2025 34th International Conference on Computer Communications and Networks (ICCCN) | 979-8-3315-0898-2/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICCCN65249.2025.11133993

Bayesian Deep Neural Network-empowered Thompson Sampling for Context-aware Task Offloading in Dynamic Fog Computing

Hoa Tran-Dang
Dept. of IT convergence engineering
Kumoh National Institute of Technology
Gumi-si, South of Korea
{hoa.tran-dang}@kumoh.ac.kr

Dong-Seong Kim
Dept. of IT convergence engineering
Kumoh National Institute of Technology
Gumi-si, South of Korea
dskim@kumoh.ac.kr

***Abstract*—Efficient task offloading in dynamic fog computing environments requires adaptive decision-making under uncertainty. This paper proposes a Bayesian Deep Neural Network (BDNN)-empowered Thompson Sampling (TS) framework for context-aware task offloading, enabling intelligent resource allocation while balancing exploration and exploitation. The BDNN models the stochastic reward function by learning a posterior distribution over network weights, capturing the uncertainty in offloading decisions. At each time step, the task node samples a set of weights from the learned posterior to estimate the expected reward of each helper node, facilitating adaptive decision-making in dynamic network conditions. Experimental results demonstrate that our approach outperforms conventional heuristics and deep learning-based methods, achieving lower latency, improved resource utilization, and better offloading efficiency in fog computing environments.**

***Index Terms*—Bayesian Deep Neural Network (BDNN), Thompson Sampling (TS), Context-aware Task Offloading, Dynamic Fog Computing, Uncertainty-aware Decision Making.**

I. Introduction

In recent years, fog computing has emerged as a promising paradigm to address the increasing demand for low-latency, resource-intensive applications by extending cloud computing capabilities to the edge of the network. Fog computing enhances the performance of Internet of Things (IoT) applications by enabling data processing, storage, and computation closer to the devices generating the data [1]. However, effective task offloading in fog computing is a complex challenge due to the dynamic nature of the network, varying computational resources of fog nodes, fluctuating network conditions, and diverse application requirements.

Task offloading refers to the process of delegating tasks from resource-constrained devices (task nodes) to more powerful computational resources (fog nodes or helper nodes) for execution. Efficient task offloading strategies can significantly improve resource utilization and overall system performance [2]. However, task nodes often have limited knowledge about the current state of the fog nodes (e.g., available resources, latency, and network congestion), which complicates the decision-making process for task offloading.

Traditional approaches to task offloading rely on predefined rules or heuristic-based methods to decide the best fog node for task execution. These methods often fall short when dealing with the uncertainty and dynamic nature of fog environments. Machine learning techniques, such as Deep Neural Networks (DNNs), have been employed to predict the reward of offloading decisions based on historical data [3]. However, these approaches typically rely on deterministic models, which do not explicitly account for uncertainty in the reward estimation. The inability to model uncertainty leads to suboptimal decisions, particularly in highly dynamic environments.

To overcome this limitation, we propose a novel framework that combines Bayesian Deep Neural Networks (BDNNs) with Thompson Sampling (TS) for context-aware task offloading in dynamic fog computing environments. The core idea of our approach is to leverage BDNNs to model the reward function as a probabilistic distribution, capturing the inherent uncertainty in offloading decisions. Instead of using a single deterministic estimate, our approach samples weights from the posterior distribution of the BDNN at each time step, allowing for adaptive decision-making that balances exploration and exploitation.

In our framework, each task node observes a variety of contextual information, such as task characteristics (e.g., size, deadline, computation requirements), network conditions (e.g., congestion, latency), and fog node features (e.g., available resources). Using this context, the BDNN estimates the expected reward for offloading the task to different fog nodes. Thompson Sampling is then used to sample the reward from the learned posterior distribution, which enables the task node to explore different offloading strategies while exploiting the most promising ones.

The key contributions of this paper are as follows:

- We propose a BDNN-based model for uncertainty-aware task offloading, where the reward function is learned probabilistically.
- We integrate TS with the BDNN to enhance exploration-exploitation tradeoffs during task offloading decisions.

- We demonstrate the effectiveness of our approach through extensive simulations and comparative experiments, showing its ability to achieve energy efficiency, lower latency, and successful offloading rate compared to existing methods.

II. Related Works

Task offloading in fog computing has attracted significant attention due to its potential to enhance the performance of latency-sensitive applications. Traditional offloading approaches often rely on *heuristic-based* or rule-based strategies. For example, methods that select the fog node with the lowest latency or the highest available resources have been explored, but these approaches tend to fail in dynamic environments where network conditions and task requirements fluctuate unpredictably. As a result, these heuristic methods are typically not adaptable to changing scenarios and often lead to inefficient resource usage.

