithm 2** Cognition-centric edge learning algorithm for computing resource balancing based on distributed Q-learning

**Input:** Total demand $D$, computing power of units $R_i$, balancing time T, break parameter M.
**Output:** Computing resource balancing strategy.
Initialize $t = 0$,$m = 0$,$\tilde{J}(\tilde{D}_{i,t}, \hat{R}'_{i,t}) = 0, \epsilon(\tilde{D}_{i,t}) = 0$
Negotiate the average step $\epsilon$ by Eq.(15)
**repeat**
  **for** $t = 0, \cdots, T-1$ **do**
    Obtain the average computing resource demand $\tilde{D}_{i,t}$ at time $t$ by Eq.(14)
    Find feasible computing resource output distribution by Eq.(8), Eq.(9), and Eq.(10)
    Execute $\hat{R}'_{i,t} = \pi_i^{\epsilon}(\tilde{D}_{i,t}, \hat{R}'_{i,t})$
    Obtain local balance cost $\overline{c}_i(\hat{R}'_{i,t})$
    Update the local action value function $\tilde{J}(\tilde{D}_{i,t}, \hat{R}'_{i,t})$ by $\overline{c}_{i,t} + \gamma min_{\hat{R}'_{i,t}} \tilde{J}(\tilde{D}_{i,t+1}, \hat{R}'_{i,t})$
    Update the local operation strategy $\pi_i(\tilde{D}_{i,t}, \hat{R}'_{i,t})$ by $argmin_{\hat{R}'_{i,t}} \tilde{J}(\tilde{D}_{i,t}, \hat{R}'_{i,t})$
  **end**
  Gradually reduce $\epsilon(\tilde{D}_{i,t})$ step by step
  $m \leftarrow m + 1$
**repeat end** until $m \geq M$

#### 4.4.4 Update local operational strategy

Update the optimal local operation strategy $\pi_i(\tilde{D}_{i,t}, \hat{R}'_{i,t})$ according to the following formula:

$$\pi_i(\tilde{D}_{i,t}, \hat{R}'_{i,t}) \leftarrow argmin_{\hat{R}'_{i,t}} \tilde{J}(\tilde{D}_{i,t}, \hat{R}'_{i,t}) \tag{16}$$

It is worth noting that in Eq.(16), if there are multiple $\hat{R}'_{i,t}$ that will result in the smallest $\tilde{J}(\tilde{D}_{i,t}, \hat{R}'_{i,t})$, agent $i$ can arbitrarily choose an $\hat{R}'_{i,t}$.

#### 4.4.5 Balanced exploration and utilization

The agent can balance the exploration and utilization of the computing resource output combination of the IoT edge resource, that is, the strategy of using the least cost in the known strategy with the probability of $1 - \epsilon_i$, and the strategy of using the unknown cost with the probability of $\epsilon_i$. If each agent i selects an action in such a way that the probability of exploration is gradually reduced, then the agent as a whole selects the action in such a manner that the probability of the exploration is gradually reduced.

The following algorithm introduces a distributed Q-learning algorithm to solve the problem of balancing edge IoT computing resources. After the computing resource balancing ends, the computing resources of the agents will be released. Meanwhile, related parameters of this service will be sent to fog and added to the knowledge base.

### 4.5 Analysis and Proof of Cognition-centric Fog Computing Resource Balancing

This part aims to prove the rationality of the distributed edge learning algorithm of our CFCRB scheme based on the principle of Q-learning.

First, we let $V(T, \gamma, R_{\mathcal{I},[1:T]})$ denote the total balance cost $\sum_{t=1}^{T} \gamma^{t-1} \sum_{i=1}^{N} C_i(R_{i,t})$ of the computing resource balancing strategy combination $R_{\mathcal{I},[1:T]} \triangleq \{R_{\mathcal{I},t}\}_{t=1}^{T}$ that satisfies $C_1$ within condition $C_2$.

We use $V(T, \gamma, R_{\mathcal{I},[1:T]})$ and $V(+\infty, \gamma, R_{\mathcal{I},[1:+\infty]})$ to represent crew combination and scheduling problems over finite and infinite time.

**Lemma 1**: Assume that both feasibility and optimality are met. The existence of $\overline{\gamma} \in (0, 1)$ (e.g., $\forall \gamma \in (\overline{\gamma}, 1)$), the optimal solution of the computational resource combination balancing call problem with the objective function $V(T, \gamma, R_{\mathcal{I},[1:T]})$ is also the optimal solution of the objective function $V(T, 1, R_{\mathcal{I},[1:T]})$.

**Proof 1**: Since the feasibility hypothesis is satisfied, $\mathcal{R}_i$ is a finite discrete set for the device $i = 1, \cdots, N$. Therefore, the set $\{V(T, 1, R_{\mathcal{I},[1:T]})\}$ is finite. Consider the following formula:

$$lim_{\gamma \to 1} \sum_{t=1}^{T} \gamma^{t-1} \sum_{i=1}^{N} U_i(R_{i,t}) = \sum_{t=1}^{T} \sum_{i=1}^{N} U \quad _i(R_{i,t}) \tag{17}$$

There exists $\overline{\gamma} \in (0, 1)$; if $\gamma \in (\overline{\gamma}, 1)$, then for any $R^a_{\mathcal{I},[1:T]}$ and $R^b_{\mathcal{I},[1:T]}$, $V(T, 1, R^a_{\mathcal{I},[1:T]}) < V(T, 1, R^b_{\mathcal{I},[1:T]})$. Then, we have:

$$V(T, 1, R^a_{\mathcal{I},[1:T]}) < V(T, \gamma, R^b_{\mathcal{I},[1:T]}) < V(T, 1, R^b_{\mathcal{I},[1:T]})) \tag{18}$$

$$V(T, \gamma, R^a_{\mathcal{I},[1:T]}) < V(T, 1, R^a_{\mathcal{I},[1:T]}) \tag{19}$$

Inequality (13) is derived from $V(T, \gamma, R_{\mathcal{I},t})$ with a monotonically increasing $\gamma$.

