Power Efficient Edge-Cloud Cooperation by Value-Sensitive Bayesian Attractor Model

Tatsuya Otoshi
Graduate School of Economics
Osaka University
Osaka, Japan
t-otoshi@econ.osaka-u.ac.jp

Hideyuki Shimonishi
Cyber Media Center
Osaka University
Osaka, Japan
shimonishi.cmc@osaka-u.ac.jp

Tetsuya Shimokawa, Masayuki Murata
Graduate School of Information Science and Technology
Osaka University
Osaka, Japan
{t-shimokawa, murata}@ist.osaka-u.ac.jp

***Abstract*—Edge computing has emerged as a critical paradigm in AI and IoT applications, offering reduced latency and improved efficiency over traditional cloud-based systems. However, the limited computational resources at the edge and the need for energy-efficient operations pose significant challenges. In this paper, we propose a novel approach for edge-cloud cooperation using the Value-Sensitive Bayesian Attractor Model (VSBAM). Our method focuses on optimizing resource allocation and decision-making processes in edge computing environments, taking into account the computational constraints and power consumption requirements of both edge and cloud systems. Although the optimization of resource allocation is problematic in terms of its computation time, VSBAM reduces the computation time by performing the computation in a distributed manner for each session. Also, for the convergence problem in a distributed system, VSBAM can converge quickly by finding a quasi-optimal solution in a short period of time based on user values. Through simulation-based evaluations, we demonstrate that our approach can significantly reduce power consumption while maintaining high performance, especially in scenarios involving numerous sessions. Our findings also show that our decentralized control strategy is robust against power model errors and performs comparably to centralized control methods.**

***Index Terms*—Bayesian Attractor Model, Value-Sensitive, Edge-Cloud Cooperation**

I. Introduction

Edge computing, which processes information near terminal devices, has gained significant attention, particularly in AI technologies as highlighted [1]. This approach is especially relevant for latency-sensitive tasks like teleoperation, where traditional cloud-based processing suffers from considerable communication delays [2]. However, deploying large-scale AI models directly on edge devices poses challenges due to the limited computational resources available at the edge compared to those in cloud environments. The technique of AI distillation, as introduced by [3], offers a solution by compressing large AI models into smaller, more manageable versions. This adaptation enables the execution of AI tasks on edge computers, which typically have fewer computational resources. Nevertheless, it's acknowledged that these distilled, smaller models often exhibit reduced accuracy compared to their larger counterparts. Moreover, power consumption is a critical factor in practical applications, necessitating careful consideration of the varying power consumption profiles of different computing platforms. Therefore, it's crucial to balance the distribution of tasks across edges and cloud systems, taking into account the interplay between computational capacity, model accuracy, and energy efficiency.

Several methods have been proposed for the collaboration between edge and cloud systems. A typical approach involves dividing the roles of learning and inference. Due to the greater resource requirement for learning compared to inference, learning is performed in the cloud, while inference is carried out on the edge. Another method involves splitting the learning data, known as federated learning, where edge devices use local raw data for learning, and the cloud integrates models learned at the edge. This protects the data privacy of each edge. While fewer in number, some studies have also considered the division of roles during inference. In practice, since inference processing is central, this paper focuses on the efficient use of resources through collaboration during inference. Specifically, by dividing inference processing between edge and cloud, we consider saving resources by terminating inference early at the edge.

Collaboration in inference requires switching between edge and cloud processing depending on the situation of each inference session. For instance, in scenarios like image recognition in a crowd, where the use of high-accuracy models is necessary, a significant amount of data needs to be transferred to the cloud if a certain level of accuracy is not maintained at the edge. Conversely, if high accuracy is not required, deploying a lightweight model on the edge can lead to decisions being made almost entirely at the edge, thus saving substantial resources. However, understanding the situation of each session is impractical due to the high observational and synchronization costs. Therefore, it's desirable for each session to independently understand its situation and select the optimal model. Nonetheless, distributed control often makes it challenging to achieve an overall optimal solution due to conflicts with other sessions.

