SCIENCE CHINA
Information Sciences

• **Supplementary File** •

Trustworthy DNN Partition for Blockchain-enabled Digital Twin in Wireless IIoT Networks

Xiumei Deng[1], Jun Li[6*], Long Shi[2*], Kang Wei[3], Ming Ding[4], Yumeng Shao[2], Wen Chen[5] & Shi Jin[6]

[1]*Pillar of Information Systems Technology and Design, Singapore University of Technology and Design, Singapore* 487372*, Singapore;*
[2]*School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing* 210094*, China;*
[3]*Department of Computing, Hong Kong Polytechnic University, Hong Kong* 999077*, China;*
[4]*Data61, CSIRO, Sydney* NSW 2015*, Australia;*
[5]*Department of Electronics Engineering, Shanghai Jiao Tong University, Shanghai* 200240*, China; National*
[6]*Mobile Communications Research Laboratory, Southeast University, Nanjing* 210096*, China*

Appendix A System Model

Appendix A.1 DNN Inference and Offloading

Following the work of [1], we formulate the stochastic data arrivals at the gateway side as a homogeneous Poisson process. Let $\mathcal{D}_n(t) = \{\boldsymbol{x}_{n,i} \in \mathbb{R}^d, y_{n,i} \in \mathbb{R}\}_{i=1}^{D_n(t)}$ denote the new data points collected by the n-th device at time slot t, where $\boldsymbol{x}_{n,i}$ and $y_{n,i}$ are the feature vector and label of the i-th data point newly collected by the n-th device at time slot t, respectively. Specifically, the new data points $D_n(t)$ collected by the n-th device and transmitted to its corresponding DT in each time slot constitute an independent and identically distributed (i.i.d.) exponential random variable with an average rate Θ_n, which is positively correlated with both the data collection rate of the n-th device and the data transmission rate from the n-th device to the associated gateway. Each device holds a local dataset that is continuously collected from its equipped sensors and running applications, and keeps synchronising its collected data with the corresponding DT maintained on its associated gateway [2, 3].

Before calculating the latency and energy consumption for the DNN inference at the gateway and AP sides, we first introduce the notations for hyper-parameters and tensor shapes involved in the DNN inference process as follows. Let B_s and S_f denote the batch size and the precision format of the data type, respectively. For convolution layers and pooling layers, H_o, W_o and C_o represent the output height, width, and channel, respectively; H_i, W_i and C_i represent the input height, width, and channel; H_f and W_f are the height and width of the filter. For the fully connected layers, S_i and S_o refer to the input and output sizes, respectively. We present the main layer-level forward output size and floating-point operation counts (FLOPs) in the DNN inference process in Table B1 to calculate the number of bits of the forward output of the bottom-layer DNN inference and FLOPs for the bottom-layer and top-layer DNN inference performed at the gateway and AP sides. Let $\mathcal{L}_n = \{1, ..., L_n\}$ denote the index set of

Table A1 Layer-level forward output size and FLOPs in the DNN inference process.

Layer Category	Forward Output Size (Bits)	FLOPs
Convolution	$S_f B_s C_o H_o W_o$	$2B_s C_i H_f W_f C_o H_o W_o$
Pooling	$S_f B_s C_o H_o W_o$	$B_s C_i H_i W_i$
Fully Connected	$B_s S_o$	$2B_s S_i S_o$

the DNN layers for the n-th DT. In the t-th time slot, the bottom $l_n(t)$ layers of the DNN inference are executed locally at the gateway side, and the top $L_n - l_n(t)$ layers of the DNN inference tasks are offloaded to the AP side. Let χ_n^l denote the FLOPs required by the n-th DT to perform the l-th layer of the DNN inference for each data point. As such, the DNN inference time of the m-th gateway in the t-th time slot is