We use $\{R^s_{\mathcal{I},[1:T]}\}$ to represent the optimal solution of the computational resource balancing strategy with the objective function $V(T, 1, R_{\mathcal{I},[1:T]})$ that

$$V(T, 1, R^s_{\mathcal{I},[1:T]}) < V(T, 1, R_{\mathcal{I},[1:T]}), \forall R_{\mathcal{I},[1:T]} \notin \{R^s_{\mathcal{I},[1:T]}\} \tag{20}$$

Then, for $\gamma \in (\overline{\gamma}, 1)$, the following formula holds.

$$V(T, \gamma, R^s_{\mathcal{I},[1:T]}) < V(T, \gamma, R_{\mathcal{I},[1:T]}), \forall R_{\mathcal{I},[1:T]} \notin \{R^s_{\mathcal{I},[1:T]}\} \tag{21}$$

We use $D^f_t$ and $D^i_t$ to represent the computational resource balancing strategy at time t, respectively, in finite and infinite cases.

**Lemma 2**: Assume that both feasibility and optimality are met. If the simulator is reset to $t = kT, k = 1, 2, \cdots$, the computational force demand $D^i_t$ has time $T$, $D^i_t = D^f_t$ at $t = 1, \cdots, T$, and the objective function $V(+\infty, \gamma, R_{\mathcal{I},[1,+\infty)})$. When the computational resource balancing problem is at $t = (k-1)T + 1, \cdots, kT$, the optimal solutions of $t = (k-1)T + 1, \cdots, kT$ and $V(T, \gamma, R_{\mathcal{I},[1:T]})$ are the same.

**Proof 2**: We consider infinite horizontal computing resource balancing and scheduling goals under a static strategy

$$V(+\infty,\gamma,R_{\mathcal{I},[1,+\infty)}) = \lim_{T'\to\infty}\sum_{t=1}^{T\prime}\gamma^{t-1}\sum_{i=1}^{N}U_i(R_{i,t})$$
$$= V(T,\gamma,R_{\mathcal{I},[1:T]}) + \gamma^T V(+\infty,\gamma,R_{\mathcal{I},[T+1,+\infty)}) \tag{22}$$

The simulator is reset to $t = kT \quad (k = 1,2,\cdots)$, there is

$$V(+\infty,\gamma,R_{\mathcal{I},[1:+\infty)}) = \frac{1}{1-\gamma^T}V(T,\gamma,R_{\mathcal{I},[1:T]}) \tag{23}$$

We use $\{R^*_{\mathcal{I},[1:T]}\}$ to represent the optimal solution for the objective function to be $V(T,\gamma,R_{\mathcal{I},[1:T]})$. If $R_{\mathcal{I},[1:T]} \in \{R^*_{\mathcal{I},[1:T]}\}$, then according to Eq.(16), $R_{\mathcal{I},[(k-1)T+1:kT]} = R_{\mathcal{I},[1:T]}, k = 1,2,\cdots$ is also the optimal solution of the objective function $V(+\infty,\gamma,R_{\mathcal{I},[1:+\infty)})$.

By using Eq.(17), it is very simple to prove that the optimal solution for the objective function $V(+\infty,\gamma,R_{\mathcal{I},[1,+\infty)})$ is also the optimal solution for the objective function $V(T,\gamma,R_{\mathcal{I},[1:T]})$ at $t = (k-1)T+1,\cdots,kT$.

**Theorem 1**: Consider an infinite level of computing resource balancing and scheduling issues. Reset the problem formulation for a limited level of computing resource balancing and scheduling over time $t = mT, m = 1,2,\cdots$. The $J(D_{[1:T]},R_{\mathcal{I},[1:T]})$ of traditional Q-learning converges to the minimum value $J^*(D_{[1:T]},P_{\mathcal{I},[1:T]})$ with probability 1 when $m \to +\infty$.

The following inference is obtained by Lemma 1-2 and Theorem 1.

**Inference 1**: Consider a limited water computing resource balancing and scheduling problem that is constructed by resetting the time to $t = mT, m = 1,2,\cdots$. Assume that both feasibility and optimality are met. There is a $\overline{\gamma} \in (0,1)$ for any $\gamma \in (\overline{\gamma},1)$, and the Q-learning algorithm finds at least a $R^*_{\mathcal{I},[1:T]}$ such that $V(T,1,P_{\mathcal{I},[1:T]})$ is the smallest.

**Proof 3**: According to Theorem 1, $J(D_{[1:T]},R_{\mathcal{I},[1:T]})$ converges to $J^*(D_{[1:T]},R_{\mathcal{I},[1:T]})$ when $m \to +\infty$. By using $\pi(D_{[1:T]}) \leftarrow argminJ(D_{[1:T]},R_{\mathcal{I},[1:T]})$, Q-learning can find an optimal progressive solution. If $\gamma \in (\overline{\gamma},1)$, this solution also minimizes $V(T,1,P_{\mathcal{I},[1:T]})$ according to Lemma 1 and Lemma 2. Since distributed Q-learning can be considered as a distributed practice of traditional Q-learning, the following inference exists.

**Inference 2**: Consider a limited computing resource balancing and scheduling problem that is constructed by resetting the time to $t = mT, m = 1,2,\cdots$. Assume feasibility, optimality and topological satisfaction. The distributed Q-learning algorithm can also prove to be feasible in Theorem 1 and Inference 1. The above proof fully illustrates the rationality and feasibility of the proposed algorithm.

