

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,300

Open access books available

171,000

International authors and editors

190M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



## Chapter

# Edge-Cloud Collaboration for Industrial IoT: An Online Approach

*You Shi, Yuye Yang, Changyan Yi, Bing Chen and Jun Cai*

## Abstract

In this chapter, we take the Industrial Internet of Things (IIoT) as the background for studying the energy-saving resource management framework to control the cloud center (CC), edge server (ES), and terminal equipment in a closed loop. In this framework, industrial sensors collect data and transmit it to the ES for aggregation. These data form computing tasks for data analysis. Our goal is to minimize the energy consumption of the whole system while ensuring satisfied data processing accuracy and service delay of all IIoT tasks. We formulate the ES preprocessing mode selection, sensor sampling rate adaptation, and edge cloud computing and communication resource allocation as a joint optimization problem. Due to the random arrival of data and time-varying channel conditions, we introduce an online dynamic algorithm with low complexity, which efficiently solves the problem.

**Keywords:** edge-cloud collaboration, industrial IoT, preprocessing method selection, sampling rate adaption, computing and communication resource allocation

## 1. Introduction

Due to the rapid development of 5G and Industry 4.0, IIoT-sensing devices, such as smart manufacturing, smart plants, and smart industrial services, have generated a abundant of data. For traditional cloud computing, it is considerably challenging to process such massive data efficiently. Fortunately, the edge computing can significantly reduce the cloud computing's computing load, and thus has been proposed as supplementary paradigm recently [1]. In industrial area, the edge server (ES) close to the data source can be enabled to process some computing tasks, so as to provide more effective data processing services and lead to less communication overhead.

Although some researchers have proposed edge cloud collaboration to increase the industrial systems' operational and energy efficiencies, there are still some inherent while unaddressed limitations. Particularly, computing and communication resources of the edge-cloud collaboration are relatively limited [2]. Hence, if closed-loop optimization is not considered in the management of cloud, ES, and terminal devices, edge cloud collaboration cannot fully make use of its advantages. Some relevant researchers have studied the resource allocation problem of IIoT's edge cloud collaboration [3, 4], including delay awareness, price-based service scheduling [2, 3], and energy-aware resource allocation [4, 5].

However, data collection and data analysis have some special requirements which will be affected by the complex industrial environment, which has been ignored by most studies: (i) In IIoT system, industrial equipment needs high-precision adjustment. Any small error may cause industrial equipment to make wrong behavior and cause serious troubles. [5]. Therefore, ensuring the accuracy of data processing in IIoT service is very important. This motivates the investigation and optimization of edge cloud management variables such as processing mode and sampling rate. (ii) There are commonly a variety of industrial noises in practical applications, such as electromagnetic noise [6]. Because of these, we cannot analyze the data collected by the sensor directly [7]. Hence, enabling data preprocessing at ESs is necessary (for example, data cleaning [7] and data denoising [6]) before conveying data to the cloud. This necessitates a balance of optimal resource allocations between the cloud and ESs. (iii) Since the IIoT system environment is always complex and there are random data arrival and time-varying channels, we are required to carefully manage the computing and communication resources with the guarantee of a long-term performance. Otherwise, the system will soon run out of limited CPU resources and network capacity [8]. As a result, the system efficiency will be seriously affected.

However, solving the aforementioned issues to achieve the closed-loop management is very challenging: (i) It is intuitive that the processing accuracy is increasing with the sampling rate. However, increasing the rate is equivalent to the increase of the computing load, and thus will also increase transmission delay and computing energy consumption, leading to the degradation of the system performance [5]. This implies that sampling rate must be carefully chosen for balancing different performance indicators. (ii) In practical applications, different preprocessing methods have different computing resource requirements and corresponding processing performance. In addition, data's edge preprocessing will bring extra computing delay and energy consumption. It is difficult to optimize service delay, processing accuracy, and power consumption with mutual trade-offs. (iii) In response to the random data's arrival and the time-varying channel, we need to jointly optimize and dynamically adjust the selection of preprocessing methods, sampling rate, and resource allocation. However, it is hard if not impossible to obtain random information of dynamic network in time, which is a necessary condition for long-term optimization of system performance. This will obviously lead to incomplete decision information of IIoT system. Lyapunov optimization method is often used to solve such problems. However, in IIoT applications, decision variables (such as preprocessing method and sampling rate) are often integers, and constraints like processing accuracy and service delay are sometimes nonlinear. This makes the problem much more complex than traditional ones.

## 2. Chapter contributions and organization

In this book chapter, we study an IIoT energy resource management framework. This framework is constructed on the basis of edge cloud collaboration, and aims to conduct a closed-loop management on the cloud center (CC), ESs, and terminal devices. To be more specific, in this chapter, we consider to optimize the selection of ESs' preprocessing mode, terminal devices' sampling rates, edge cloud computing and communication resource allocation for jointly to minimize the system's energy consumption. Meanwhile, we ensure service delay and accuracy of data processing in the long term. In addition, considering the random arrival of data and time-varying

channel conditions, we introduce a dynamic online algorithm with a low complexity to solve this problem.

In particular, based on the network state of the current time slot, we decompose the long-term optimization problem into a sequence of deterministic instantaneous subproblems. After that, we define a continuous probability model and take into account the future influences, and by such we use the Markov approximation algorithm to solve these subproblems to near optimal. Finally, we theoretically analyze the system performance in terms of its asymptotic upper bound.

This chapter's main contributions are listed as follows.

- For controlling the IIoT edge cloud collaboration system in a closed-loop manner, we formulate a joint optimization with computing and communication resource allocation of CC, preprocessing method selection of ESs, and sampling rate adaptation of end terminals to minimize the energy consumption of the whole system while ensuring that all applications' service delay requirements and data processing accuracy demands can be met.
- Based on Markov approximation and Lyapunov optimization, we introduce a novel online joint optimization algorithm with a polynomial time complexity.
- Theoretical analysis and simulation results evaluate the proposed algorithm's asymptotic optimality and show its advantage with the comparison of counterparts.

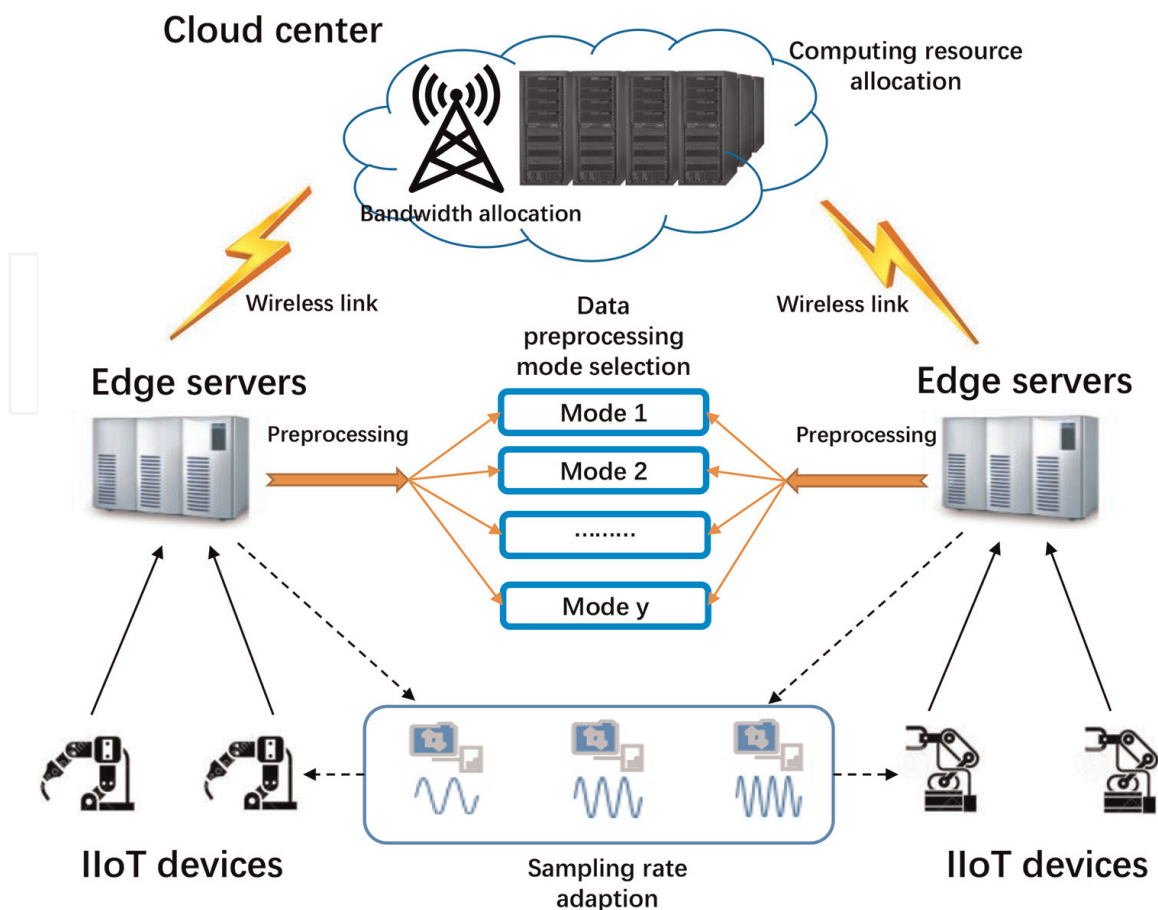
This chapter's rest contents are listed below. Section 3 models the IIoT edge-cloud collaboration's system. Section 4 formulates the corresponding joint online optimization problem for the closed-loop resource management. Section 5 introduces a novel algorithm with a low complexity based on the Markov approximation and Lyapunov optimization. Section 6 analyzes theoretical performance. Section 7 demonstrates the simulation outcomes and Section 8 concludes the chapter.