$$\tau_m^{\mathrm{exe,G}}(t) = \frac{\sum_{n=1}^{N} a_{n,m} D_n(t) \sum_{l=1}^{l_n(t)} \chi_n^l}{\phi_m^{\mathrm{G}} f_m^{\mathrm{G}}}, \tag{A1}$$

where ϕ_m^{G} is the FLOPs per clock cycle of the m-th gateway, and f_m^{G} is the computation frequency of the m-th gateway for the DNN inference. The energy consumption of the m-th gateway for DNN inference tasks in the t-th time slot is

$$e_m^{\mathrm{exe,G}}(t) = \frac{v_m^{\mathrm{G}} \left(f_m^{\mathrm{G}}\right)^2}{\phi_m^{\mathrm{G}}} \sum_{n=1}^{N} a_{n,m} D_n(t) \sum_{l=1}^{l_n(t)} \chi_n^l, \tag{A2}$$

* Corresponding author (email: slong1007@gmail.com, jleesr80@gmail.com)

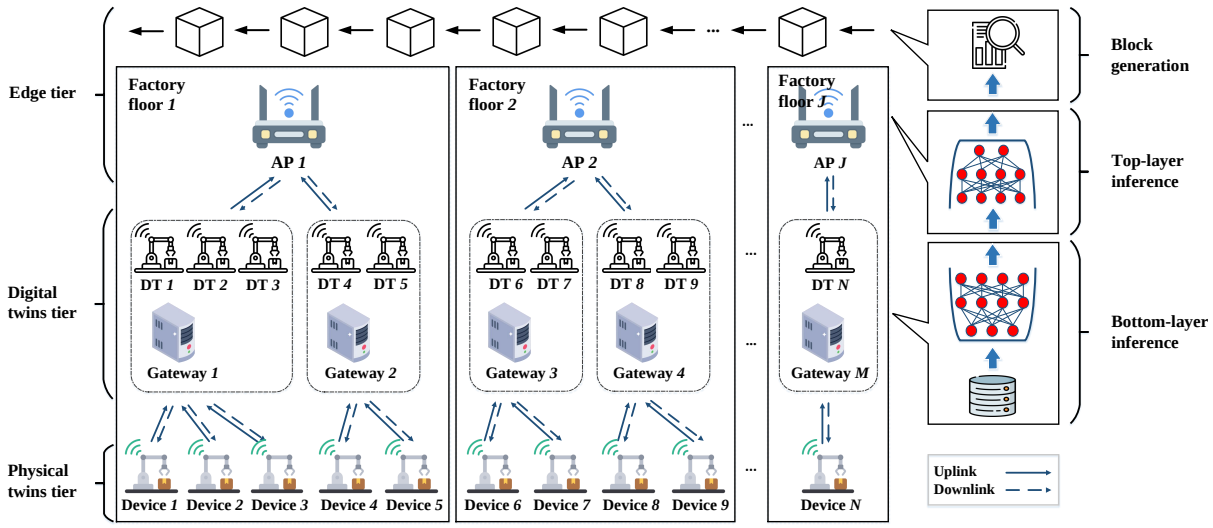

Figure A1 The system model of the three-tier blockchain-enabled DT framework for wireless IIoT networks.

where v_m^{G} is the effective switched capacitance.

Assume that the wireless channels between the gateways and APs experience i.i.d. block fading [4, 5]. Specifically, the wireless channel remains static in each time slot but varies among different time slots. We model the data transmission between the gateways and APs using the multiple channel access method of orthogonal frequency division multiple access (OFDMA). Define an $M \times J$ connection matrix as $\boldsymbol{b}$ with entry $b_{m,j} \in \{0, 1\}$, $\forall m \in \mathcal{M}$, and $j \in \mathcal{J}$. If $b_{m,j} = 1$, the m-th gateway is deployed with the j-th AP on the same factory floor, and offloads the top-layer DNN inference tasks to the j-th AP via wireless link in each time slot. Notably, each gateway can only communicate with the AP deployed on the same shop floor, i.e., $\sum_{j=1}^{J} b_{m,j} = 1$, $\forall m \in \mathcal{M}$. The uplink channel power gain from the m-th gateway to its associated AP is modeled as