This paper proposes a collaboration method for inference processing between edge and cloud using the Value-Sensitive Bayesian Attractor Model (VSBAM) [4]. Value sensitivity allows humans to make choices when faced with multiple alternatives with different values, and is influenced by the sum of the values of the alternatives (called the magnitude) [5], [6]. This enables individuals to avoid deadlocked choices due to small differences between high-value options or to wait for better choices in the future when only low-value options are available [7]. Such properties are also known to play an essential role in consensus building in group decision-making [8].

This paper makes the following contributions:

- We formulate the problem of reducing power consumption in the collaboration between edge and cloud through the partitioning and placement of AI models.
- We propose the Magnitude Sensitive Bayesian Attractor Model (MSBAM) that captures changes in AI input data as environmental fluctuations, enabling rapid decision-making for adaptation to such changes.
- We demonstrate that solutions obtained with the proposed method can reduce power consumption equivalently to those obtained with an optimization solver, especially in scenarios with numerous sessions, but within a shorter computation time.
- We shows the robustness of the proposed method against errors in the power model.

The remainder of this paper is organized as follows. In Section II, we introduce the system model of edge-cloud cooperation. In Section III, we introduce the VSBAM. In Section IV, we propose the distributed method of edge-cloud cooperation by applying the VSBAM. In section V, we evaluate the behavior of the proposed model. Finally, in section VI, we summarize and discuss future work.

II. Edge-Cloud Cooperation

To effectively operate AI, it is essential to efficiently utilize resources in both edge and cloud computing. AI demands different resource quantities for learning and inference, and the required resources and achievable accuracy vary with model size. Techniques like distillation, quantization, and pruning are used to create variations in model sizes. However, resource availability differs spatially, with smaller resources typically at the edge and larger resources at the cloud. Therefore, deploying different models at the edge and cloud according to resource constraints and collaborating between them enables the maximum utilization of resources in AI operations.

A. Existing Methods of Cloud-Edge Collaboration

Various methods exist for integrating cloud and edge computing [9]–[11]. One method involves dividing roles between cloud and edge from the perspective of learning. Generally, as learning demands more resources than inference, learning is conducted in the cloud, while inference is done at the edge. Another method involves role division based on the data used for learning. A prominent example is federated learning, where edges use local data for learning, and the cloud integrates models learned at edges. This approach protects data privacy at each edge. Inference can also be divided between cloud and edge, where the AI model is split into a front part processed at the edge and a rear part at the cloud. Particularly, if conclusions can be drawn from intermediate results of the front part, inference can be done with lightweight processing at the edge. Another method involves splitting the amount of data processed between edge and cloud, offloading to the cloud when edge resources are insufficient. The suitability of these collaboration methods varies with the application, and they can often be combined for optimal use.