### 3. System model

An IIoT system with a remotely located CC and multiple distributed on-site ESs is considered, as shown in **Figure 1**. Each ES connects multiple IIoT sensors for a known purpose (e.g., mechanical bearings' vibration monitoring) and is used to control devices (e.g., management of its data sampling rates), preprocessing the gathered original data, and further analyzing by offloading them to CC. Denote the ESs's set as  $\mathcal{I} = \{1, 2, \dots, N\}$ , where  $|\mathcal{I}| = N$ . Denote  $S_i$  as the set of ES  $i$ 's associated devices. For example, we consider mechanical bearing vibration monitoring task [5], the vibration sensors sample the conditions of operation in the coverage area managed by ES  $i$  with specific sampling rate. After that, ES  $i$  collects vibration signals, then selects an appropriate mode<sup>1</sup> to preprocessing it. Afterwards, to conduct the computation-intensive data analysis, preprocessed data will be offloaded by ES  $i$  to CC. Clearly, this collaboration between edge clouds needs to be sustained by a good closed-loop management with three task decisions: (i) sampling rate adaptation of sensors, (ii) pre-processing mode selection for ESs, and (iii) computational and communication resources' allocation.

---

<sup>1</sup> Here, the preprocessing modes potentially denote different data cleaning or denoising methods [9].



**Figure 1.**  
An example of edge preprocessing enabled IIoT.

In addition, we investigate a time-slotted operation framework in order to portray the intrinsic IIoT systems' time-varying dynamics. Denote  $\tau$  as time index, and  $\tau \in \mathcal{T} = \{1, 2, \dots, T\}$ . We show in **Table 1** all the important symbols appeared in this chapter for easy reference.

### 3.1 Communication model

IIoT devices' sampling rates can be adjusted according to the purpose of different applications. Define  $\mathcal{K} = \{\varphi_1, \varphi_2, \dots, \varphi_K\}$  as the candidate sampling rate set, where  $\varphi_K$  (measured by Hz) indicates original sampling rate,  $K$  as the maximum sampling rate level. Calculate any level  $k$ 's sampling rate as  $\varphi_k = k\varphi_K/K$ , where  $1 \leq k \leq K$ . In addition,  $v_K$  is defined as the original data size generated in each time slot with maximum sampling rate  $\varphi_K$ .  $\theta$  indicates the time slot duration. We define  $x_{i,k}^\tau = 1$ , which denotes that in time slot  $\tau$ , ES  $i$  chooses  $k$ -th sampling rate of all its affiliated devices. Else,  $x_{i,k}^\tau = 0$ . Denote  $\lambda_{i,n}^\tau$  (measured by every second's data number) as device  $n$ 's data arrival rate under ES  $i$ 's control. Knowing all devices' sampling rate, the task size generated in time slot  $\tau$  ES  $i$  is formulated by combining collected data from every its affiliated devices in current time slot, which is expressed as

$$d_i(\tau) = \sum_{n \in S_i} \sum_{k=1}^K x_{i,k}^\tau \lambda_{i,n}^\tau \theta v_K k / K. \quad (1)$$

Symbol	Definition
$\lambda_{i,n}^\tau$	Device $n$ 's data arrival rate in slot $\tau$
$h_{i,c}^\tau$	Channel gain between ES $i$ and CC in slot $\tau$
$x_{i,k}^\tau$	ES $i$ 's sampling rate vector in slot $\tau$
$d_i(\tau)$	Task size received by ES $i$ in slot $\tau$
$b_i(\tau)$	Preprocessed task size in slot $\tau$
$W$	Bandwidth level
$p_i^{ES}$	ES $i$ 's transmit power
$\alpha_i(\tau)$	ES $i$ 's bandwidth allocation
$c_i(\tau)$	Cloud resource allocation
$E_i(\tau)$	ES $i$ 's service energy in slot $\tau$
$A_i(\tau)$	Accuracy of ES $i$ 's task in slot $\tau$
$A_i^{th}$	Long-term accuracy requirement
$H_i^{th}$	Long-term delay requirement
$V$	Lyapunov parameter

**Table 1.**

List of important notations.

Then, ES  $i$  will preprocess this task. As data denoising application for preprocessing [10] usually reduces the computational task size, we particularly focus on it here. Assuming that  $m_y$  denotes maximum transforming rate in size of the computational task using the preprocessing mode  $y \in \mathcal{Y}$ , then after preprocessing, ES  $i$ 's task size in time slot  $\tau$  is represented as

$$b_i(\tau) = d_i(\tau) \left[ \sum_{y \in \mathcal{Y}} I_{i,y}^\tau m_y \right]. \quad (2)$$

All tasks' input bits should further transmit to CC for learning and analysis. The wireless channel between CC and ESs remains constant within each time slot and varies between time slots following independent identical distribution [11]. Based on the Shannon formula, CC and ES  $i$ 's transmission rate is

$$r_i(\tau) = \begin{cases} \alpha_i(\tau) W \log_2 \left( 1 + \frac{p_i^{ES} h_{i,c}^\tau}{\alpha_i(\tau) W N_0} \right), & \alpha_i(\tau) > 0 \\ 0, & \alpha_i(\tau) = 0 \end{cases} \quad (3)$$

where  $W$  and  $N_0$  denote communication bandwidth and channel noise power spectral density, respectively.  $p_i^{ES}$  is the predefined transmission power of ES  $i$ . From ES  $i$  to CC, channel gain is defined as  $h_{i,c}^\tau$ . It includes the effects of small-scale fading, path loss, and shadowing [12]. Significantly,  $h_{i,c}^\tau$  is an environmental state uncontrollably where positive constant  $h_{max}$  has an upper bound [13]. Denote  $\alpha_i(\tau)$  as ES  $i$ 's bandwidth allocation ratio over time slot  $\tau$ . It should satisfy conditions listed below:  $0 \leq \alpha_i(\tau) \leq 1$ , and  $\sum_{i=1}^N \alpha_i(\tau) \leq 1$ . The similar definitions appear in [2, 13].

Computational task's transmission delay after preprocessing from ES  $i$  to CC is calculated as

$$F_i^{tra}(\tau) = b_i(\tau)/r_i(\tau), \quad (4)$$

and the corresponding transmission energy consumption is

$$e_i^{tra}(\tau) = p_i^{ES} F_i^{tra}(\tau). \quad (5)$$

## 3.2 Computation model

### 3.2.1 Edge preprocessing model

ES can provide several optional modes of data preprocessing. Denote  $y$  as a feasible algorithm of data denoising in a complete set  $\mathcal{Y}$ , and the corresponding CPU computation speed (cycles/s) for each data denoising algorithm is denoted by  $f_y, \forall y \in \mathcal{Y}$ .