## 5 Simulation and Discussion

The efficiencies and reasonable explanations of the CFCRB scheme are given in this section by a series of comparisons on convergence time, task processing, computing resource utilization, etc.

Our CFCRB scheme is simulated on a server with an Intel i7 6700 CPU and 16 GB RAM. We simulate the entire system using different numbers of nodes, and each fog node manages a certain number of resource nodes. Some of the devices managed by each fog node are IoT devices with idle resources. A unit within the area generates computing resource requirements for a period of time. After the helper discovery process, the unit becomes the decision unit (DU), and all helpers became assisting units (AUs). Each AU is operated by the DU. The directed graphs used to represent the topology between agents are strongly connected. In this case, we assume that one unit requests three units for cognitive sharing to balance the computing resources.

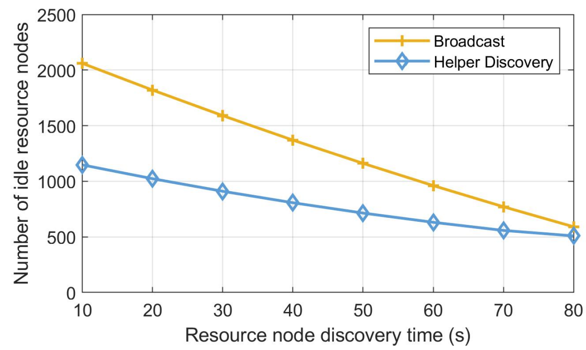



Fig. 5: Comparison of node discovery algorithms.

First, to verify the effectiveness of our proposed node discovery algorithm, we simulated the process of discovering computing resource nodes. We set up different numbers of resource nodes for the system. The criterion for measuring the effectiveness of an algorithm is the time required to find a given total amount of computing resources. It can be seen from Fig. 5 that our proposed node discovery algorithm is significantly faster than the broadcast and query method.

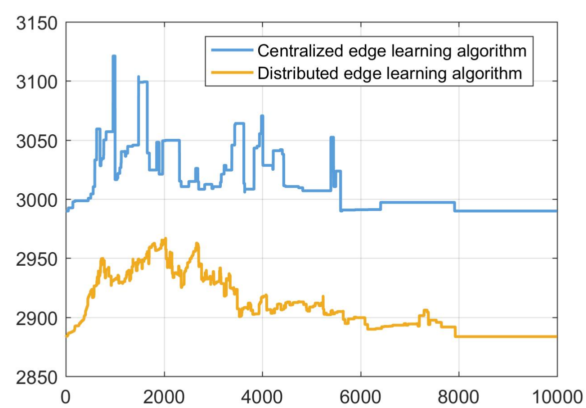
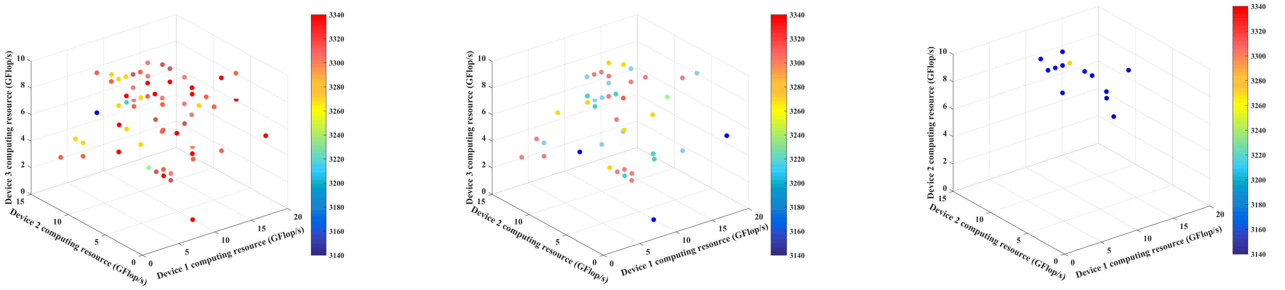


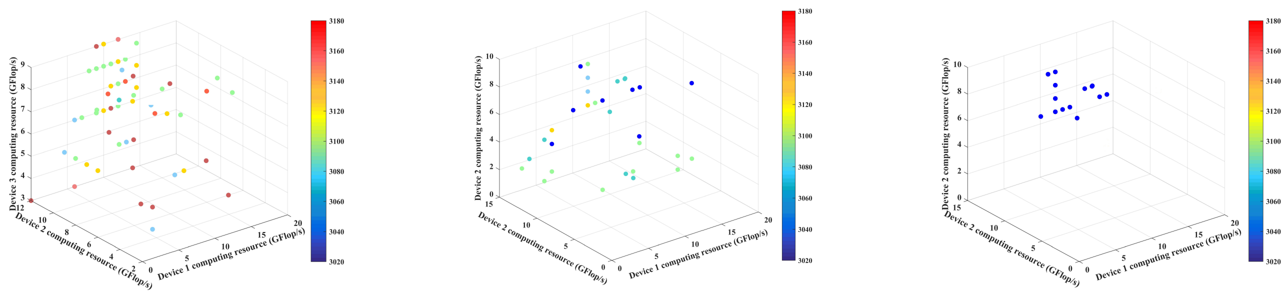

Fig. 6: Comparison of algorithm convergence results.

Then, we tested the difference between traditional centralized Q-learning algorithms and our distributed Q-learning algorithm under the same conditions (task requirements, unit available resources, number of units, utility function, etc.).