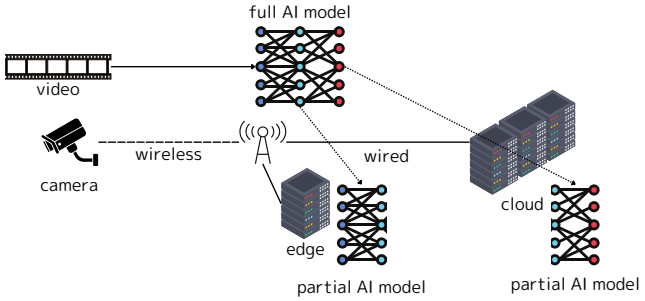


Fig. 1. System Model

B. System Model

This paper focuses on cloud-edge collaboration in the AI inference phase, particularly targeting AI for anomaly detection in surveillance cameras. Fig. 1 shows the system model. Video data acquired by surveillance cameras is input to the AI, and the AI detects abnormalities such as the presence of suspicious persons. In anomaly detection, multi-stage determinations can efficiently utilize resources while improving accuracy. If clear anomalies are detected in the initial stages, subsequent stages can be skipped to save resources. We propose placing initial and subsequent recognition models at the edge and cloud, respectively, and making decisions at the edge alone if accuracy in the initial stage is sufficiently high. The data is assumed to arrive at fixed intervals, and processing times at the edge and cloud are expected to fit within certain time constraints. Placing a larger model at the edge can conclude most processing there, skipping cloud processing, but would always require significant edge processing. Conversely, a smaller model at the edge lightens its load but reduces the probability of skipping cloud processing. The optimal model placement depends on the current data situation; for instance, in crowded or complex scenes, a low-accuracy model at the edge may necessitate transferring almost all data to the cloud. Thus, switching models according to data conditions is necessary.

C. Formulation as an Optimization Problem

The efficiency of resource usage is evaluated by the total power consumption P. The P includes the load-independent power P_{const} and the load-dependent power. Load-dependent power includes power consumed by computers and networks. Each is modeled as proportional to the number of flops processed and the amount of data. They depend on how the model is partitioned (m, m'). The actual power consumption and the model have errors, but we will consider the impact of these errors in the evaluation section. For data x offloaded to the cloud, in addition to the power consumption of the edge, the power consumption of the cloud and the power consumption of the network in transferring the data to the cloud are included. For data that is not offloaded, only edge power consumption varies load-dependently. Similarly for latency, processing latency at the edge and in the cloud, as well as transmission latency in the network are considered. To ensure real-time performance, constraints are set so that

latency is below a certain level. Also, to guarantee the system, we constrain the confidence $A_e(x, m)$ of inference by the edge AI on the data to be above a certain value.

The problem is formulated to find a model placement that satisfies this, along with latency constraints.

$$\begin{aligned} \text{minimize} : P &= \sum_{(x,s)} \{P_e(x_s, m_s) \\ &\quad + c(x_s, m_s)(P_c(x_s, m'_s) \\ &\quad + P_{ec}(x_s, m_s))\} + P_{const} \end{aligned} \tag{1}$$

$$\begin{aligned} s.t. : D(x_s, m_s) &= D_e(x_s, m_s) \\ &\quad + c(x_s, m_s)(D_c(x_s, m'_s) \\ &\quad + D_e c(x_s, m_s)) < D \end{aligned} \tag{2}$$

$$c(x_s, m_s) = 1_{a:a<A}(A_e(x_s, m_s)) \tag{3}$$

where s is the session, x is the input video, m and m' are the initial and subsequent models, P_e, P_c, and P_{ec} are the power consumption of edge, cloud, and network respectively; D_e, D_c, and D_{ec} are the latencies of edge, cloud, and network respectively; D is the upper limit of latency constraints; A is the lower limit of accuracy constraints; and $c(x, m)$ indicates whether the cloud is utilized or not, with $1_X(x)$ being the indicator function.

III. Value-Sensitive Bayesian Attractor Model (VSBAM)

In this chapter, we propose a new model of decision-making that is magnitude sensitive and highly applicable to engineering. First, we introduce the basic model, BAM. Our research group has applied BAM to various network control problems, and BAM is a model with high applicability. However, since the original BAM is a feature-based classification model, which is different from the value-based decision-making targeted in this paper, we introduce an extended BAM for value-based decision-making. We then introduce a value-sensitive model.

A. Value-based BAM

BAM [12] is a model of brain decision-making that involves the process of updating internal states based on observations, and making decisions based on the updated internal states.