Denote  $I_{i,y}^r \in \{0, 1\}$  as ES  $i$ 's selection index of preprocessing mode at time slot  $\tau$ .  $I_{i,y}^r = 1$  represents ES  $i$  has selected preprocessing algorithm  $y \in \mathcal{Y}$ . Otherwise,  $I_{i,y}^r = 0$ . It is obvious that we have  $\sum_{y \in \mathcal{Y}} I_{i,y}^r = 1$ . It is important to note that the computational delay for selecting mode  $y$  for data preprocessing of ES  $i$  at time slot  $\tau$  can be indicated as

$$T_i^y(\tau) = d_i(\tau)\beta_i/f_y, \quad (6)$$

where  $\beta_i$  denotes CPU cycles' number that are needed to compute one bit in ES  $i$ . Thus  $d_i(\tau)\beta_i$  is computational resource which is required by preprocessing step. After that, ES  $i$ 's preprocessing computational delay in slot  $\tau$  is indicated as

$$F_i^{pre}(\tau) = \sum_{y \in \mathcal{Y}} I_{i,y}^r T_i^y(\tau). \quad (7)$$

Therefore, ES  $i$ 's CPU computation speed (in cycles/s) is described as  $f_i(\tau) = \sum_{y \in \mathcal{Y}} I_{i,y}^r f_y$ . Hence, ES  $i$ 's preprocessing energy consumption in time slot  $\tau$  can be characterized as

$$e_i^{pre}(\tau) = z_i^{ES} f_i^3(\tau) F_i^{pre}(\tau), \quad (8)$$

where  $z_i^{ES}$  is ES  $i$ 's effective switching capacitance, which is dependent on its chip architecture [14].

### 3.2.2 Cloud computing model

Tasks received by CC are collected from multiple ESs and each ES's allocated cloud resources. Let  $c_i(\tau)$  be cloud resources' allocated proportion for ES  $i$ . The proportion should satisfy conditions list below:  $0 \leq c_i(\tau) \leq 1$ , and  $\sum_{i=1}^N c_i(\tau) \leq 1$ . After that, CC's computation delay in processing ES  $i$ 's computation tasks should be indicated as

$$F_i^c(\tau) = b_i(\tau)\varepsilon_c / (c_i(\tau)f_c), \quad (9)$$

where  $f_c$  and  $\varepsilon_c$  respectively denote CC's CPU computation speed and CPU cycles number required to compute one bit in CC. In addition, CC's energy consumption for processing ES  $i$ 's computational task is represented as

$$e_i^c(\tau) = z_c (c_i(\tau) f_c)^3 F_i^c(\tau), \quad (10)$$

where  $z_c$  is effective switched capacitance of CC [14].

### 3.3 Accuracy model for IIoT data analysis

Tasks' processing accuracy relies on two factors, ESs' data preprocessing method and IIoT devices' data sampling rate. We hypothesize that  $g(\varphi_k), \forall \varphi_k \in \mathcal{K}$  is the association between sampling rate and accuracy, and  $g_y(y), \forall y \in \mathcal{Y}$  is the association between preprocessing mode and accuracy. Moreover, learning models deployed in cloud center may have computational errors, resulting in reduced processing accuracy, which is denoted by  $h_c \leq 1$  [5].

Because of sampling rate control, cloud process and preprocessing method control are independent with each other [2]. Task processing's accuracy for ES  $i$  is

$$A_i(\tau) = g\left(\sum_{k=1}^K x_{i,k}^\tau \varphi_k\right) \cdot \left[\sum_{y \in \mathcal{Y}} I_{i,y}^\tau g_y(y)\right] h_c. \quad (11)$$

It is important that we can modify the model flexibly in practice to another form depending on various demands. The analysis framework remains valid. In addition, data-based experiments allow to obtain accuracy values regarding sampling rate and preprocessing mode [2].

## 4. Problem formulation

ESs' preprocessing mode and sensors' sampling rate can influence the computing accuracy, and the computing resources and bandwidth allocation of CC can influence the efficiency of computing and transmission. Significantly, these decisions are all closely associated. For example, if the wireless channel condition is poor or computation load of the system is large, ESs can select efficient preprocessing method and reduce the sampling rate. Hence, the CC may increase the computing and communication resources allocation ratios correspondingly. The advantage of the operations is to decrease the execution delay, transmission delay, and energy consumption. The disadvantage is that it will lose some processing accuracy. It shows that a trade-off exists in service delay, processing accuracy, and system energy consumption. For improving IIoT system's energy efficiency, we intend to minimize system energy consumption, which includes ESs' computing energy consumptions, ECs' offloading transmission energy consumption, and CC's energy consumption, with guaranteed processing accuracy and service delay. Every ES's computation task represents an application. We calculate time slot  $\tau$ 's energy consumption as

$$E_i(\tau) = e_i^{ES}(\tau) + e_i^{tra}(\tau) + e_i^c(\tau). \quad (12)$$

Because of the industrial environment's time-varying features, for minimizing long-term dynamic energy consumption, we need to manage the IIoT edge cloud



collaboration system. Therefore, the EC computing system's average energy consumption is selected as performance measurement, i.e.,  $\frac{1}{T} \sum_{\tau=1}^T \sum_{i=1}^N E_i(\tau)$ . Then we jointly optimize sampling rates of sensors, preprocessing method of ESs, and computing and communication resources of edge cloud, which denotes as

$J_i(\tau) \triangleq [x_{i,k}^\tau, I_{i,y}^\tau, \alpha_i(\tau), c_i(\tau)]$ ,  $\forall i \in \mathcal{I}$ , can be represented as

$$2\mathcal{P}_1 : \min_{\{J_i(\tau), \forall i \in \mathcal{I}\}} \frac{1}{T} \sum_{\tau=1}^T \sum_{i=1}^N E_i(\tau) \quad (13)$$

$$\text{s.t. } 0 \leq \alpha_i(\tau) \leq 1, \sum_{i=1}^N \alpha_i(\tau) \leq 1, \forall i \in \mathcal{I}, \quad (14)$$

$$0 \leq c_i(\tau) \leq 1, \sum_{i=1}^N c_i(\tau) \leq 1, \forall i \in \mathcal{I}, \quad (15)$$

$$I_{i,y}^\tau \in \{0, 1\}, \sum_{y \in \mathcal{Y}} I_{i,y}^\tau = 1, \forall i \in \mathcal{I}, \quad (16)$$

$$x_{i,k}^\tau \in \{0, 1\}, \sum_{k=1}^K x_{i,k}^\tau = 1, \forall i \in \mathcal{I}, \quad (17)$$

$$\lim_{T \rightarrow +\infty} \frac{1}{T} \sum_{\tau=1}^T H_i(\tau) \leq H_i^{th}, \forall i \in \mathcal{I}, \quad (18)$$

$$\lim_{T \rightarrow +\infty} \frac{1}{T} \sum_{\tau=1}^T A_i(\tau) \geq A_i^{th}, \forall i \in \mathcal{I}, \quad (19)$$

where (14)-(17) respectively represent the constraints of bandwidth allocation, cloud computing resource allocation, preprocessing method selection, and sampling rate.

Constraint (18) is average service delay in the long term. Constraint (19) is processing accuracy. They are the constraints of ES  $i \in \mathcal{I}$  which have unique purposes, where  $H_i^{th}$  and  $A_i^{th}$  are requirements of application related threshold.  $H_i(\tau)$  denotes the ES  $i$ 's delay performance, so that  $H_i(\tau) = F_i^{ES}(\tau) + F_i^{tra}(\tau) + F_i^c(\tau)$ ,  $\forall i \in \mathcal{I}$ .

**Remark P1** is a dynamic stochastic optimization problem and we must decide all these decisions in the time slot. The problem's objective is minimizing system energy consumption in a long-term way in dynamic network. Due to two reasons, solving the problem is challenging: (i) due to the number of previous information is extremely large, the statistics of data arrival rates and time-varying channel conditions may be hard to obtain in IIoT systems; and (ii) because of the quick growth of the ES and IIoT devices's number, traditional dynamic programming is difficult to handle its large state space and action space, which will result in high computational complexity [11]. Lyapunov optimization [15] is used to solve long-term stochastic optimization problem, but preprocessing mode selection  $I_{i,y}^\tau$  and sampling rate adaption  $x_{i,k}^\tau$  are discrete binary decision variable. Further, constraints (18) and (19) are nonlinear, making P1 be a mixed integer nonlinear programming problems. To handle the problem, heuristic algorithms are a low-complexity solution. However, the solution cannot guarantee algorithm performance, so in IIoT applications, it is not recommended. To address this issue, an online algorithm is designed to solve this question in the next section. In the

first step, according to current network states, we use Lyapunov optimization method to decompose the long-term optimization problem into real-time optimization sub-problems. In the second step, based on Markov approximation technology, we developed an approximation algorithm. We consider the future impact and introduce the continuous probability model, and obtain the asymptotic optimal solution of the subproblem within the verified analysis range.

## 5. Online resource management algorithm

### 5.1 Lyapunov-based online method

Dealing with the long-term accuracy and delay constraints are main challenges to solve problem  $P_1$ . Therefore, we use the Lyapunov optimization method. For IIoT applications, Lyapunov optimization method constructs overdue delay queues and accuracy deficit queues. The problem decomposition's detailed procedure is shown as follows.