As shown in Fig. 6, the distributed edge learning algorithm that we designed has a lower overall utility level. The utility-iteration curve of the traditional centralized learning algorithm fluctuates more during the iterative process, and the overall utility is higher. In addition, the advantages of distributed computing cause our edge learning algorithm




(a) Traditional Q-Learning Stage 1 (b) Traditional Q-Learning Stage 2 (c) Traditional Q-Learning Stage 3




(d) Cognitive Edge Learning Stage 1 (e) Cognitive Edge Learning Stage 2 (f) Cognitive Edge Learning Stage 3

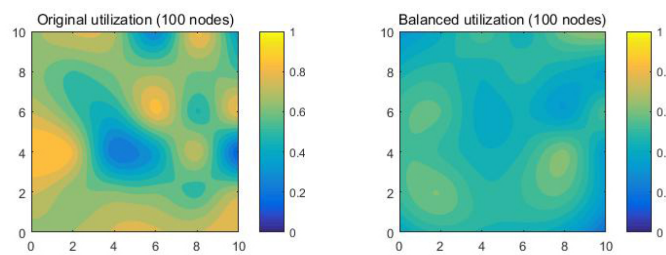
Fig. 7: Comparison of the convergence rates and convergence results of the algorithms

to converge faster, which means it can obtain a cognitive resource balancing solution faster than the traditional method.

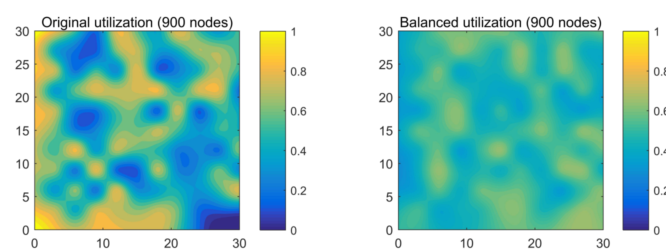
Since the utility-iteration curve can only reflect some of the macro parameters of the algorithm, we further analyze and visualize the process of the algorithms. We divided the centralized and distributed edge learning algorithms into three phases.

As shown in Fig. 7, the three coordinate axes of the 3-D image represent the calculated resource values of each of the three AUs. The scatter points therein represent the corresponding utility of each iteration result. The three pictures on the first line of Fig. 7(a) (b) (c) show the three stages of traditional centralized Q-learning, with the iterations of 1-4000, 4000-8000 and 8000-12000, respectively. As the algorithm runs, we can see that the scatter becomes more concentrated and its distribution area becomes smaller. This observation indicates that the result of the algorithm, the pattern of resource balancing, is converging. The three pictures on the second line of Fig. 7(d) (e) (f) represent the three stages of distributed learning; they represent the utility function values of the iterations of 1-4000, 4000-8000 and 8000-12000, respectively. Unlike the traditional algorithm, we can clearly see that scatter is more concentrated and is concentrated more quickly. At the same time, the dots in the second line are darker; that is, the overall cost utility is smaller. This result shows that the distributed edge learning algorithm has better performance and can converge on better solutions faster. The above two sets of pictures illustrate that the proposed algorithm can obtain better resource balancing methods and faster speeds than traditional algorithms.

To validate the superiority of the algorithm, we also examine whether the computing resources at the IoT edge are fully utilized under the scheme that we proposed. Therefore, we simulated the performance of our CFCRB scheme for the IoT edge scenario.



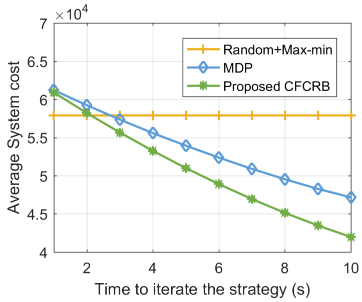

(a) Comparison of resource utilization of 100 nodes



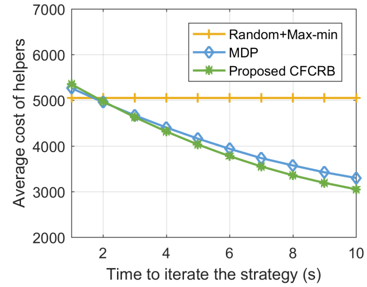

(b) Comparison of resource utilization of 900 nodes

Fig. 8: Utilization in the case of a large number of nodes

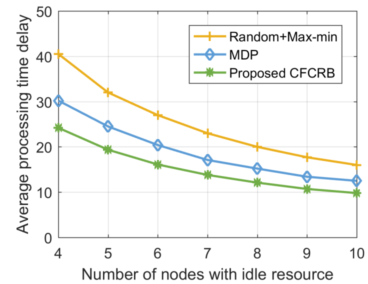
To demonstrate computing resource utilization, we use color brightness to express the utilization of computing resources in Fig. 8. The brighter the color is, the busier the device is, and the higher the computing resource utilization. Conversely, a darker area indicates a lower computational resource utilization. As shown in Fig. 8(a) and Fig. 8(b), we simulated the utilization of computing resources with 100 nodes and 900 nodes, respectively. We find that the original utilization of computing resources of the device does not appear to be regular, and the color difference is also




(a) Average system cost utility




(b) Average helper cost utility




(c) Average time delay

Fig. 9: Efficiency comparison of CFCRB with other methods

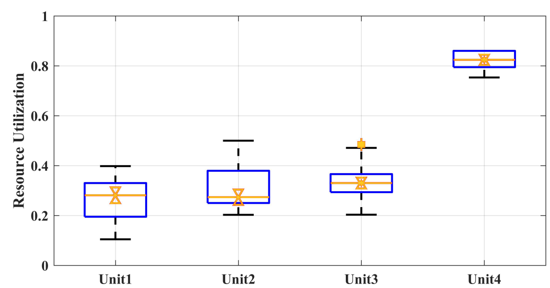
large. We maintain the same computing task parameters and use the cognitive resource balancing scheme proposed in this paper. It can be found that the color of the balanced utilization figures becomes more moderate and that the color difference between the areas becomes smaller. This result indicates that the fog computing resources of the IoT are more efficiently utilized through the proposed distributed edge learning algorithm and the cognition-centric computing resource balancing scheme. Additionally, the effectiveness of the methods proposed in this paper has been verified under different numbers of nodes.