$$H_m(t) = h_0 \rho_m(t) \left(\frac{d_0}{d_m}\right)^{\nu}, \tag{A3}$$

where h_0 is the path loss constant, $\rho_m(t)$ is the small-scale fading channel power gain from the m-th gateway to its associated AP in the t-th time slot, d_m is the distance from the m-th gateway to its associated AP, d_0 is the reference distance, and ν is the large-scale path loss factor, respectively. Let o_n^l denote the forward output size of the l-th layer for the n-th DT in the DNN inference process. The data transmission time from the m-th gateway to the associated AP is

$$\tau_m^{\mathrm{off}}(t) = \frac{\sum_{n=1}^{N} a_{n,m} D_n(t) o_n^{l_n(t)}}{B \log\left(1 + \frac{P_m H_m(t)}{\eta_m(t) + N_0 B}\right)}, \tag{A4}$$

where P_m is the transmit power of the m-th gateway, N_0 is the noise power spectral density, and $\eta_m(t)$ is the co-channel interference. The energy consumption of the m-th gateway for DNN inference task offloading is represented as

$$e_m^{\mathrm{off}}(t) = \frac{P_m \sum_{n=1}^{N} a_{n,m} D_n(t) o_n^{l_n(t)}}{B \log\left(1 + \frac{P_m H_m(t)}{\eta_m(t) + N_0 B}\right)}. \tag{A5}$$

The time consumption for the top-layer DNN inference offloaded from the m-th gateway is

$$\tau_m^{\mathrm{exe,A}}(t) = \frac{\sum_{n=1}^{N} a_{n,m} D_n(t) \sum_{l=l_n(t)+1}^{L_n} \chi_n^l}{\phi_m^{\mathrm{A}} f_m^{\mathrm{A}}(t)}, \tag{A6}$$

where ϕ_m^{A} is the FLOPs per clock cycle of the m-th gateway's associated AP, and $f_m^{\mathrm{A}}(t)$ is the computation frequency of the m-th gateway's associated AP for DNN inference tasks. The energy consumption for the top-layer DNN inference offloaded from the m-th gateway is represented as

$$e_m^{\mathrm{exe,A}}(t) = \frac{v_m^{\mathrm{A}} \left(f_m^{\mathrm{A}}(t)\right)^2}{\phi_m^{\mathrm{A}}} \sum_{n=1}^{N} a_{n,m} D_n(t) \sum_{l=l_n(t)+1}^{L_n} \chi_n^l, \tag{A7}$$

where v_m^{A} is the effective switched capacitance of the m-th gateway's associated AP.

Appendix A.2 Reputation Based Consensus Mechanism

We next clarify the statistical property of the stochastic process with respect to the block generation. From [6, 7], the successful query attempts of the j-th AP converge to a Poisson process with the average rate given by

$$\theta_j(t) = \frac{f_j^{\mathrm{bloc}}(t)}{\gamma_j(t)}, \tag{A8}$$

where $f_j^{\text{bloc}}(t)$ is the computation frequency of the j-th AP for block generation in the t-th time slot. Since the APs work independently on the hash puzzle in each time slot, the successful query attempts of the entire blockchain network can be formulated as a Poisson process with the average rate:

$$\hat{\theta}(t) = \sum_{j=1}^{J} \theta_j(t). \tag{A9}$$

As such, the block generation time in each time slot, denoted by $\tau^{\text{bloc}}(t)$, can be formulated as an i.i.d. exponential random variable with the average rate $\hat{\theta}(t)$. Based on the theorem of order statistics [8], the cumulative distribution function of the block generation time in the t-th time slot is given by