The BAM models the process of reaffirmation, in which features are used as input and representative values are updated by comparing them to representative values bound to a previously given choice, called an attractor. The specific state update is performed by Bayesian updating based on observed values, using the relationship between attractors and representative values, and the endogenous dynamics of the state as a generative model. The generative model is as follows

$$\boldsymbol{z_t} = f(\boldsymbol{z_{t-1}}) + q\boldsymbol{w_t} \tag{4}$$

$$\boldsymbol{x_t} = M\sigma(\boldsymbol{z_t}) + s\boldsymbol{v_t} \tag{5}$$

where x_t is the observed value, z_t is the internal state, and w_t, v_t is the noise. Eq. (4) is an expression for the internal variation of the state, where f is the Hopfield dynamics with K attractors $\phi_1, \cdots, \phi_K$. Eq. (5) is an expression for the relation between representative values and attractors, where M is a matrix of representative values μ_i corresponding to the attractor ϕ_i and $M = (\mu_1, \cdots, \mu_K)$. The $\sigma(z_t)$ is an element-wise sigmoid function. Also, q, s are the parameters for the magnitude of each noise term, called dynamic uncertainty and sensory uncertainty.

In [4], we introduce the value-based decisions in the BAM. Let $V_{i,t}$ be the value of choice ϕ_i at time t, and let BAM obtain a value estimated value $\bar{v}_{i,t}$ through reward feedback information. This sequence of value estimates is the information that BAM can observe at time t, and the observed value as value is defined as follows.

$$\boldsymbol{x}_t = (\bar{v}_{1,t}, \cdots, \bar{v}_{K,t}) \tag{6}$$

In value-based decision-making, we need to find the highest-value alternative. To handle this in the recognition scheme of BAM, the representative value μ_i of the choice ϕ_i is the observed value such that the choice is of maximum value. That is, using the standard value $\bar{v}$ and the maximum value $\bar{v}_{max} = \max\{\bar{v}_{1,t}, \cdots, \bar{v}_{K,t}\}$, we determine the representative value as follows

$$\boldsymbol{\mu}_i = (\bar{v}_0, \cdots, \bar{v}_0, \overset{(i)}{\bar{v}_{max}}, \bar{v}_0, \cdots, \bar{v}_0). \tag{7}$$

where i-th element is $\bar{v}_{max}$ and the other elements are $\bar{v}_0 = 0$.

Given observed value, each confidence $P(\boldsymbol{z}_t = \phi | x_t)$ is calculated.

B. Value Sensitivity

Several models have been proposed that are sensitive to value considerations. In [4], we have proposed a model that integrates value-based Biologically Adaptive Models (BAM) with the Leaky Competing Accumulator (LCA), as detailed in [5], [13]. This paper also employs this model to coordinate inference processes between edge computing and cloud computing.

In the LCA model, the internal state is updated by a drift term and a noise term. The value magnitude sensitivity is expressed by varying the drift term in a stimulus size-dependent manner. Specifically, the internal state X_i for each option is updated according to the following equation.

$$X_{i,t+1} = I_i(t) + (1 - \gamma)X_{i,t} - \beta \sum_{j \neq i} X_j \tag{8}$$

where $I_i(t)$ represents the stimulus magnitude, and γ, β are the parameters. The first term indicates that the more valuable the choice, the larger the movement of the state. The second term indicates self-activation, and the third term indicates suppression of other choices by the active choice. In the steady state, only one higher-value option is active, similar to BAM.

In value-based BAM, the definition of $M, \boldsymbol{x}_t$ in eq. (5) indirectly causes an internal state update according to the magnitude of the value. However, it is s that controls the effect of Eq. (5) on the state update. The smaller s is, the more strictly $\boldsymbol{x}$ follows the equation, and the more deterministically $\boldsymbol{x}$ updates in the Bayesian update.

Therefore, by varying s as a function of value magnitude, the value-based z update can be more directly controlled. In

this paper, s is varied by the reciprocal of the magnitude as follows.

$$s = \frac{s_0}{V_t} \tag{9}$$

where s_0 is the reference sensory uncertainty and $V_t = \sum_i \bar{v}_i$ is the magnitude. Assume $s_0 = 1$ unless otherwise noted.