To verify the effectiveness of our proposed CFCRB mechanism, we simulated the computational cost of the system, the computational cost of the nodes, and the computational delay. The methods compared include 1) using maximum and minimum fairness after randomly selecting nodes 2) the Markov decision process (MDP), and 3) the node discovery mechanism and distributed Q-learning algorithm proposed in this paper. Fig. 9 (a) shows the computational cost of the system over time. We can see that over time, our proposed CFCRB mechanism can obtain lower computational costs faster. For the max-min fairness method, although it iterates quickly and its cost is low at the beginning, its results are inferior to those of the other two algorithms over time. Fig. 9(b) shows the average cost of helpers for the above three methods. Our proposed CFCRB mechanism also performs optimally, showing that we can achieve acceptable computing time while making better use of IoT resources at the edge. Fig. 9(c) presents the performance of the above three methods when considering computational delay. We performed experiments separately with different numbers of available resource nodes. From the figure, we can see that our proposed algorithm can complete the computing task faster. Especially when there are fewer resource nodes, the advantage of the CFCRB mechanism is even more obvious.

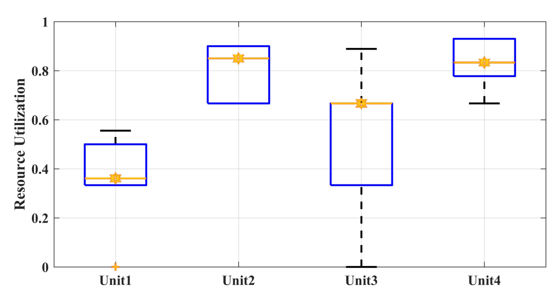
TABLE 3: Complexity comparison of the algorithms

| *Algorithms* | *Time* | *Space* |
|---|---|---|
| $Distuributed - QL$ | O($KH$) | O($SAH$) |
| $QL$ | O($KH$) | O($SAH$) |
| $MDP$ | O($S^2H$) | O($S^2AH$) |
| $Max - Min$ | O($N$) | O($N$) |

At the same time, we also compared the complexity of the algorithm proposed in this paper with the algorithms listed above. The specific complexity comparison results are shown in the following table. In this table, $K$ represents the number of episodes, $H$ represents the number of steps in an episode, $S$ represents the number of states and $A$ represents the number of actions. From the perspective of complexity, the complexity of distributed Q-learning is the same as that of traditional Q-learning, and their complexity is lower than that of MDP. From the perspective of time and space complexity, the complexity of maximum and minimum fairness is relatively low, but the result of Fig. 9 also fully shows that its performance is limited.




(a) Resource utilization with non-resource balancing




(b) Resource utilization under the proposed scheme

Fig. 10: Statistical distribution of unit resource utilization

In addition, to measure the performance of the proposed algorithm over a period of time, we collected the utilization of computing resources over time. The resource utilization of the four units whose resources needed to balanced is collected, and a box plot is used to show the overall resource utilization. A box plot can be used to show a set of discrete data. The horizontal lines above and below the box diagram indicate the range of valid data. The upper and lower limits of the block represent the two quartiles of the chosen data. The horizontal line in the box represents the median of the data. Therefore, a box plot can well reflect the overall level and approximate range of the computing resource utilization.

Among the four units, unit 4 has a larger demand for computing resources. As a resource requester, it requested three helpers to share computing resources. It is not difficult to see from the Fig. 10(a) that if the resource balancing scheme is not adopted, unit 4 will have a higher resource utilization rate for a long time. The other three units adjacent to it have more idle resources, and their resources can be shared with unit 4. Fig. 10(b) shows the full use of idle computing resources. Through the cognitive balancing of computing resources, the computing resources of the idle units are used by unit 4, and the box plot shows that their computing resource utilization has been significantly improved.

## 6 Conclusion

In this paper, we focused on making fog computing resources cognitively balanced in the Internet of Things. The cognition-centric fog computing resource balancing scheme is proposed to further expand the intelligence of the network to the edge of the IoT. A cognitive IoT edge computing architecture that includes the sensing, interacting and learning capabilities of fog computing is designed. This paper proposes a distributed edge learning optimization algorithm based on Q-learning and an efficient resource node discovery method. Distributed algorithms are implemented by integrating multiagent consistency theory. These distributed algorithms involve only local computation and communication and have the potential to increase the robustness and scalability of the IoT.

We still expect that the CFCRB scheme that we proposed can be implemented through more efficient protocols. For future work, we plan to exploit the corresponding protocol (e.g., IEEE 21451) to perform efficient collaboration and resource management scheduling of edge IoT on hardware platforms.

## References

[1] H. F. Atlam, R. J. Walters, G. B. Wills "Software-defined Networks with Mobile Edge Computing and Caching for Smart Cities: A Big Data Deep Reinforcement Learning Approach," *2018 3rd Cloudification of the Internet of Things (CIoT)*, pp. 31-37, 2018.

[2] J. Siryani, B. Tanju, T. J. Eveleigh, "A Machine Learning Decision-Support System Improves the Internet of Things Smart Meter Operations," *IEEE Internet of Things Journal*, vol. 4, no. 4, pp. 1056-1066, 2017.

[3] G. Li, J. Wu, J. Li, K. Wang and T. Ye, "Service Popularity-based Smart Resources Partitioning for Fog Computing-enabled Industrial Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4702-4711, 2018.

[4] L.D. Xu, W. He, S. Li, "Internet of Things in Industries: A Survey," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2233-2243, 2014.

[5] H. Shah-Mansouri, V. W. S. Wong "Hierarchical Fog-Cloud Computing for IoT Systems: A Computation Offloading Game," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 1-1, 2018.