$$\Pr(\tau^{\text{bloc}}(t) < \tau) = 1 - e^{-\hat{\theta}(t)\tau}. \tag{A10}$$

Assume that the new block can be generated when $p_0 = \Pr(\tau^{\text{bloc}} < \tau)$ approaches one. Thus, the block generation time in the t-th time slot is given by

$$\tau^{\text{bloc}}(t) = -\frac{\ln(1-p_0)}{\hat{\theta}(t)}. \tag{A11}$$

The energy consumption of the j-th AP for block generation in the t-th time slot is expressed as

$$e_j^{\text{bloc}}(t) = v_j^A \tau^{\text{bloc}}(t) \Big(f_j^{\text{bloc}}(t)\Big)^3. \tag{A12}$$

Appendix A.3 Summary of main notations

For ease of reference, Tables B2 lists the main notations used in this paper.

Table A2 Summary of main notations.

Notation	Description
$\mathcal{N}$	Index set of the devices
$\mathcal{M}$	Index set of the gateways
$\mathcal{J}$	Index set of the APs
$\mathcal{T}$	Index set of the time slots
$a_{n,m}$	Connection indicator between the n-th device and the m-th gateway
$b_{m,j}$	Connection indicator between the m-th gateway and the j-th AP
$D_n(t)$	New data point arrivals at the n-th device at time slot t
$\mathcal{L}_n$	Index set of the DNN layers at the n-th DT
$l_n(t)$	Number of bottom layers of DNN inference tasks performed at the gateway side at time slot t
$\tau_m^{\text{exe,G}}(t)$	DNN inference time of the m-th gateway at time slot t
$e_m^{\text{exe,G}}(t)$	Energy consumption of the m-th gateway for DNN inference at time slot t
$\tau_m^{\text{off}}(t)$	Data transmission time from the m-th gateway to its associated AP at time slot t
$e_m^{\text{off}}(t)$	Energy consumption of the m-th gateway for DNN inference task offloading at time slot t
$\tau_m^{\text{exe,A}}(t)$	Time consumption for top-layer DNN inference tasks offloaded from the m-th gateway at time slot t
$f_m^{\text{A}}(t)$	Computation frequency of the m-th gateway's associated AP for DNN inference at time slot t
$e_m^{\text{exe,A}}(t)$	Energy consumption for top-layer DNN inference tasks offloaded from the m-th gateway at time slot t
$U_j(t)$	Off-chain reputation of the j-th AP at time slot t
$O_j(t)$	Top-layer DNN reference tasks offloaded from the gateway side to the j-th AP at time slot t
$\gamma_j(t)$	Block generation difficulty at time slot t
$\tau^{\text{bloc}}(t)$	Block generation time at time slot t
$e_j^{\text{bloc}}(t)$	Energy consumption of the j-th AP for block generation at time slot t
$\tau(t)$	Duration of time slot t
$e_m^{\text{G}}(t)$	Energy consumption of the m-th gateway at time slot t
$e_j^{\text{A}}(t)$	Energy consumption of the j-th AP at time slot t

Appendix B Problem Solution

We next introduce a dynamic **D**NN **p**artitioning and **r**esource **a**llocation (DPRA) algorithm to solve the long-term stochastic problem in **P0**, which is shown in **Algorithm** 1. By leveraging the Lyapunov optimization method, DPRA first transforms the long-term stochastic optimization problem in **P0** into a sequence of one-shot static optimization problems, and then subsequently solves the transformed deterministic problem in each time slot. Unlike existing DNN partitioning approaches that employ a predefined partition point, the proposed DPRA algorithm dynamically optimizes both the DNN partitioning point and the computation frequency considering time-varying channels and energy arrivals. As shown in **Algorithm 1**, our proposed DPRA algorithm is performed at the AP side to optimize DNN partitioning points and computation resource allocation in each time slot. To be specific, at the beginning of each time slot, APs exchange the virtual queue length and channel state information with each other. Based on the collected virtual queue length and channel state information, our proposed DPRA algorithm is performed at each AP to optimize DNN partitioning point $\boldsymbol{l}(t)$, computation frequency of the APs for DNN inference $\boldsymbol{f}^{\text{A}}(t)$ and block mining $\boldsymbol{f}^{\text{bloc}}(t)$ in each time slot.