IV. Cloud-Edge Cooperation by VSBAM

This chapter presents a method for solving the cloud-edge collaboration problem using the Value-Sensitive Bayesian Attractor Model (VSBAM). Specifically, we map the attractors and values of VSBAM to the cloud-edge collaboration problem and propose a method to compute the distribution of models for each session using VSBAM, enabling a decentralized approach to solving the problem.

A. VSBAM-Based Solution for Cloud-Edge Collaboration Problem

1) Attractors: We define the combination of model divisions across all sessions (m_s, m'_s) as attractors. The division for each session is determined by the layers or components of the AI model.

2) Values: The value of an attractor is determined by the total power consumption of the model division it represents. This value is updated according to observed results. As lower power consumption is desired and considered better, the value is set as the inverse of the total power consumption. P_{max} is a constant to ensure the value remains positive:

$$F(a) = P_{max} - P \tag{10}$$

3) Introducing Latency Constraints: Since VSBAM cannot directly handle constraints, we introduce latency constraints into the value of the attractors. Attractors that violate latency constraints are penalized:

$$F_d(a) = F(a) - \lambda C \tag{11}$$

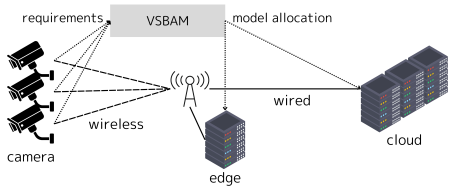
$$C = \sum_s \max\{D(x_s, m_s) - D, 0\} \tag{12}$$

B. Decentralized VSBAM-Based Solution for Cloud-Edge Collaboration Problem

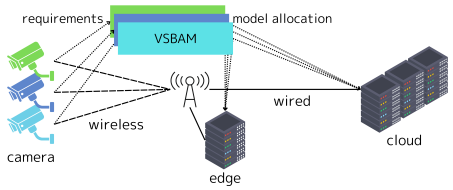
While the above method centrally determines the division of models across all sessions, considering the cost of information gathering, computation, and synchronization, it is more efficient to determine the model division for each session in a decentralized manner. Therefore, we propose a method to compute the model division for each session using VSBAM. Fig 2 shows the VSBAM with centralized and decentralized manner. In the centralized VSBAM, attractors represent the model division choices for all sessions. In the decentralized VSBAM, attractors represent the model division choices for each session (m_s, m'_s). For each session, VSBAM runs on the terminal connected to the camera. Power consumption is calculated in the same way as in the centralized approach by obtaining the consumption of the used edge and cloud. For latency constraints, only the delay of the session itself is considered:

$$F_{dd}(a) = F(a) - \lambda C_s \tag{13}$$

$$C_s = \max\{D(x_s, m_s) - D, 0\} \tag{14}$$



(a) Centralized



(b) Decentralized

Fig. 2. Centralized VSBAM and Decentralized VSBAM

V. Evaluation

To verify the effectiveness of the proposed method, we conduct a simulation-based evaluation.

A. Simulation Environment

Our simulations are based on a model of power consumption [14], which is based on actual measurements. We assume a scenario where each session utilizes a common edge node and a common cloud node. The edge node is assumed to have a maximum computing resource of 16200 GFLOPS $\times$ 45% (NVidia RTX 3060 Ti), and the cloud node 27770 GFLOPS $\times$ 45% (NVidia RTX A5000). The edge node has a fixed consumption of 121.66W and 220W per unit load, while the cloud node has 121.66W fixed and 250W per unit load. Network propagation delay is set at 10ms, with an edge-cloud bandwidth of 1000Mbps. The network incurs power consumption proportional to the data size transferred from edge to cloud, with 0.06W per Mbit. The robustness of the proposed method against errors is also evaluated by simulation, considering the case where such a model of power consumption deviates from the actual power consumption.