[6] S. Mumtaz, A. Alsohaily, Z. Pang et al., "Massive Internet of Things for Industrial Applications," *IEEE Industrial Electronics Magazine*, vol. 11, pp. 28-33, 2017.

[7] N. Javaid, A. Sher, H. Nasir, N. Guizani, "Intelligence in IoT-based 5G Networks: Opportunities and Challenges", *IEEE Communications Magazine*, vol. 56 no. 10, pp. 94-100, 2018.

[8] S. Liao, J. Li, J. Wu, W. Yang, Z. Guan, "Fog-Enabled Vehicle as a Service for Computing Geographical Migration in Smart Cities," *IEEE Access*, vol. 7, pp. 8726-8736, 2018.

[9] Y. Xiao and M. Krunz, "Dynamic Network Slicing for Scalable Fog Computing Systems with Energy Harvesting," *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 36, no. 12, pp. 2640 - 2654, 2018.

[10] A. Kliem, O. Kao, "The Internet of Things Resource Management Challenge," *IEEE International Conference on Data Science and Data Intensive Systems*, 2015.

[11] T. X. Tran, A. Hajisami, P. Pandey, D. Pompili, "Collaborative Mobile Edge Computing in 5G Networks: New Paradigms, Scenarios, and Challenges," *IEEE Communications Magazine*, vol. 55, no. 4, pp. 54-61, 2017.

[12] Y. Xiao and M. Krunz, "Distributed Optimization for Energy-efficient Fog Computing in the Tactile Internet," *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 36, no. 11, pp. 2390 - 2400, 2018.

[13] F. Caglar, S. Shekhar, A. Gokhale, X. Koutsoukos, "Intelligent, Performance Interference-Aware Resource Management for IoT Cloud Backends," *IEEE First International Conference on Internet-of-Things Design and Implementation*, 2016.

[14] Y. Meidan, M. Bohadana, A. Shabtai, J. D. Guarnizo, M. Ochoa, N. O. Tippenhauer, "ProfilIoT: A Machine Learning Approach for IoT Device Identification Based on Network Traffic Analysis," *ACM Symposium on Applied Computing* pp. 506-509, 2017.

[15] A. Ahmad, S. Ahmad, M. H. Rehmani and N. U. Hassan, "A Survey on Radio Resource Allocation in Cognitive Radio Sensor Networks," in IEEE Communications Surveys & Tutorials, vol. 17, no. 2, pp. 888-917, Secondquarter 2015.

[16] S. Haykin, "Cognitive radio: brain-empowered wireless communications," in IEEE Journal on Selected Areas in Communications, vol. 23, no. 2, pp. 201-220, Feb. 2005.

[17] S. E. Collier, "The Emerging Enernet: Convergence of the Smart Grid with the Internet of Things," *IEEE Industry Applications Magazine*, vol. 23, no. 2, pp. 12-16, 2016.

[18] J. Kim, Y. Jeon, H. Kim, "The Intelligent IoT Common Service Platform Architecture and Service Implementation," *Journal of Supercomputing*, vol. 74, no. 9, pp. 42424260, 2018.

[19] R. Morello, C. De Capua, G. Fulco and S. C. Mukhopadhyay, "A Smart Power Meter to Monitor Energy Flow in Smart Grids: The Role of Advanced Sensing and IoT in the Electric Grid of the Future," *IEEE Sensors Journal*, vol. 17, no. 23, pp. 7828-7837, 2017.

[20] Y. Xiao, M. Krunz and T. Shu, "Multi-Operator Network Sharing for Massive IoT," *IEEE Communications Magazine*, vol. 57, no. 4, pp. 96-101, 2019.

[21] M. S. Mushtaq, A. Shahid and S. Fowler, "QoS-Aware LTE Downlink Scheduler for VoIP with Power Saving," *2012 IEEE 15th International Conference on Computational Science and Engineering*, Nicosia, 2012, pp. 243-250, doi: 10.1109/ICCSE.2012.41.

[22] S. Randel et al., "Adaptive MIMO signal processing for mode-division multiplexing," *OFC/NFOEC*, Los Angeles, CA, 2012, pp. 1-3.

[23] M. I. Ashraf, M. Bennis, C. Perfecto and W. Saad, "Dynamic Proximity-Aware Resource Allocation in Vehicle-to-Vehicle (V2V) Communications," *2016 IEEE Globecom Workshops (GC Wkshps)*, Washington, DC, 2016, pp. 1-6, doi: 10.1109/GLOCOMW.2016.7848885.

[24] G. Li, J. Wu, J. Li, Z. Guan and L. Guo, "Fog Computing-Enabled Secure Demand Response for Internet of Energy Against Collusion Attacks Using Consensus and ACE," *IEEE Access*, vol. 6, no. 1, pp. 11278-11288, 2018.

[25] J. Wu, M. Dong, K. Ota, J. Li and Z. Guan, "FCSS: Fog-Computing-based Content-Aware Filtering for Security Services in Information-Centric Social Networks," *IEEE Transactions on Emerging Topics in Computing*, DOI: 10.1109/TETC.2017.2747158, pp. 1-12, 2017.

[26] H. Li, K. Ota, M. Dong, "Learning IoT in edge: Deep Learning for the Internet of Things with Edge Computing," *IEEE Network*, vol. 32, no. 1, pp. 96-101, 2018.

[27] R. Morabito, V. Cozzolino, A. Y. Ding, N. Beijar, O. Jrg, "Consolidate IoT Edge Computing with Iightweight Virtualization," *IEEE Network*, vol. 32 , no. 1 , pp. 102-111, 2018.