First, we define the delay overflow queues and accuracy deficit queues of IIoT applications. For each ES  $i$ , the dynamic changes of the accuracy deficit queue are represented below:

$$Q_i(\tau + 1) = [A_i^{th} - A_i(\tau)]^+ + Q_i(\tau). \quad (20)$$

Here, if  $\mathcal{Z}$  is a non-negative value,  $[\mathcal{Z}]^+ = \mathcal{Z}$ . Otherwise, it is 0.  $Q_i(\tau)$  indicates that there exists a deviation between the instantaneous accuracy and ES $i$ 's required long-term accuracy at time slot  $\tau$ .

We represent overdue delay queue's dynamic change of ES  $i$  below:

$$M_i(\tau + 1) = [H_i(\tau) - H_i^{th}]^+ + M_i(\tau), \quad (21)$$

where  $M_i(\tau)$  represents the  $i$ -th ES's deviation between computation task's service delay and long-term required delay in time slot  $\tau$ .

Then, Lyapunov function is defined according to [15] as

$$L(\Theta(\tau)) \triangleq \frac{1}{2} \sum_{i \in \mathcal{I}} [Q_i(\tau)^2 + M_i(\tau)^2], \quad (22)$$

where  $\Theta(\tau) \triangleq [Q_i(\tau), M_i(\tau)]$ .

Thus, the Lyapunov drift can be represented as

$$\Delta(\Theta(\tau)) = \mathbb{E}[L(\Theta(\tau + 1)) - L(\Theta(\tau)) | \Theta(\tau)]. \quad (23)$$

Accordingly, we represent the Lyapunov drift-penalty function as

$$\Delta_V(\Theta(\tau)) = \Delta(\Theta(\tau)) + V \cdot \mathbb{E}[E_i(\tau) | \Theta(\tau)], \quad (24)$$

where  $V \in (0, +\infty)$  is a control parameter. Next, the upper bound of  $\Delta_V(\Theta(\tau))$  is derived for any feasible solution  $J_i(\tau), \forall i \in \mathcal{I}$ , which is written in Theorem 1.1.

Theorem 1.1  $\Delta_V(Q(\tau))$  has an upper bounded for any  $J_i(\tau), \forall i \in \mathcal{I}$ , which can be written as

$$\begin{aligned} \Delta_V(\Theta(\tau)) \leq & B + \mathbb{E} \sum_{i=1}^N \{Q_i(\tau) [A_i^{th} - A_i(\tau)] + M_i(\tau) [H_i(\tau) - H_i^{th}] | \Theta(\tau)\} \\ & + V \cdot \mathbb{E}[E_i(\tau) | \Theta(\tau)], \end{aligned} \quad (25)$$

where  $B$  is a positive constant which can adjust the tradeoff between the satisfaction degree of the long-term accuracy and service delay constraints and the energy consumption cost.

**Proof:** We square two sides of accuracy deficit dynamics and we have

$$\begin{aligned} Q_i^2(\tau + 1) &= \left[ [A_i^{th} - A_i(\tau)]^+ \right]^2 + Q_i^2(\tau) + 2Q_i(\tau) [A_i^{th} - A_i(\tau)]^+ \\ &\leq [A_i^{th} - A_i(\tau)]^2 + Q_i^2(\tau) + 2Q_i(\tau) [A_i^{th} - A_i(\tau)]. \end{aligned} \quad (26)$$

We subtract  $Q_i^2(\tau)$  from two sides and then multiply by 0.5. Then, for  $i \in \mathcal{I} = \{1, 2, 3, \dots, N\}$ , we sum up these inequalities. So, we have

$$\frac{1}{2} \sum_{i=1}^N [Q_i^2(\tau + 1) - Q_i^2(\tau)] \leq \frac{1}{2} \sum_{i=1}^N [A_i^{th} - A_i(\tau)]^2 + \sum_{i=1}^N Q_i(\tau) [A_i^{th} - A_i(\tau)]. \quad (27)$$

Similarly, for delay overflow dynamics in (15), we can make such operations

$$\begin{aligned} M_i^2(\tau + 1) &= \left[ [H_i(\tau) - H_i^{th}]^+ \right]^2 + M_i^2(\tau) + 2M_i(\tau) [H_i(\tau) - H_i^{th}]^+ \\ &\leq [H_i(\tau) - H_i^{th}]^2 + M_i^2(\tau) + 2M_i(\tau) [H_i(\tau) - H_i^{th}]. \end{aligned} \quad (28)$$

We subtract  $M_i^2(\tau)$  from two sides and multiply by 0.5. Then, for  $i \in \mathcal{I}$ , we sum up these inequalities:

$$\frac{1}{2} \sum_{i=1}^N [M_i^2(\tau + 1) - M_i^2(\tau)] \leq \frac{1}{2} \sum_{i=1}^N [H_i(\tau) - H_i^{th}]^2 + \sum_{i=1}^N M_i(\tau) [H_i(\tau) - H_i^{th}]. \quad (29)$$

Combine (22) and (23), we can get

$$\begin{aligned} L(\Theta(\tau + 1)) - L(\Theta(\tau)) &\leq \frac{1}{2} \sum_{i=1}^N [A_i^{th} - A_i(\tau)]^2 + \frac{1}{2} \sum_{i=1}^N [H_i(\tau) - H_i^{th}]^2 \\ &\quad + Q_i(\tau) [A_i^{th} - A_i(\tau)] + M_i(\tau) [H_i(\tau) - H_i^{th}]. \end{aligned} \quad (30)$$

Finally,  $V \cdot E_i(\tau)$  is added to two sides of (30) and take expectation of two sides of  $\Theta(\tau)$  as the condition. Then, desired result can be obtained in (25), where  $B = \frac{1}{2} \sum_{i=1}^N [A_i^{th} - A_i(\tau)]^2 + \frac{1}{2} \sum_{i=1}^N [H_i(\tau) - H_i^{th}]^2$ .

The online joint preprocessing method selection, sampling rate adaption, and resource management algorithm aims to minimize  $\Delta_V(\Theta(\tau))$ 's upper bound in Theorem 1.1. The service delay and processing accuracy should be maintained at an expected level. At the same time, we can minimize the CC and ESs' system energy consumption. Algorithm 1 shows the details. Note that  $P2$ 's constraints and  $P1$ 's

constraints are the same. The  $P2$ 's objective function corresponds to the right side of (25). In every time slot  $\tau$ , we solve  $P2$  to obtain the optimal preprocessing method selection, sampling rate adaption, and resource management. Then, we update overdue delay queues and accuracy deficit queues.

---

**Algorithm 1:** Online Joint Sampling Rate Adaption, Preprocessing Mode and Resource Management Algorithm (OSPRA).

---

1. **Initialization:** At the beginning of slot  $\tau$ , collect the status information of CC, ESs, and each sensors.
  2. Observe the queue set  $\Theta(\tau)$ , channel gain  $h_i(\tau)$ , and data arrival rate  $\lambda_n(\tau)$  of  $n$ -th device.
  3. Determine  $x_{i,k}^\tau$  for each sensor,  $I_{i,y}^\tau$  for each ES,  $\alpha_i(\tau)$  for each edge cloud link, and  $c_i(\tau)$  for cloud computing by solving
 
$$\begin{aligned} P_2 : \min_{\{J_i(\tau)\}} U_i(\tau) &= B + \mathbb{E} \sum_{i=1}^N \{Q_i(\tau) [A_i^{th} - A_i(\tau)] + M_i(\tau) \cdot [H_i(\tau) - H_i^{th}] | \Theta(\tau)\} \\ &+ V \mathbb{E}[E_i(\tau) | \Theta(\tau)] \\ \text{s.t.} \quad &(14) - (19). \end{aligned}$$
  4. Update queue  $Q_i(\tau)$  and  $M_i(\tau)$  depending on (20) and (21).
  5. Return the best value of  $J_i(\tau)$ .
  6.  $\tau = \tau + 1$ .
- 

Sampling rate adaption  $x_{i,k}^\tau$  and preprocessing method selection  $I_{i,y}^\tau$  are discrete binary decision variable. Meanwhile, the CC's bandwidth and computation resource allocation is nonlinear. Therefore, problem  $P2$  is mixed integer nonlinear programming (MINLP) problem.

Theorem 1.2 Problem  $P2$  is NP-hard.

**Proof:** Firstly, we discuss a specific problem case. In this case, we fix the CC's bandwidth and computation resource allocation. Therefore, sampling rate and preprocessing method are selected discretely in problem  $P2$ . We can easily reduce the case to a multiple knapsack problem and the problem is known as NP-hard [16].

To address this issue, network configurations are set up as a time-reversible continuous-time Markov chain's states. We can prove that after a finite number of iterations, the CC's preprocessing method selection, sampling rate adaption, and the bandwidth and computation resource allocation can achieve stable states.