For the AI model, it is assumed that each session processes a total of 7 GFLOPS [15]. The load allocation between edge and cloud is proportional to the division ratio of the model. The division ratio at the edge varies from 0 to 100% in increments of 20%, with the remainder processed by the cloud. The accuracy of AI ranges from 0 to 0.9 depending on the size of the model.

AI decisions are made 15 times per second, setting a latency constraint of $1/15$ second. Data has a difficulty level ranging from 0 to 1, and it is assumed that if the difficulty is below the

AI's accuracy, a confident decision can be made. The difficulty level for each data follows a Dirichlet distribution with $\alpha = (10, 10)$.

B. Solver Settings

To evaluate the performance of solutions derived by the proposed method, comparisons are made with solutions obtained using a solver. The solver uses Bayesian optimization via Optuna. The expected value of power consumption is calculated over 100 timesteps per trial, with a total of 100 trials, thus optimizing over 10,000 timesteps of data.

C. Results

1) Centralized v.s. Decentralized: The proposed method can solve distributed cloud-edge collaboration problems by determining the division of the model for each session. Decentralized optimization, compared to centralized optimization, can reduce the costs of information gathering, computation, and synchronization, but may potentially lead to lower optimization performance. To evaluate the performance of decentralized optimization in the proposed method, a comparison with centralized optimization is presented. Figure 3 shows the timeseries of total power consumption and delay constraint violations for centralized and decentralized optimization methods, with the number of sessions set to 10. The figure reveals that there is no significant difference in performance between centralized and decentralized optimization. Power consumption is slightly lower in the centralized method, but the difference is minor. Additionally, violations of delay constraints are resolved similarly in both methods within a short period. Thus, it is evident that decentralized control can perform comparably to centralized control.

2) Robustness to Power Consumption Noise: The proposed method estimates the power consumption of each node using a power model, but actual node power consumption may differ from the model. To evaluate the robustness of the proposed method to power consumption noise, performance under added noise to node power consumption is assessed. Specifically, Gaussian noise is added to the computed power consumption with a mean of zero and standard deviations of 0%, 10%, 20%, and 30% of the original power consumption. Since the total power consumption is around 250W, the standard deviations are 0, 25, 50, and 75W, respectively.

Figure 4 shows the distribution of total power consumption and delay constraint violations for various standard deviations of noise, with the number of sessions set to 10. The number of delay constraint violations is counted on the timestep, and the maximum is 100 timesteps. The figure indicates that as noise increases, violations of delay constraints also increase due to the deviation of estimated power consumption from actual values, leading to the selection of solutions that break delay constraints. However, the increase in violations is gradual, and even with 30% error, the average number of violations per session is less than one. Moreover, the impact of noise on power consumption is minimal. At 30% error, slightly higher power-consuming solutions are selected, but the degree is small. These findings demonstrate the robustness of the proposed method to variations in the power consumption model.

According to the literature [14], the error between the power consumption on the model and the actual measured power consumption is less than 10%. This method is robust enough to handle larger errors, so there should be no problem in practical use.

3) Comparison with Solver Results: To verify the effectiveness of the proposed method in minimizing power consumption, a comparison with solutions derived from a solver is conducted. Figure 5 illustrates the timeseries of power consumption for varying numbers of sessions. The red line means the expected value by solver. As the number of sessions increases, so does the computational load on edge-clouds, leading to higher overall power consumption. However, appropriate distribution between edge and cloud can significantly reduce the potential power consumption. In practice, accommodating numerous sessions and maintaining a high load rate is economically advantageous, making the power performance in scenarios with many sessions crucial. The results from the figure indicate that as the number of sessions increases, the proposed method approaches the solver's solution, particularly showing equivalent performance with approximately 200 sessions. This suggests that the proposed method can perform near-optimally in realistic scenarios. Moreover, the proposed method is computationally more efficient as it optimizes at every timestep, whereas the solver uses data for 100x100 timesteps for optimization. The computational time required to obtain one solution was 0.16 seconds for the proposed method, while the solver required about one hour.