To enhance adaptability, Reinforcement Learning (RL) has been proposed for task offloading in fog networks [4]. Q-learning was one of the first RL techniques applied to this problem, where the offloading decision was modeled as a sequential decision process [5]. However, Q-learning struggles with large state spaces and extended training times, particularly in the context of large-scale networks. More advanced methods like *Deep Q-Networks (DQNs)* have been employed to overcome these limitations by approximating the Q-function with deep neural networks. [6] applied DQNs to predict the most efficient fog node for task offloading, showing improvements in decision-making. However, these methods still rely on deterministic models, which fail to account for the inherent uncertainty present in the reward function.

Deep Reinforcement Learning (DRL) techniques have been further explored to enhance the decision-making process in dynamic fog networks. Techniques such as Advantage Actor-Critic (A2C) and Proximal Policy Optimization (PPO) [7], [8] have shown promising results by leveraging policy-based learning to optimize offloading decisions. DRL-based approaches provide flexibility in handling continuous state spaces and dynamic environments. However, these methods often suffer from high training complexity and convergence instability, especially in non-stationary environments.

Multi-Agent Reinforcement Learning (MARL) has also been explored in multi-node environments, where each task node learns to make decisions by interacting with other task nodes and the fog infrastructure. [9] introduced a MARL-based approach where multiple task nodes cooperate to optimize resource allocation in fog networks. This work demonstrated the ability to handle more complex interactions between task nodes and fog nodes. However, these approaches often face challenges in managing the uncertainty and variability in network conditions, which are critical in dynamic fog environments.

In response to the need for uncertainty modeling, Bayesian methods have been employed to address reward uncertainty in task offloading. The work [10] explored Bayesian optimization for offloading decisions in fog computing, providing a probabilistic framework for reward modeling. While Bayesian optimization is effective in static settings, it faces limitations in real-time decision-making in dynamic environments. To improve upon these methods, BNNs have been integrated with RL techniques. The authors in [11] demonstrated the use of BNNs in dynamic task offloading, where the posterior distribution over model parameters provided an estimate of the uncertainty. However, these works often do not fully leverage the exploration-exploitation tradeoff that TS offers.

Recent studies have explored bandit learning-based approaches for task offloading in fog-enabled networks. The BLOT algorithm [12] employs a bandit learning model to optimize task offloading under non-stationary environments, minimizing long-term costs by considering latency, energy consumption, and switching costs. The study demonstrates the asymptotic optimality of the BLOT algorithm in dynamic settings, showcasing its effectiveness in real-world applications. Similarly, [13] proposed the D2CIT scheme, which leverages a multi-armed bandit model for decentralized task offloading in IoT-enabled fog networks. By dynamically selecting fog nodes for task allocation based on an -greedy policy, D2CIT significantly reduces latency and energy consumption while optimizing resource utilization. Comparative results indicate that D2CIT achieves a 17% reduction in latency and a 59% speedup compared to conventional fog computing approaches.

These bandit learning-based solutions provide strong adaptability to changing network conditions, addressing the limitations of traditional RL methods. Unlike conventional RL models that require extensive training, bandit-based strategies effectively balance exploration and exploitation, leading to improved decision-making under uncertainty. Consequently, integrating Bayesian methods with Thompson Sampling presents a promising direction for optimizing task offloading in dynamic fog computing environments.

III. System Model and Problem Formulation

A. System Model

1) Dynamic FCN: We consider a FCN consists of a TN and M HNs, enabling decentralized and low-latency task computation as illustrated in Fig. 1.

TN generates computational tasks, while HNs, such as fog servers, assist in processing. This framework is particularly applied in real-time applications with resource constraints. A motivating example is Simultaneous Localization and Mapping (SLAM) [12], where a mobile robot builds a map and tracks its position in real time. Offloading SLAM tasks—such as feature extraction and pose estimation—to nearby fog nodes reduces latency and improves efficiency. Given the robot's lack of prior knowledge about available resources, dynamic exploration and real-time offloading are necessary. While this work focuses on a simplified network, the proposed algorithm can be extended to scalable networks with multiple agents and hierarchical fog layers, which will be explored in future work. Let $\mathcal{H} = \{1, 2, ..., H\}$ represent the set of helper

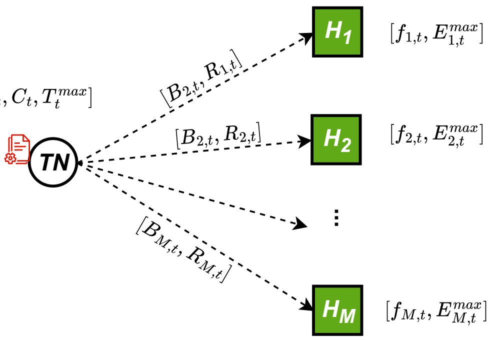

Fig. 1: Task offloading problem between a TN with a generated task and multiple HNs with heterogeneous and time-varying computing resource.

nodes. The network operates in T time slots indexed by $t \in \mathcal{S} = \{1, 2, ..., T\}$, where system conditions, such as task arrivals, network states, and resource availability, evolve dynamically.