Algorithm B1 Dynamic DNN partitioning and resource allocation (DPRA) algorithm

Require: Auxiliary queue lengths and channel state in the t-th time slot;
Ensure: $\boldsymbol{X}(t) = \left[\boldsymbol{l}(t), \boldsymbol{f}^{\mathrm{A}}(t), \boldsymbol{f}^{\mathrm{bloc}}(t)\right]$;

1: Initialize the auxiliary queue lengths $\boldsymbol{Q}(t) = \frac{1}{2}\left(U^{\min} + U^{\max}\right)$, $\boldsymbol{S}(t) = \frac{1}{2}\left(U^{\min} + U^{\max}\right)$;
2: **for** $t = 1, 2, ..., T$ **do**
3: Optimize the DNN partirion point $\boldsymbol{l}(t)$, the computation frequency for the top-layer DNN inference $\boldsymbol{f}^{\mathrm{A}}(t)$, and the computation frequency for block generation $\boldsymbol{f}^{\mathrm{bloc}}(t)$ with block coordinate descent method;
4: Update $\boldsymbol{Q}(t)$ and $\boldsymbol{S}(t)$ according to (B1) and (B2);
5: **end for**

Appendix B.1 Problem Transformation

To decouple the long-term stochastic optimization problem presented in **P0** into a sequence of one-shot static problems, we first define two auxiliary queues for each AP as

$$Q_j(t+1) = \max\left\{Q_j(t) - U_j(t) + U^{\min}, 0\right\}, \tag{B1}$$

and

$$S_j(t+1) = \max\left\{S_j(t) + U_j(t) - U^{\max}, 0\right\}. \tag{B2}$$

According to Lyapunov optimization method [9–11], the long-term off-chain reputation constraint **C6** can be equivalently transformed into the queue stability constraints for the auxiliary queues $Q_j(t)$ and $S_j(t)$ in (B1) and (B2) as

$$\lim_{t\to\infty} \frac{\mathbb{E}\left\{|Q_j(t)|\right\}}{t} = 0, \forall j \in \mathcal{J}, \tag{B3}$$

and

$$\lim_{t\to\infty} \frac{\mathbb{E}\left\{|S_j(t)|\right\}}{t} = 0, \forall j \in \mathcal{J}. \tag{B4}$$

By substituting the long-term inequality constraint **C6** with the queue stability constraints in (B3) and (B4), the original problem in **P0** can be rewritten as

$$\begin{aligned} \mathbf{P1}: \ & \min_{\boldsymbol{X}(t)} \ \overline{\tau} = \frac{1}{T}\sum_{t=1}^{T}\tau(t) \\ \text{s.t.} \ \ & \mathbf{C1} \sim \mathbf{C5}, \\ & \mathbf{C7}: \lim_{t\to\infty} \frac{\mathbb{E}\left\{|Q_j(t)|\right\}}{t} = 0, \forall j \in \mathcal{J}, \quad \mathbf{C8}: \lim_{t\to\infty} \frac{\mathbb{E}\left\{|S_j(t)|\right\}}{t} = 0, \forall j \in \mathcal{J}. \end{aligned} \tag{B5}$$

The transformed problem **P1** is now a standard optimization problem for the Lyapunov optimization method. To solve **P1**, we proceed by formulating the Lyapunov function, characterizing the conditional Lyapunov drift, and minimizing the Lyapunov drift-plus-penalty ratio function as follows.