VI. Conclusion

In this paper, we presented a novel approach for edge-cloud cooperation using the VSBAM. By employing the VSBAM, we were able to make distributed decisions that optimize power consumption while considering the delay constraints of each session. Our simulation-based evaluations demonstrated that the proposed method could achieve power savings as well as offline optimization solver, especially in scenarios with numerous sessions. We also show that the proposed method is robust against power model errors, and that distributed implementation can achieve nearly the same level of power consumption reduction as centralized systems.

Future work includes the evaluation with dynamic environments, such as the arrival of new sessions and the departure of existing sessions.

Acknowledgment

This work was supported by MIC under a grant entitled "R&D of ICT Priority Technology (JPMI00316)".

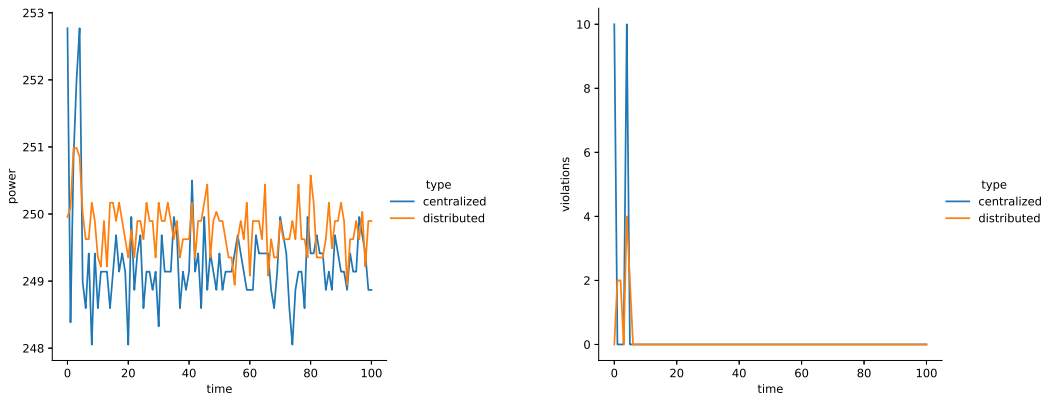
References

[1] J. Chen and X. Ran, "Deep learning with edge computing: A review," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1655–1674, 2019.

[2] S. K. Sharma, I. Woungang, A. Anpalagan, and S. Chatzinotas, "Toward tactile internet in beyond 5g era: recent advances, current issues, and future directions," *Ieee Access*, vol. 8, pp. 56 948–56 991, 2020.

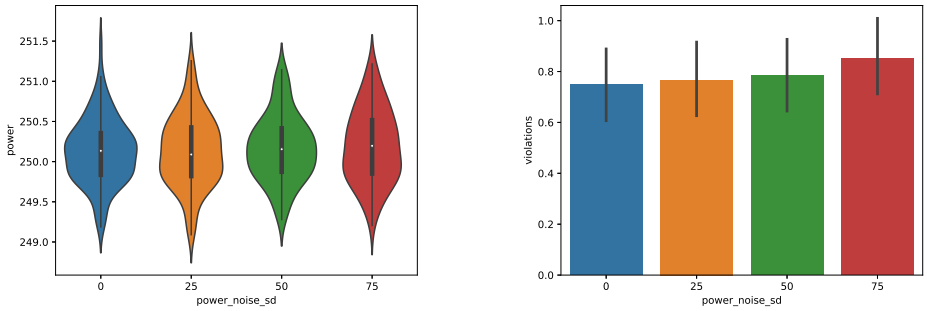
[3] G. Hinton, O. Vinyals, J. Dean *et al.*, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, vol. 2, no. 7, 2015.

[4] T. Otoshi, M. Murata, H. Shimonishi, and T. Shimokawa, "Distributed timeslot allocation in mMTC network by magnitude-sensitive Bayesian attractor model," in *2023 IEEE 9th International Conference on Network Softwarization (NetSoft)*. IEEE, 2023, pp. 212–216.