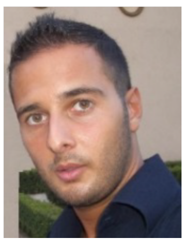
[28] S. F. Abedin, Md. G. R. Alam, S. M. A. Kazmi, N. H. Tran, D. Niyato, C. S. Hong, "Resource Allocation for Ultra-Reliable and Enhanced Mobile Broadband IoT Applications in Fog Network," *IEEE Transactions on Communications*, vol. 67, no. 1, pp. 489-502, 2019.
[29] B. Jia, H. Hu, Y. Zeng, T. Xu, Y. Yang, "Double-matching resource allocation strategy in fog computing networks based on cost efficiency," *Journal of Communications and Networks*, vol. 20, no. 3, pp. 237-246, 2018.
[30] D. Lee, H. M. Lee, "IoT service classification and clustering for integration of IoT service platforms," *The Journal of Supercomputing*, vol.74, no. 12, pp. 68596875, 2018.
[31] W. Kang, D. Kim, "DeepRT: A Predictable Deep Learning Inference Framework for IoT Devices," *IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI)*, 2018.
[32] Y. He, F.R. Yu, N. Zhao, V.C.M. Leung, H. Yin, "Software-defined Networks with Mobile Edge Computing and Caching for Smart Cities: A Big Data Deep Reinforcement Learning Approach," *IEEE Communications Magazine*, vol. 55, no. 37, pp. 31-37, 2017.
[33] D. M. Kalathil and R. Jain, "Spectrum Sharing through Contracts for Cognitive Radios*IEEE Transactions on Mobile Computing*, vol. 12, no. 10, pp. 1999-2011, 2013.
[34] S. S. Mwanje, H. Sanneck and A. Mitschele-Thiel, "Synchronized Cooperative Learning for Coordinating Cognitive Network Management Functions," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 2, pp. 244-256, 2018.
[35] G. Baggio, R. Bassoli and F. Granelli, "Cognitive Software-Defined Networking Using Fuzzy Cognitive Maps," *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 3, pp. 517-539, 2019.
[36] F. Li, J. Qin and W. X. Zheng, "Distributed Q-Learning-Based Online Optimization Algorithm for Unit Commitment and Dispatch in Smart Grid," *IEEE Transactions on Cybernetics*.
[37] M. Aazam and E. Huh, "Fog Computing: The Cloud-IoTIoE Middleware Paradigm," *IEEE Potentials*, vol. 35, no. 3, pp. 40-44, 2016.
[38] N. Zhao, Y. Liang, D. Niyato, Y. Pei, Y. Jiang, "Deep Reinforcement Learning for User Association and Resource Allocation in Heterogeneous Networks," *IEEE Global Communications Conference (GLOBECOM)*, 2018.
[39] X. Liu, Y. Liu, Y. Chen, "Reinforcement Learning in Multiple-UAV Networks: Deployment and Movement Design," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, 2019.
[40] T. Jianxing, S. Bin, C. Rui, Y. Yao and W. Guosong, "Economic Dispatching Method Based on Dynamic Network Loss Factor," 2018 International Conference on Power System Technology (POWERCON), Guangzhou, 2018, pp. 601-605, doi: 10.1109/POWERCON.2018.8601865.
[41] Y. Jia, Z. Y. Dong, C. Sun and K. Meng, "Cooperation-Based Distributed Economic MPC for Economic Load Dispatch and Load Frequency Control of Interconnected Power Systems," in IEEE Transactions on Power Systems, vol. 34, no. 5, pp. 3964-3966, 2019, doi: 10.1109/TPWRS.2019.2917632.
[42] S. Chen, Y. Zheng, W. Lu, V. Varadarajan and K. Wang, "Energy-Optimal Dynamic Computation Offloading for Industrial IoT in Fog Computing," in IEEE Transactions on Green Communications and Networking, vol. 4, no. 2, pp. 566-576, June 2020, doi: 10.1109/TGCN.2019.2960767.
[43] C. You and K. Huang, "Exploiting Non-Causal CPU-State Information for Energy-Efficient Mobile Cooperative Computing," in IEEE Transactions on Wireless Communications, vol. 17, no. 6, pp. 4104-4117, 2018, doi: 10.1109/TWC.2018.2820077.
[44] C. You and K. Huang, "Exploiting Non-Causal CPU-State Information for Energy-Efficient Mobile Cooperative Computing," in IEEE Transactions on Wireless Communications, vol. 17, no. 6, pp. 4104-4117, 2018, doi: 10.1109/TWC.2018.2820077.

**Siyi Liao** received the B.S. degree from School of Electronic Information Engineering in Beijing Jiao Tong University, Beijing, China, in 2017. Now, he is a candidate Ph.D. at School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China. His research interests are focusing on Multi-access Edge Computing (MEC), Blockchain, Internet of Vehicles (IoV), etc. He is a student member of IEEE.

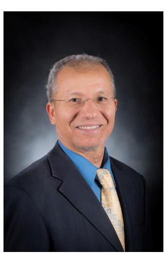
**Jun Wu** (S'08-M'12) received the Ph.D. degree in information and telecommunication studies from Waseda University, Japan, in 2011. He was a Post-Doctoral Researcher with the Research Institute for Secure Systems, National Institute of Advanced Industrial Science and Technology (AIST), Japan, from 2011 to 2012. He was a Researcher with the Global Information and Telecommunication Institute, Waseda University, Japan, from 2011 to 2013. He is currently a professor of School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, China. He is also the vice dean of Institute of Cyber Science and Technology and vice director of National Engineering Laboratory for Information Content Analysis Technology, Shanghai Jiao Tong University, China. He is the chair of IEEE P21451-1-5 Standard Working Group. He has hosted and participated in a lot of research projects including National Natural Science Foundation of China (NFSC), National 863 Plan and 973 Plan of China, Japan Society of the Promotion of Science Projects (JSPS), etc. His research interests include the advanced computing, communications and security techniques of software-defined networks (SDN), information-centric networks (ICN) smart grids, Internet of Things (IoT), 5G/6G, etc., where he has published more than 140 refereed papers. He has been the Track Chair of VTC 2019, VTC 2020 and the TPC Member of more than ten international conferences including ICC, GLOBECOM, etc. He has been a Guest Editor of the IEEE Sensors Journal, Sensors, ICT Express. He is an Associate Editor of the IEEE Access, IEEE Networking Letters.