Lyapunov function: The Lyapunov function is defined as

$$L(t) = \frac{1}{2}\sum_{j=1}^{J}\left(\left(Q_j(t)\right)^2 + \left(S_j(t)\right)^2\right). \tag{B6}$$

Conditional Lyapunov drift: Let $\boldsymbol{\Phi}(t) = \left\{Q_j(t), S_j(t), \forall j \in \mathcal{J}\right\}$ be a set that collects all auxiliary queues in the t-th time slot. We define the conditional Lyapunov drift as

$$\Delta L(t) = \mathbb{E}\left\{L(t+1) - L(t) | \boldsymbol{\Phi}(t)\right\}. \tag{B7}$$

Lyapunov drift-plus-penalty function: Given any Lyapunov control parameter $V > 0$, we define the Lyapunov drift-plus-penalty function as

$$\Delta(t) = V\tau(t) + \Delta L(t). \tag{B8}$$

From [9–11], minimizing the conditional Lyapunov drift $\Delta L(t)$ promotes the stability of the auxiliary queues $Q_j(t)$ and $S_j(t)$. Therefore, we minimize the Lyapunov drift-plus-penalty function to reduce the system latency while ensuring the stability of the auxiliary queues. Note that the control parameter V allows to adjust the trade-off between minimising system latency and satisfying the long-term off-chain reputation constraint. For delay-sensitive applications, we can reduce the system latency by setting a larger V value, albeit at the cost of decreased block generation difficulty.

Lemma B1. Given the auxiliary queue lengths $\boldsymbol{\Phi}(t)$, the conditional Lyapunov drift $\Delta L(t)$ is upper bounded by

$$\begin{aligned} \Delta L(t) \leqslant & \sum_{j=1}^{J}\mathbb{E}\Big\{Q_j(t)\left(U^{\min} - U_j(t)\right) \\ & + S_j(t)\left(U_j(t) - U^{\max}\right)\Big|\boldsymbol{\Phi}(t)\Big\} + H, \end{aligned} \tag{B9}$$

where $H = \sum_{j=1}^{J} g\left(\sum_{m=1}^{M}\sum_{n=1}^{N} b_{m,j}a_{n,m}D_n(t)\sum_{l=1}^{L_n}\chi_n^l\right)^2 + \frac{J}{2}\left(\left(U^{\min}\right)^2 + (U^{\max})^2\right)$.

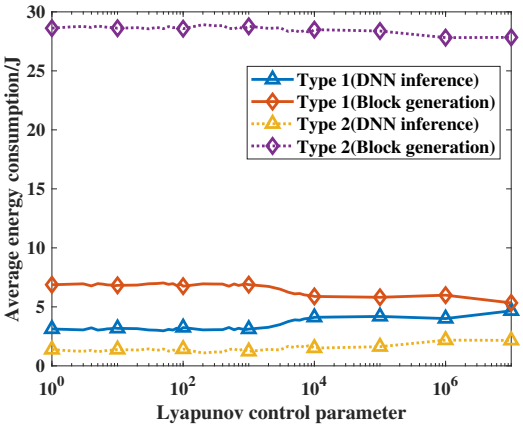

Figure C1 The average energy consumption comparison between DNN inference and block generation under DPRA.

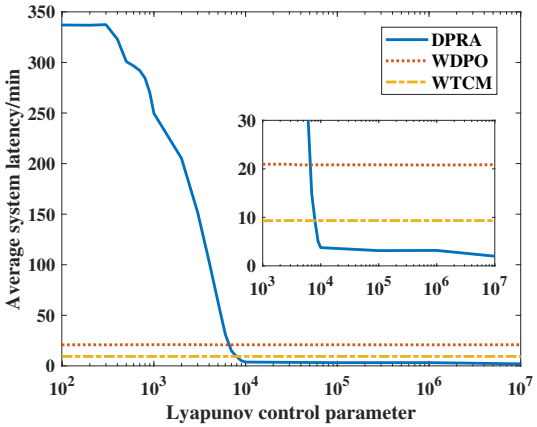

Figure C2 The average system latency comparison between DPRA and baselines.