(a) Total Power (b) Delay Constraints Violations

Fig. 3. Timeseries of Total Power and Delay Constraints Violations for Centralized and Decentralized Methods



(a) Average Total Power (b) Delay Constraints Violations

Fig. 4. Total Power and Delay Constraints Violations with Various Noise Levels

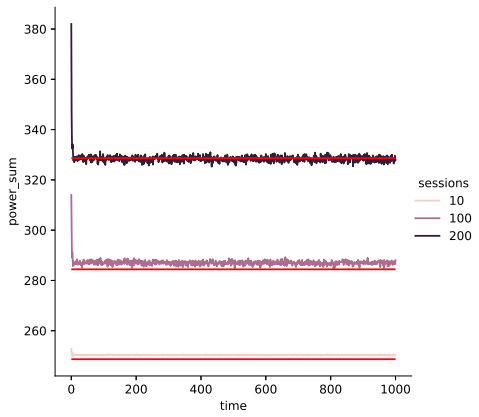


Fig. 5. Timeseries of Power Consumption with Various Numbers of Sessions

[5] R. Ratcliff, C. Voskuilen, and A. Teodorescu, "Modeling 2-alternative forced-choice tasks: Accounting for both magnitude and difference effects," *Cognitive Psychology*, vol. 103, pp. 1–22, 2018.

[6] A. Pirrone, A. Reina, T. Stafford, J. A. Marshall, and F. Gobet, "Magnitude-sensitivity: rethinking decision-making," *Trends in Cognitive Sciences*, vol. 26, no. 1, pp. 66–80, 2022.

[7] S. Tajima, J. Drugowitsch, and A. Pouget, "Optimal policy for value-based decision-making," *Nature communications*, vol. 7, no. 1, pp. 1–12, 2016.

[8] D. Pais, P. M. Hogan, T. Schlegel, N. R. Franks, N. E. Leonard, and J. A. Marshall, "A mechanism for value-sensitive decision-making," *PLOS ONE*, vol. 8, no. 9, p. e73216, 2013.

[9] J. Yao, S. Zhang, Y. Yao, F. Wang, J. Ma, J. Zhang, Y. Chu, L. Ji, K. Jia, T. Shen *et al.*, "Edge-cloud polarization and collaboration: A comprehensive survey for ai," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 7, pp. 6866–6886, 2022.

[10] Y.-C. Wang, J. Xue, C. Wei, and C.-C. J. Kuo, "An overview on generative ai at scale with edge-cloud computing," 2023.

[11] D. Liu, X. Chen, Z. Zhou, and Q. Ling, "Hiertrain: Fast hierarchical edge ai learning with hybrid parallelism in mobile-edge-cloud computing," *IEEE Open Journal of the Communications Society*, vol. 1, pp. 634–645, 2020.

[12] S. Bitzer, J. Bruineberg, and S. J. Kiebel, "A Bayesian attractor model for perceptual decision making," *PLoS Computational Biology*, vol. 11, no. 8, p. e1004442, 2015.

[13] A. R. Teodorescu, R. Moran, and M. Usher, "Absolutely relative or relatively absolute: violations of value invariance in human decision making," *Psychonomic Bulletin & Review*, vol. 23, no. 1, pp. 22–38, 2016.

[14] H. Shimonishi, M. Murata, G. Hasegawa, and N. Techasarntikul, "Energy optimization of distributed video processing system using genetic algorithm with bayesian attractor model," in *2023 IEEE 9th International Conference on Network Softwarization (NetSoft)*. IEEE, 2023, pp. 35–43.

[15] M. Agarla, P. Napoletano, and R. Schettini, "Quasi real-time apple defect segmentation using deep learning," *Sensors*, vol. 23, no. 18, p. 7893, 2023.