The TN generates a computational task at time slot t characterized by $\tau_t = \{D_t, C_t, T_t^{max}\}$, where D_t is the task size data, C_t is the required CPU cycles per bit, and T_t^{max}, which specifies the maximum time constraint for task completion. These task parameters change over time, reflecting variations in workload demands.

Each HN $H_i \in \mathcal{H}$ possesses computing resources that are also time-varying. Specifically, the computational capability of HN i at time t is denoted as $f_{i,t}$, representing the available CPU cycles per second. In addition, the communication bandwidth between TN and HN j is denoted as $B_{i,t}$, which defines the achievable data transmission rate. Due to the dynamic nature of the fog network, these resources fluctuate over time as multiple tasks compete for computational and network resources.

At each time slot, TN must make offloading decisions, selecting HNs to execute their tasks while considering communication and computation constraints.

2) Offloading Delay Model: The task execution time when offloading from TN to HN i consists of transmission time $T_{i,t}^{comm}$ and computation time $T_{i,t}^{comp}$, which are expressed by $T_{i,t}^{comm} = \frac{D_t}{R_i^t}, T_{i,t}^{comp} = \frac{D_t C_t}{f_{i,t}}$, where $R_{i,t} = B_{i,t} \log_2(1+\gamma_{i,t})$ is the achievable data rate, and $\gamma_{i,t}$ is the signal-to-noise ratio.

3) Energy Consumption Model: The energy consumption for transmission and computation are expressed by $E_{i,t}^{\text{comm}} = P_t \frac{D_t}{R_{i,t}}$, $E_{i,t}^{\text{comp}} = \kappa(f_{i,t})^2 \frac{D_t C_t}{f_{i,t}}$, respectively, where P_t is the transmission power of TN, and κ is a hardware-dependent coefficient.

The total cost for offloading the task to HN i is modeled as $U_{i,t} = (T_{i,t}^{total} + \gamma E_{i,t}^{\text{total}})$, where $T_{i,t}^{total} = T_{i,t}^{comm} + T_{i,t}^{comp}$, $E_{i,t}^{\text{total}} = E_{i,t}^{\text{comm}} + E_{i,t}^{\text{comp}}$, and γ is a trade-off parameter.

B. Problem Formulation

The objective is to minimize the total offloading cost subject to the task deadline and energy consumption constraints. Definitely, the optimization problem can be formulated as:

$$\textbf{P:} \quad \min_{i \in \mathcal{H}} \sum_{t=1}^{T} \mathbb{E}\left[U_{i,t}\right] \tag{1}$$

$$\textbf{s.t:} \quad C_1 : T_{i,t}^{total} \leq T_t^{max}, \tag{2}$$

$$C_2 : E_{i,t}^{comp} \leq E_{i,t}^{max}, \tag{3}$$

where $E_{i,t}^{max}$ is the residual energy at HN H_i at time slot t.

Solving the above defined problem presents several significant challenges. Uncertainty and variability in the dynamic fog environment arise due to fluctuating network conditions and varying resource availability at helper nodes, making it difficult to predict task execution outcomes accurately. Additionally, the lack of an explicit model for the reward function further complicates decision-making, as the function is inherently complex and nonlinear, influenced by dynamic communication and computational environments. Moreover, the computational complexity of real-time offloading decisions is substantial, as the system must evaluate multiple helper nodes simultaneously to identify the optimal offloading strategy. Traditional optimization approaches struggle to handle this challenge efficiently, necessitating learning-based methods capable of adaptive and scalable decision-making.