Based on the derived upper bound of the conditional Lyapunov drift $\Delta L(t)$ in (B9), the goal of the DPRA algorithm is to minimize the Lyapunov drift-plus-penalty function in each time slot as

$$\mathbf{P2}: \min_{\boldsymbol{X}(t)} V\tau(t) + \sum_{j=1}^{J} \left(S_j(t) - Q_j(t)\right) U_j(t) \tag{B10}$$

$$\text{s.t.} \quad \mathbf{C1} \sim \mathbf{C5}.$$

The mixed-integer nonlinear problem in **P2** can be solved using the block coordinate descent method. Note that the mixed-integer nonlinear problem in **P2** is NP-hard. As shown in **Algorithm** 1, we optimize the DNN partirion point $\boldsymbol{l}(t)$, the computation frequency for the top-layer DNN inference $\boldsymbol{f}^{\mathrm{A}}(t)$, and the computation frequency for block generation $\boldsymbol{f}^{\mathrm{bloc}}(t)$ with block coordinate descent method. Note that the sub-problems are exactly solved with optimality in each iteration in order to guarantee the convergence to at least a local optimum [12–14].

Appendix C Experimental Results

In this section, we evaluate the performance of the proposed DPRA algorithm. Consider that the B-DT system consists of $N = 60$ devices, $M = 20$ gateways, and $J = 4$ APs. Each gateway is associated with 3 devices, and each AP is associated with 5 gateways. The APs are equally divided into two types. For **Type 1** APs and the associated devices, we set the energy arrivals $E_j^{\mathrm{A,max}} = 10$ J and the data arrivals $\Theta_n = 100$. For **Type 2** APs and the associated devices, we set $E_j^{\mathrm{A,max}} = 30$ J and $\Theta_n = 50$. For each AP, $v_j^{\mathrm{A}} = 10^{-24}$, $\phi_m^{\mathrm{A}} = 32$ FLOPs per CPU cycle, and $f_j^{\max} = 0.1$ GHz. For each gateway, $E_j^{\mathrm{A,max}} = 0.5$ J, f_m^{G} is uniformly distributed within $[1, 10]$ MHz, $\phi_m^{\mathrm{G}} = 8$ FLOPs per CPU cycle, $v_m^{G} = v_j^{\mathrm{A}} = 10^{-24}$, $P_m = 100$ mW, and d_m is uniformly distributed within $[0, 50]$ m. The channel parameters are set as $d_0 = 1$ m, $h_0 = 10^{-3}$, $B = 5$ MHz, and $N_0 = -174$ dBm/Hz. The uplink inference $\eta_m(t)$ is produced by the Gaussian distribution, and the channel power gain $\rho_m(t)$ is exponentially distributed with unit mean. Besides, we set $p_0 = 1 - 10^{-15}$, $\alpha = 5 \times 10^{-5}$, $\beta = -29$, $U^{\min} = 25$, and $U^{\min} = 75$. In addition, we adopt VGG-11 on Cifar-10 dataset and CNN on Fashion-MNIST dataset to demonstrate the DTs' DNN inference tasks.

- **Fashion-MNIST**. Fashion-MNIST consists of 60000 28 × 28 grayscale images of 10 fashion categories, along with a test set of 10000 images.
- **CIFAR**-10. The CIFAR-10 dataset consists of 60000 32 × 32 RGB images in 10 classes (from 0 to 9), with 50000 training images and 10000 test images per class.

All the experiments are conducted on personal computers with 11-th Gen Intel(R) Core(TM) i7-11800H @2.30GHZ CPU and NVIDIA GeForce RTX 3070 GPU.