**Shahid Mumtaz** has more than 10 years of wireless industry/academic experience and is currently working as Principal Research Scientist and Technical Manager at Instituto de Telecomunicaes (IT) Portugal. He has received his Master and PhD degrees in Electrical & Electronic Engineering from Blekinge Institute of Technology, Sweden, and University of Aveiro, Portugal in 2006 and 2011, respectively. From 2005 to 2006 for Ericsson and Huawei at Research Labs in Sweden as Researcher. Since January 2015, he has been with the Department of Electrical and Electronic Engineering at the University of Aveiro, Portugal where he is Adjunct Associate Professor and Director of the Wireless Networking and Communications Group. He uses mathematical and system level tools to model and analyze emerging wireless communication architectures, leading to innovative and/or theoretically optimal new communication techniques. He is working closely with leading R&D groups in the industry to transition these ideas to practice. His work is currently supported by Samsung, Huawei, and Ericsson, as well as FCT. He has also been providing consultancy to Intel, Huawei, and Portugal Telecom. In January 2017, IEEE has started a new standard on P1932.1: Standard for Licensed/Unlicensed Spectrum Interoperability in Wireless Mobile Networks and was elected as Vice chair for this standard. Furthermore, in January 2015, he was also elected Vice-Chair of IEEE Research Project on Vision of Green Standardization due to his pioneer work in green communication as well. He is also actively involved in 3GPP standardization on LTE release 12 onwards, along with major manufacturers (Huawei and Intel).Dr Mumtaz has published 3 books and more than 150 publications in very high-rank IEEE transactions, journals, book chapters, international conferences. He is serving as Scientific Expert and Evaluator for Research Funding Agencies, such as European Commission, and COST.

**Jianhua Li** is a professor/Ph.D. supervisor and the dean of School of Information Security Engineering, Shanghai Jiao Tong University, Shanghai, China. He got his BS, MS and Ph.D. degrees from Shanghai Jiao Tong University, in 1986, 1991 and 1998, respectively. He was the chief expert in the information security committee experts of National High Technology Research and Development Program of China (863 Program) of China. He is the member of the committee of information security area of the state 10th five-year plan of China. Also, he is a committee expert of China State Secrecy Bureau and Shanghai Secrecy Bureau. He was the leader of more than 30 state/province projects of China, and published more than 200 papers. He published 6 books and has about 20 patents. He made 3 standards and has 5 software copyrights. He got the Second Prize of National Technology Progress Award of China in 2005. He got the First Prize of National Technology Progress Award of Shanghai in 2003 and 2004, and he got two First Prize of National Technology Progress Awards of Shanghai in 2004. His research interests include cyberspace security, Next Generation Networks, etc.



**Rosario Morello** (M'03) was born in Reggio Calabria, Italy, in 1978. He received the M.Sc. Degree (cum laude) in Electronic Engineering and the Ph.D. Degree in Electrical and Automation Engineering from the University Mediterranea of Reggio Calabria, Italy, in 2002 and 2006, respectively. Since 2005, he has been Postdoctoral Researcher of Electrical and Electronic Measurements at the Department of Information Engineering, Infrastructure and Sustainable Energy of the same University. At the present he is an Assistant Professor. His main research interests include the design and characterization of distributed and intelligent measurement systems, wireless sensor network, environmental monitoring, decision-making problems and measurement uncertainty, process quality assurance, instrumentation reliability and calibration, energy, smart grids, battery testing, biomedical applications and statistical signal processing, non-invasive systems, biotechnologies and measurement, instrumentation and methodologies related to Healthcare. Dr. Morello is a member of the Italian Group of Electrical and Electronic Measurements (GMEE).



**Mohsen Guizani** (S'85-M'89-SM'99-F'09) received the B.S. (with distinction) and M.S. degrees in electrical engineering, the M.S. and Ph.D. degrees in computer engineering from Syracuse University, NY, USA, in 1984, 1986, 1987, and 1990, respectively. He is currently a Professor at the CSE Department in Qatar University, Qatar. Previously, he served in different academic and administrative positions at the University of Idaho, Western Michigan University, University of West Florida, University of Missouri-Kansas City, University of Colorado-Boulder, and Syracuse University. His research interests include wireless communications and mobile computing, computer networks, mobile cloud computing, security, and smart grid. He is currently the Editor-in-Chief of the IEEE Network Magazine, serves on the editorial boards of several international journals and the Founder and Editor-in-Chief of Wireless Communications and Mobile Computing journal (Wiley). He is the author of nine books and more than 500 publications in refereed journals and conferences. He guest edited many special issues in IEEE journals and magazines. He also served as a member, Chair, and General Chair of a number of international conferences. Throughout his career, he received three teaching awards and four research awards. He also received the 2017 IEEE Communications Society WTC Recognition Award and the 2018 AdHoc Technical Committee Recognition Award for his contribution to outstanding research in wireless communications and Ad-Hoc Sensor networks. He was the Chair of the IEEE Communications Society Wireless Technical Committee and the Chair of the TAOS Technical Committee. He served as the IEEE Computer Society Distinguished Speaker and is currently the IEEE ComSoc Distinguished Lecturer. He is a Fellow of IEEE and a Senior Member of ACM.