## 5.2 Approximately optimal solution for $P2$

Denote feasible solutions's set as  $J$ , and the feasible solution of problem  $P2$  satisfy  $j \in J$ . Denote the probability of adopting solution  $j$  at time slot  $\tau$  as  $q_j$ . Then, problem  $P2$  can be converted into the equivalent form:

$$\begin{aligned} \min_{q \geq 0} \quad & \sum_{j \in J} q_j \sum_{i \in \mathcal{I}} U_i(j, \tau) \\ \text{s.t.} \quad & \sum_{j \in J} q_j = 1, \end{aligned} \quad (31)$$

where  $\sum_{i \in \mathcal{I}} U_i(j, \tau)$  is  $q_j$ 's weight. The optimal solution of problem (31) results in minimum weight. We may transform the problem continuously by using log-sum-exp approximation [17].

First, convex log-sum-exp function  $G_\delta(j, \tau)$  is used to approximate the optimization objective  $U_i(j, \tau)$ , which is represented below:

$$G_\delta(j, \tau) = \frac{1}{\delta} \log \left[ \sum_{j \in J} \exp \left( \delta \sum_{i \in \mathcal{I}} U_i(j, \tau) \right) \right], \quad (32)$$

and we analytically show its approximation gap in Theorem 1.3.

**Theorem 1.3** The convex log-sum-exp function  $G_\delta(j, t)$  in (32) can approximate the optimization objective in (31) by

$$\min_{j \in J} \sum_{i \in \mathcal{I}} U_i(j, \tau) - \frac{1}{\delta} \log |J| \leq G_\delta(j, \tau) \leq \min_{j \in J} \sum_{i \in \mathcal{I}} U_i(j, \tau),$$

where  $\delta$  is a positive constant and the approximation gap is upper-bounded by  $\frac{1}{\delta} \log |J|$ .

**Proof:** Given the constant  $\delta$ , inequality holds:

$$\begin{aligned} \min_{j \in J} \sum_{i \in \mathcal{I}} U_i(j, \tau) & \geq \frac{1}{\delta} \log \left[ \exp \left( \delta \min_{j \in J} \sum_{i \in \mathcal{I}} U_i(j, \tau) \right) \right] \\ & \geq \min_{j \in J} \sum_{i \in \mathcal{I}} U_i(j, \tau) - \frac{1}{\delta} \log |J|. \end{aligned} \quad (33)$$

Convex log-sum-exp function's value precisely approximates min function's result, when  $\delta$  approaches infinity, i.e.,

$$\min_{j \in J} \sum_{i \in \mathcal{I}} U_i(j, \tau) = \lim_{\delta \rightarrow \infty} \frac{1}{\delta} \log \left[ \sum_{j \in J} \exp \left( \delta \sum_{i \in \mathcal{I}} U_i(j, \tau) \right) \right]. \quad (34)$$

What's more, the value of the problem's optimal solution and log-sum-exp function  $G_\delta(j, \tau)$  are equal according to Theorem 1.3, which is represented as follows:

$$\begin{aligned} \min_{q \geq 0} \quad & \sum_{j \in J} q_j \sum_{i \in \mathcal{I}} U_i(j, \tau) + \frac{1}{\delta} \sum_{j \in J} q_j \log q_j, \\ \text{s.t.} \quad & \sum_{j \in J} q_j = 1. \end{aligned} \quad (35)$$

Namely, we can convert problem (31) to problem (35).

By solving the KKT condition [17] of problem (35) can be represented as follows:

$$\begin{aligned} \sum_{i \in \mathcal{I}} U_i(j, \tau) - \frac{1}{\delta} \log q_j^* + \frac{1}{\delta} + \eta &= 0 \\ \sum_{j \in J} q_j^* &= 1, \\ \eta &\geq 0, \end{aligned} \quad (36)$$

We can obtain the probability distribution  $q_j^*$  of optimal solution as follows:

$$q_j^* = \frac{\exp(-\sum_{i \in \mathcal{I}} \delta U_i(j, \tau))}{\sum_{j' \in J} \exp(-\sum_{i \in \mathcal{I}} \delta U_i(j', \tau))}, \forall j \in J. \quad (37)$$

We can see that the different solutions's probabilities have direct ratio with corresponding weights  $U_i(j, \tau)$ . Every solution  $j \in J$  is paired with a specific state, we construct a time-reversible Markov sequential chain [18] which has a stationary distribution  $q_j^*$ . Switching one solution to another and transitioning between two states are equal. ES selecting new preprocessing mode and sampling rate and CC adopting new computation and communication resource allocation decision trigger it.

---

**Algorithm 2:** Markov Approximation-Based Algorithm for P2.

---

1. **Initialization:** Initialize  $U_i(j, \tau)$  by initializing sampling rate  $x_{i,k}^\tau$ , preprocessing mode  $I_{i,y}^\tau$ , and resource allocation in ESs and cloud center.
  2. **End initialization**
  3. **Loop:**
  4. Select a random solution and perform the following steps:
  5. Compute all other feasible solutions for the bound  $U_i(j, \tau)$ .
  6. Using the probability derived from (37), choose a feasible solution.
  7. **Update** the feasible solution.
  8. Record the optimal solution  $j^*$  when  $U_i(j^*, \tau)$  is the smallest.
  9. **End Loop**
- 

We must guarantee that random two states can convert if we want to build a time-reversible Markov sequential chain. Therefore, we limit one preprocessing method selection and sampling rate in ES and CC's one communication and computation resource allocation in time slot. If we make a decision of preprocessing mode  $I_i^y(\tau)$ , sampling rate  $x_{k_i}$ , computation and communication resource allocation  $\alpha_i(\tau)$ ,  $c_i(\tau)$ , previous solution  $j$  converts to new solution  $j'$  of transition rate  $q_{j,j'}$  non-negatively. To satisfy the time-reversibility feature, we have designed a transition rate which can satisfy the equation which is written as follows:

$$q_j^* \cdot q_{j,j'} = q_{j'}^* \cdot q_{j',j}, \forall j, j' \in J, \quad (38)$$

The feasible solution's transition rate is represented as

$$q_{j,j'} = \vartheta \exp \left[ -\frac{1}{2} \delta (U(j', \tau) - U(j, \tau)) \right], \forall j, j' \in J \quad (39)$$

where  $\vartheta$  is a positive constant. Transition rate  $q_{j,j'}$  increases if weight gap  $j' - j$  increases. It means that adopting a lower-weight solution has higher probability.

Algorithm 2 shows the algorithm based on Markov approximation which intends to solve problem P2. It can be executed on network platform. It can collect large amount of network state information to make real-time decisions. The algorithm combines in feasible solutions randomly to update time-reversible Markov sequential chain's state in update iterations. If  $U_i(j^*, \tau)$  is minimized by a feasible solution  $j^*$ , it will be recorded and the algorithm explores the following combination in solutions until all combinations have been attempted.

## 6. Performance analysis

The algorithm combines Markov approximation and Lyapunov optimization and the performance of algorithm is analyzed theoretically.

### 6.1 Time complexity analysis

The introduced algorithm includes two algorithms (algorithms 1 and 2) mainly. In algorithm 1, Lyapunov optimization is used to resolve resource management, preprocessing method selection, and sampling rate adaption in dynamic environment. Therefore, algorithm 1 generates many feasible solutions. In solution update iteration, algorithm 2 records the optimal solution found up to now until it explores all feasible solutions. The Markov approximation algorithm converges with linear rate quickly, through adjusting appropriate parameters. Therefore, we can get the asymptotic optimal solution quickly. In the process of iteration, a solution is chosen by the system randomly for updating control information. Because long-term problem is decomposed into some instant subquestions by using Lyapunov optimization, we focus on solving approximate solutions' complexity. As we defined in Section III, Sampling rate adaptation has  $K$  feasible solutions at most, and preprocessing method has  $y$  feasible solutions at most. There are  $yK$  feasible solutions at most in set  $U(j, \tau)$ , because both are discrete. Each ES traverses all the solutions. Denote  $\rho$  as the ESs' average iteration number which aims to get the stationary Markov chain. Moreover, OSPRA algorithm's complexity of time can be represented as  $O(Ky\rho)$ .

### 6.2 Optimality analysis

Theorem 1.4 We set up coefficients  $\delta$  and  $V$ , the optimality gap of initial problem's optimal solution and introduced algorithm's approximate solution theoretically is written as follows:

$$\sum_{\tau=0}^{T-1} \mathbb{E}[E_i(\tau) | \Theta(\tau)] \leq p^* + B/V + \log|J|/(\delta V), \quad (40)$$

where  $p^*$  means the optimal solution theoretically.