Fig. C2 shows the average system latency under the DPRA algorithm over the Lyapunov control parameter $V \in (0, 10^7]$. For comparison purpose, we also simulate two baselines as follows: (a) computation resource allocation policy without DNN partitioning point optimization (WDPO), and (b) DNN partitioning point optimization and computation resource allocation policy without reputation based consensus mechanism (WTCM). First, it can be found that the average system latency of the proposed DPRA decreases as V increase. It reveals that a larger value of V can lead to a smaller system latency, which conforms to **Theorem ??**. Second, DPRA shows a lower system latency than baseline schemes. By jointly optimizing the DNN partitioning point and the computation frequency in each time slot, DPRA achieves a significant 91% reduction in system latency compared to WDPS. Compared with WTCM, DPRA adjusts the block generation difficulty according to the off-chain reputation, which reduces the system latency while guaranteeing the trustworthiness of the B-DT system.

References

1 Z. Yang, W. Bao, D. Yuan, N. H. Tran, and A. Y. Zomaya, "Federated learning with nesterov accelerated gradient," *IEEE Trans. Parallel Distributed Syst.*, vol. 33, no. 12, pp. 4863–4873, 2022.
2 C. Zhou, J. Gao, M. Li, N. Chen, X. S. Shen, and W. Zhuang, "Digital-twin-based 3-d map management for edge-assisted device pose tracking in mobile AR," *IEEE Internet Things J.*, vol. 11, no. 10, pp. 17 812–17 826, 2024.
3 J. Zheng, T. H. Luan, Y. Hui, Z. Yin, N. Cheng, L. Gao, and L. X. Cai, "Digital twin empowered heterogeneous network selection in vehicular networks with knowledge transfer," *IEEE Trans. Veh. Technol.*, vol. 71, no. 11, pp. 12 154–12 168, 2022.

4 D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.
5 A. Goldsmith, *Wireless Communications*. Cambridge University Press, 2005.
6 S. R. Pokhrel and J. Choi, "Federated learning with blockchain for autonomous vehicles: Analysis and design challenges," *IEEE Trans. Commun.*, vol. 68, no. 8, pp. 4734–4746, 2020.
7 Y. Qu, L. Gao, T. H. Luan, Y. Xiang, S. Yu, B. Li, and G. Zheng, "Decentralized privacy using blockchain-enabled federated learning in fog computing," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5171–5183, 2020.
8 S. Hu, Y. Pei, and Y. Liang, "Sensing-mining-access tradeoff in blockchain-enabled dynamic spectrum access," *IEEE Wirel. Commun. Lett.*, vol. 10, no. 4, pp. 820–824, 2021.
9 X. Deng, J. Li, C. Ma, K. Wei, L. Shi, M. Ding, W. Chen, and H. V. Poor, "Blockchain assisted federated learning over wireless channels: Dynamic resource allocation and client scheduling," *IEEE Trans. Wirel. Commun.*, vol. 22, no. 5, pp. 3537–3553, 2023.
10 X. Deng, J. Li, L. Shi, Z. Wei, X. Zhou, and J. Yuan, "Wireless powered mobile edge computing: Dynamic resource allocation and throughput maximization," *IEEE Trans. Mob. Comput.*, vol. 21, no. 6, pp. 2271–2288, 2022.
11 M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*, ser. Synthesis Lectures on Communication Networks. Morgan & Claypool Publishers, 2010.
12 Y. Fu, H. Mei, K. Wang, and K. Yang, "Joint optimization of 3d trajectory and scheduling for solar-powered UAV systems," *IEEE Trans. Veh. Technol.*, vol. 70, no. 4, pp. 3972–3977, 2021.
13 M. Hua, Y. Wang, Q. Wu, H. Dai, Y. Huang, and L. Yang, "Energy-efficient cooperative secure transmission in multi-uav-enabled wireless networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7761–7775, 2019.
14 Q. Wu, Y. Zeng, and R. Zhang, "Joint trajectory and communication design for multi-uav enabled wireless networks," *IEEE Trans. Wirel. Commun.*, vol. 17, no. 3, pp. 2109–2121, 2018.