**Proof:** Since executing drift-penalty algorithm can get accuracy strategies  $H_i(\tau)$  and time delay  $A_i(\tau)$  in time slot  $\tau$ , we presume that accuracy actions  $H_i^*(\tau)$  and time delay  $A_i^*(\tau)$  are in the best decision.

Based on Theorem 1.1, the both sides' expectations can be represented as

$$\begin{aligned}
 \Delta_V(\Theta(\tau)) &= \Delta(\Theta(\tau)) + V \cdot \mathbb{E}[E_i(\tau)|\Theta(\tau)] \\
 &\leq B + \mathbb{E} \sum_{i=1}^N \{Q_i(\tau) [A_i^{th} - A_i^*(\tau)] + M_i(\tau) [H_i^*(\tau) - H_i^{th}] \Theta(\tau)\} + V \cdot \mathbb{E}[E_i^*(\tau)|\Theta(\tau)] \\
 &\leq B + V \cdot p^*.
 \end{aligned} \tag{41}$$

Then, by getting the summation of above derivation (41), we get

$$\begin{aligned}
 (B + V \cdot p^*) \cdot T &\geq \sum_{\tau=0}^{T-1} \mathbb{E}[\Delta_V(\Theta(\tau))|\Theta(\tau)] \\
 &= \mathbb{E}[L(\Theta(T))] + V \cdot \sum_{\tau=0}^{T-1} \mathbb{E}[E_i(\tau)|\Theta(\tau)] - \mathbb{E}[L(\Theta(0))].
 \end{aligned} \tag{42}$$

Finally, we move  $\mathbb{E}[L(\Theta(0))]$  to the inequality's left side, and divide two sides by  $V$ . Since  $\mathbb{E}[L(\Theta(0))] \geq 0$ , we can find Theorem 1.4's conclusion.

From Theorem 1.4, if  $V$  (the control parameter) is sufficiently large, the algorithm can obtain the approximate solution which reaches the optimal solution  $p^*$  infinitely.

## 7. Simulation results

We carry on simulations to evaluate the proposed online algorithm's performance on joint preprocessing method selection, sampling rate adaption, and resource management in IIoT systems with the support of edge-cloud collaboration. Particularly, we show the performance of energy consumption, service delay, and processing accuracy, respectively.

### 7.1 Simulation setup

We setup an IIoT system with edge-cloud collaboration which has a remote CC and multiple distributed ESs on site. For example, in bearing vibration fault monitoring applications, each ES connects to 10 IIoT devices and is used to collect mechanical equipments's data of bearing vibration. Then, by one of the three preprocessing methods, the raw data can be preprocessed (which are WT [19], BiNOSP [20], CLPM [10]) and are offloaded to CC for further data analysis. We define that there are three candidate sampling rates for IIoT devices, which are initial sampling rates 33, 66, and 100%, and set the initial sampling rate  $\varphi_K=18$  kHz [5]. Referring to [5], 0.59, 0.73, and 0.884 are respectively the three sampling rates' corresponding processing accuracies (**Table 2**).

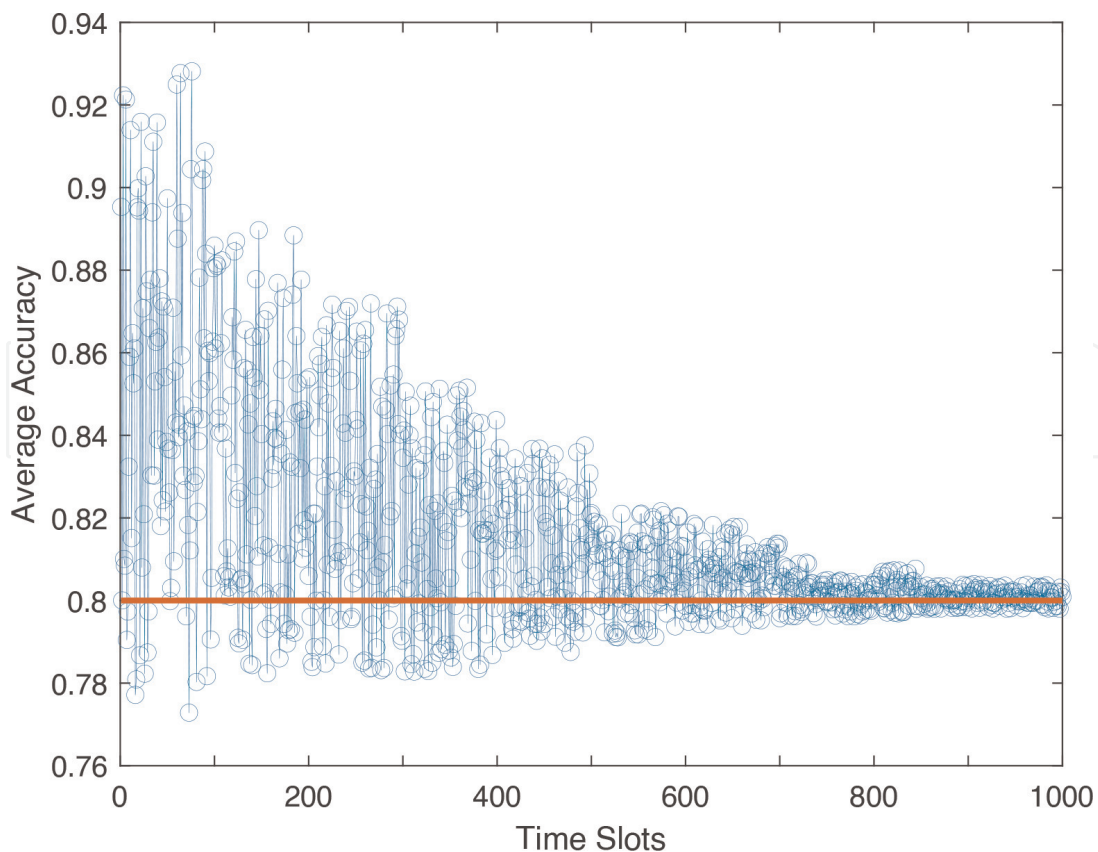
Besides, we simulate the following benchmarks for comparison, aiming to show the advantage of the OSPRA algorithm.



Parameter	Value	Parameter	Value
$\lambda_{i,n}^x$	[0.5, 1] data/s	$N$	10
$\varphi_K$	18 kHz	$A_i^{th}$	0.8
$h_c$	0.9	$f_i^y$	1.2, 1.7, 2.2G cycles/s
$W$	[5, 25] MHz	$g_y(y)$	1.3, 1.5, 1.7
$f_c$	2.8 G cycles/s	$p_i^{ES}$	500 mW
$m_y$	[0.7, 0.95]	$\beta_i, \epsilon_c$	550, 1200 cycles/bit
$z_i^{ES}, z_c$	$10^{-7}, 10^{-27}$	$N_0$	-174 dBm/Hz

**Table 2.**  
Simulation parameters.

- Accuracy-Guaranteed Resource Management Algorithm (AGRMA):** It does not preprocess data at edge servers and simply solves the joint sampling rate adaption, computing and communication resource allocation at the cloud. A deep reinforcement learning (DRL) method is used to address the random data arrival pattern and dynamic channel variation for guaranteeing long-term service accuracies [5].
- Lagrangian-Based Offloading Scheduling Algorithm (LOSPA):** It applies the Lagrangian dual decomposition method in a definitive way to solve a resource



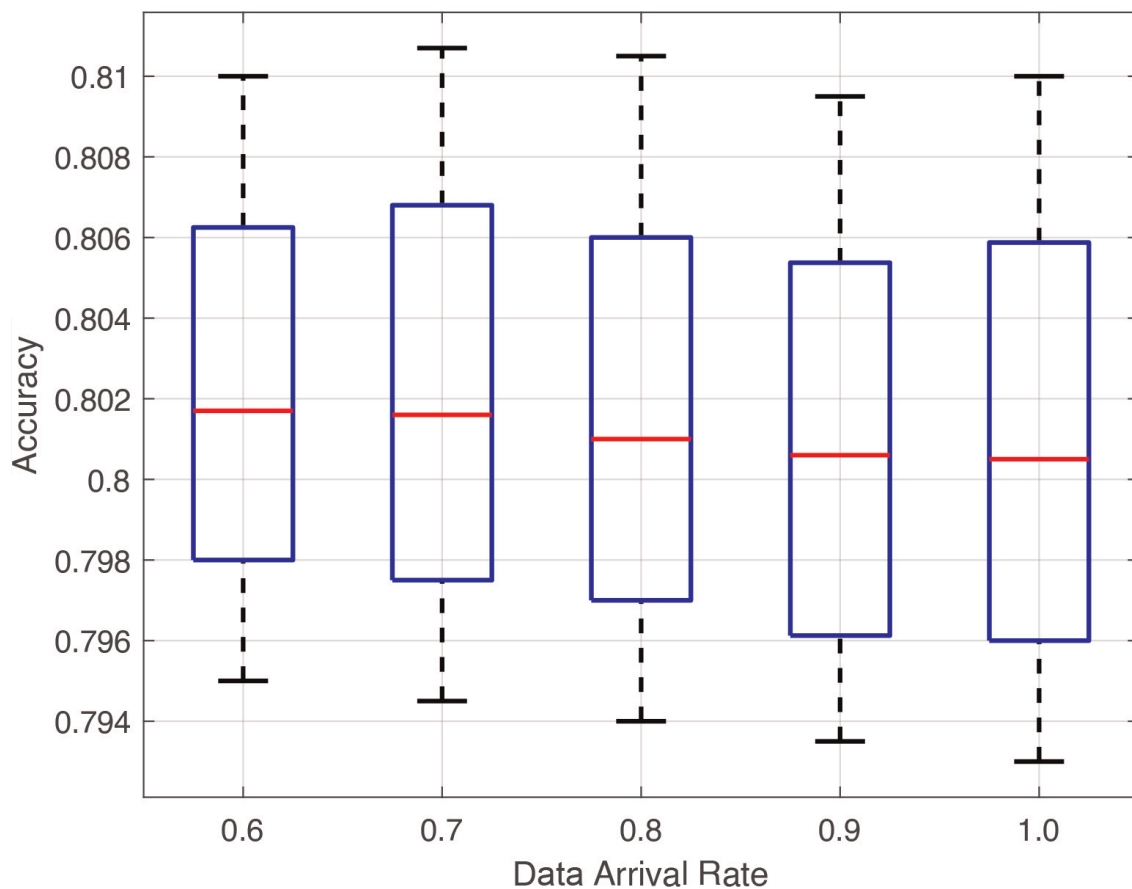
**Figure 2.**  
Introduced algorithm's accuracy performance.

management problem and optimal offloading decision for an edge-cloud collaboration framework. However, it ignores the network uncertainties [21].

## 7.2 Performance evaluation

Here, we consider the computation tasks' average processing accuracy in ESs over a long run, and illustrate the algorithm's convergence performance in **Figure 2**. This figure shows that the initial average accuracy maintains in a high level, as the initial accuracy value is initially defined as zero. Furthermore, since the obtainable communication and computation resources are sufficient at the beginning, this algorithm tends to increase the average accuracy to the target value very rapidly. Moreover, despite the fluctuations, the computation tasks' average accuracy converges to the target value quickly after some time slots. Therefore, it can be concluded that the algorithm's stability is verified by the simulation.

Considering that the arrival rate of the end terminal's data traffic is an uncontrolled environmental variable while it directly affects each computation task's size, the impact of various data arrival rate on the final data processing accuracy can significantly affect the adaptability of OSPRA in such complicated and noisy environments. **Figure 3** demonstrates the box-plot distribution of the data processing accuracy under different rates of data arrivals. By increasing this rate, the distribution range of the accuracy only fluctuates very slightly. In particular, it shows that the maximum probability of error is smaller than 0.7. Moreover, the average value of the processing

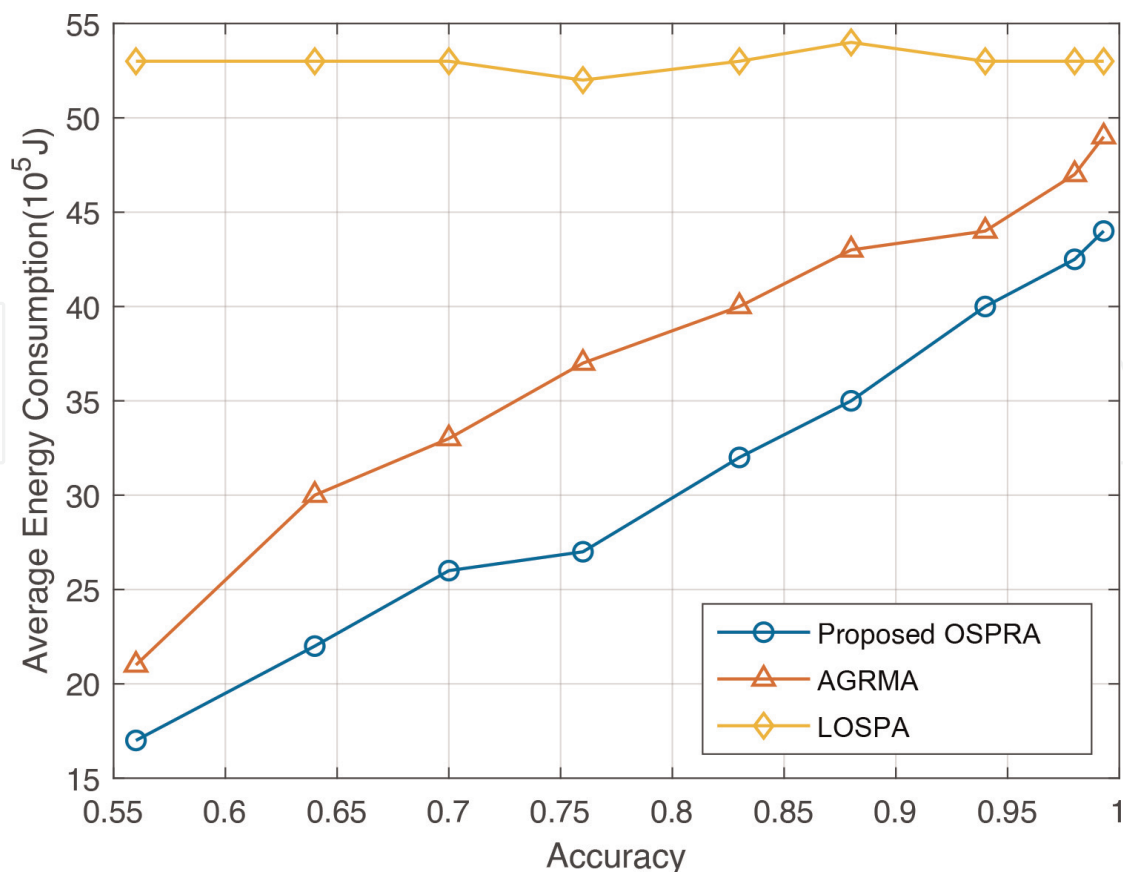


**Figure 3.** Introduced algorithm's accuracy performance in terms of data arrival rates.

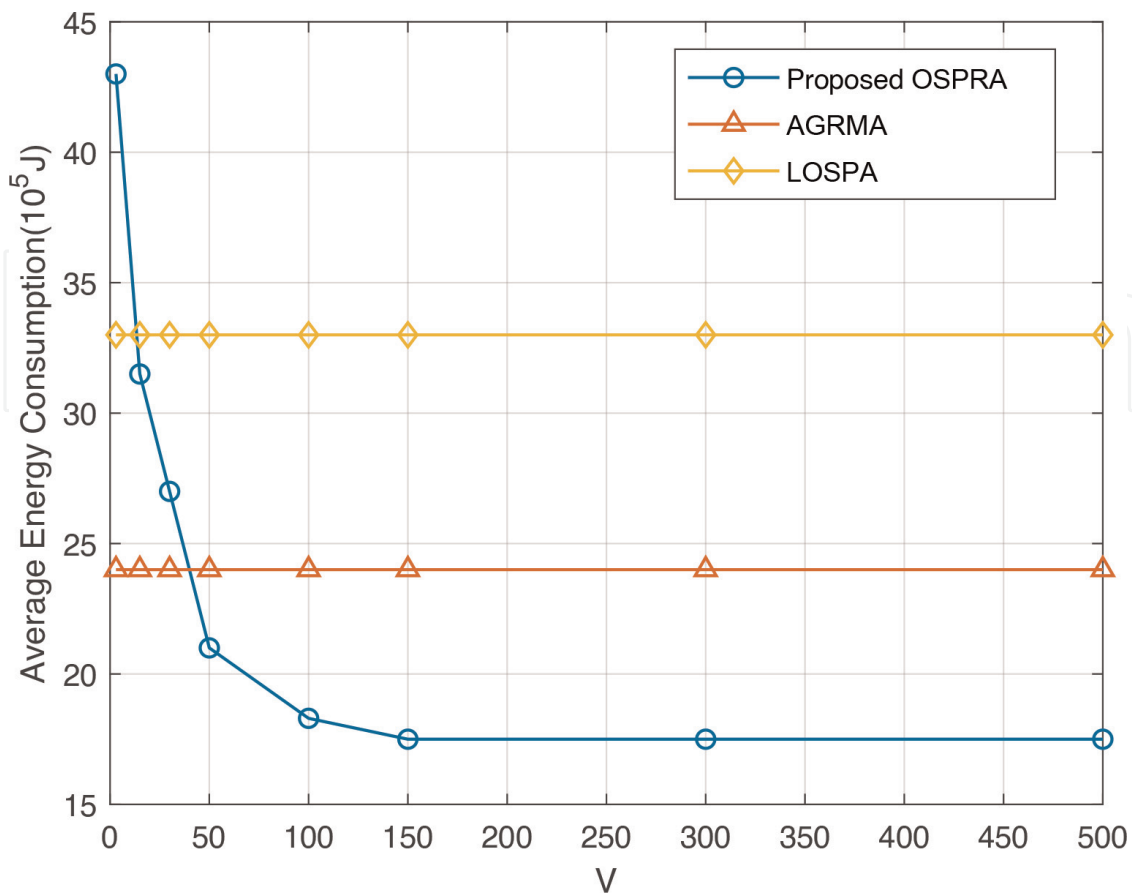
accuracy can be well controlled within a restricted range between 0.8 and 0.802, which indicates a good robustness of the proposed OSPRA to the instantaneous variations of the task size.

The average energy consumption of the proposed algorithm with various accuracy requirements is shown in **Figure 4**. It can be observed that if the accuracy requirement improves, the system has to in turn increase the sampling rate or choose a superior denoising performance in edge preprocessing method. As a result, the average energy consumption increases. LOSPA algorithm overlooks network uncertainties, so that the average energy consumption reduces smoothly. Changing processing accuracy does not have decisive performance impact. Oppositely, the average energy consumption increases for both OSPRA and AGRMA algorithms. However, in terms of minimizing the energy consumption, the proposed OSPRA obviously performs better. The reason is that AGRMA does not choose to preprocess at ESs, so that the sampling rate must increase when accuracy requirements increase. Oppositely, OSPRA can reach a trade-off between the preprocessing method selection and sampling rate, and therefore it can decrease the energy consumption and satisfy the accuracy requirements concurrently.

**Figure 5** evaluates the energy consumption of the overall system with the increase of the Lyapunov control parameter  $V$  for different algorithms. When  $V \leq 50$ , the average energy consumption rapidly reduces as  $V$  increases. When  $V \geq 100$ , the average energy consumption prefers to be stable, because computing resources and channel capacity have an upper limit. The asymptotic optimality of the proposed algorithm can be seen as there exists a bounded deviation between the proposed algorithm's



**Figure 4.** Introduced algorithm's average energy consumption in terms of different accuracy requirements.



**Figure 5.**  
Average energy with different parameter  $V$ .

average energy consumption and the optimal one, which numerically verifies Theorems 1.1 and 1.3. Besides, in this figure, we can find that the proposed OSPRA can control parameter  $V$  for adjusting the energy consumption weights of different users. That is to say, it guides us to select the parameter  $V$  according to various application requirements. It is worth noting that although we also draw the other two benchmark schemes' performances for comparison, both of them are independent of the control parameter  $V$ .

## 8. Conclusion

In this chapter, we have investigated a joint optimization problem of preprocessing method selection, sampling rate adaptation, and computing and communication resource allocation for IIoT systems with edge-cloud collaboration. With the objective of minimizing the energy consumption of the whole system while guaranteeing all applications' long-term service delay and data processing accuracy, a novel algorithm, called OSPRA, has been proposed. It has been proved that this proposed algorithm can solve the formulated problem in a dynamic way under network uncertainties. In addition, the feasibility and superiority of OSPRA have also been verified by extensive theoretical analysis and simulations.

IntechOpen

### **Author details**

You Shi<sup>1</sup>, Yuye Yang<sup>1</sup>, Changyan Yi<sup>1</sup>, Bing Chen<sup>1</sup> and Jun Cai<sup>2\*</sup>


1 Institution No. 1, The College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing, China

2 Institution No. 2, The Department of Electrical and Computer Engineering, Concordia University, Montréal, Canada

\*Address all correspondence to: jun.cai@concordia.ca

### **IntechOpen**

---

© 2023 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

## References

- [1] Mach P, Becvar Z. Mobile edge computing: A survey on architecture and computation offloading. *IEEE Communications Surveys & Tutorials*. 2017;**19**(3):1628-1656
- [2] Yi C, Cai J. A queueing game based management framework for fog computing with strategic computing speed control. *IEEE Transactions on Mobile Computing*. 2022;**21**(5):1537-1551
- [3] Yi C, Cai J, Su Z. A multi-user mobile computation offloading and transmission scheduling mechanism for delay-sensitive applications. *IEEE Transactions on Mobile Computing*. 2019;**19**(1):29-43
- [4] Yi C, Cai J. A truthful mechanism for scheduling delay-constrained wireless transmissions in IoT-based healthcare networks. *IEEE Transactions on Wireless Communications*. 2018;**18**(2): 912-925
- [5] Wu W, Yang P, Zhang W, Zhou C, Shen S. Accuracy-guaranteed collaborative dnn inference in industrial IoT via deep reinforcement learning. *IEEE Transactions on Industrial Informatics*. 2020;**17**(7):4988-4998
- [6] Yamaguchi M, Maruta K, Ono H. Operating mechanism for RF electromagnetic noise suppression sheets. *IEEE Transactions on Magnetics*. 2005;**41**(10):3565-3567
- [7] Wang T, Ke H, Zheng X, Wang K, Liu A. Big data cleaning based on mobile edge computing in industrial sensor-cloud. *IEEE Transactions on Industrial Informatics*. 2019;**16**(2):1321-1329
- [8] Mao W, Zhao Z, Chang Z, Min G, Gao W. Energy efficient industrial internet of things: Overview and open issues. *IEEE Transactions on Industrial Informatics*. 2021;**PP**(99):1-1
- [9] Hariharakrishnan J, Mohanavalli S, Srividya, Kumar K. Survey of pre-processing techniques for mining big data. In: 2017 International Conference on Computer, Communication and Signal Processing (ICCCSP). Chennai, India: IEEE; 2017. pp. 1-5. <https://ieeexplore.ieee.org/author/37086002878>
- [10] Rui L, Zhu Y, Gao Z, Qiu X. CLPM: A cooperative link prediction model for industrial internet of things using partitioned stacked denoising autoencoder. *IEEE Transactions on Industrial Informatics*. 2020;**17**(5): 3620-3629
- [11] Guo K, Gao R, Xia W, Quek T. Online learning based computation offloading in MEC systems with communication and computation dynamics. *IEEE Transactions on Communications*. 2020;**69**(2): 1147-1162
- [12] Zhang X, Mao Y, Zhang J, Letaief KB. Multi-objective resource allocation for mobile edge computing systems. In: 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC). Montreal, Canada: PIMRC; 2017. pp. 1-5
- [13] Lin R, Zhou Z, Luo S, Xiao Y, Zukerman M. Distributed optimization for computation offloading in edge computing. *IEEE Transactions on Wireless Communications*. 2020;**19**(12): 8179-8194
- [14] Mao S, Leng S, Maharjan S, Zhang Y. Energy efficiency and delay tradeoff for wireless powered mobile-edge

computing systems with multi-access schemes. *IEEE Transactions on Wireless Communications*. 2019;**19**(3): 1855-1867

[15] Neely MJ. Stochastic network optimization with application to communication and queueing systems. *Synthesis Lectures on Communication Networks*. 2010;**3**(1):1-211

[16] Pisinger D. Where are the hard knapsack problems? *Computers & Operations Research*. 2005;**32**(9): 2271-2284

[17] Boyd S, Vandenberghe L. *Convex Optimization*. UK: Cambridge University Press; 2004

[18] Chen M, Liew SC, Shao Z, Kai C. Markov approximation for combinatorial network optimization. *IEEE Transactions on Information Theory*. 2013;**59**(10):6301-6327

[19] Xie ZJ, Song BY, Zhang Y, Zhang F. Application of an improved wavelet threshold denoising method for vibration signal processing. *Advanced Materials Research*. 2014;**889-890**: 799-806

[20] Zhang H, Chen X, Zhang X, Zhang X. A bi-level nested sparse optimization for adaptive mechanical fault feature detection. *IEEE Access*. 2020;**8**:19 767-19 782

[21] Kuang Z, Li L, Gao J, Zhao L, Liu A. Partial offloading scheduling and power allocation for mobile edge computing systems. *IEEE Internet of Things Journal*. 2019;**6**(4):6774-6785