IV. Proposed Algorithm Design

To address these challenges, we propose using a BDNN-TS to estimate the reward distribution for each HN under uncertainty. BDNN approximates the posterior distribution over model parameters, enabling Thompson Sampling-based exploration. By learning from past interactions, the BDNN dynamically refines its reward predictions, leading to better offloading decisions in real time. Basically, the BDNN consists of input layer, three hidden layer, and an output layer as illustrated in Fig. 2. The input layer receives the concatenated

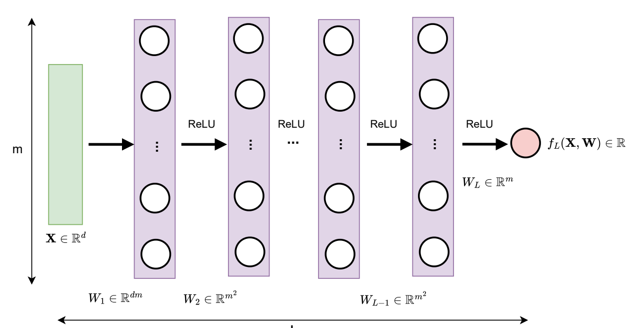

Fig. 2: General architecture of BDNN for estimating the reward of HN

feature vector $\mathbf{X}_{i,t}$ in each time slot to estimate the reward of H_i. The three hidden layers, each with 64, 32, and 16 neurons respectively, use ReLU activation to capture complex interactions among features. In particular, the first hidden layer helps extract high-dimensional representations, and the two subsequent hidden layers reduce dimensionality while preserving task-related features. The output layer represents the estimated mean reward. The task offloading problem P is resumed in Fig. 3.

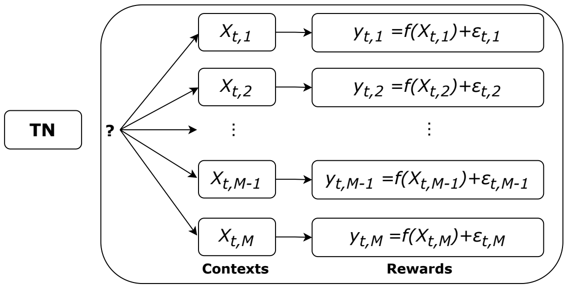

Fig. 3: Contextual bandit learning model with non-linear reward for the online task offloading problem **P**

Algorithm 1 summarizes the procedures to implement out proposed BDNN-TS-based task offloading approach.

At each time slot t, the TN observes the contextual information $\mathbf{X}_{i,t}$ including the task characteristics, the features of H_i, as well as, their communication conditions. Then, $\mathbf{X}_{i,t}$ is served as input for BDNN, which outputs the $y_{i,t}$ as the mean reward of H_i. Since the true reward is unknown and dynamically changing, we sample from the BDNN posterior $\hat{y}_{i,t} \sim \mathcal{N}(y_{(i,t)}, \sigma^2_{(i,t)})$, where $\sigma^2_{(i,t)}$ is the BDNN-estimated uncertainty.

The TN selects the HN $H_t^* = \arg\max_{i \in \mathcal{H}} \hat{y}_{i,t}$ that maximizes the estimated reward at time slot t . Once the offloading decision is executed, the actual reward $y^*_{i,t}$ is observed, and the dataset is updated:

$$\mathcal{D}_{t+1} = \mathcal{D}_t \cup \{(\mathbf{X}_{i,t}, H^*_{i,t}, y^*_{i,t})\}, \tag{4}$$

where the actual reward $y^*_{i,t} = -(T^{total}_{i,t} + \lambda E^{total}_{i,t}) - \beta 1(T^{total}_{i,t} > T^{max}_{i,t})$, $1(T^{total}_{i,t} > T^{max}_{i,t})$ is an indicator function that takes a value of 1 if the task execution time exceeds the deadline and 0 otherwise, and β is a penalty factor applied when the deadline is violated. To refine the model, the BDNN updates its posterior distribution using the new dataset. This involves minimizing a loss function $L(\mathbf{W})$ defined as follow:

$$L(\mathbf{W}) = \frac{1}{2} \sum_{i=1}^{t} \left[f\left(\mathbf{X}_{i,t}; \boldsymbol{W}\right) - y^*_{i,t} \right]^2 + \frac{1}{2} \lambda \left\| \boldsymbol{W} - \boldsymbol{W}_0 \right\|_2^2, \tag{5}$$

where λ is a regularization parameter. The weight vector **W** is updated via the learning rate η as $W_t = W_{t-1} - \eta \nabla L_W(W)$.

Algorithm 1: BDNN-TS for Task Offloading

Input: Features of task, HNs, and network conditions
Output: Optimal HN H_t^* at time slot t

Initialization: $\mathbf{W}_0 \sim \mathcal{N}(0, \mathbf{I})$; $\mathcal{D} = \emptyset$; $\mathbf{U}_0 = \lambda \mathbf{I}$
for $t = 1, 2, \ldots, T$ **do**
  for $i = 1, 2, \ldots, M$ **do**
    Get context: $\mathbf{X}_{(i,t)}$;
    // Compute gradients
    $\mathbf{g}(X_{(i,t)}; W_{(i,t-1)}) = \nabla_W h(X_{(i,t)}; W_{(i,t-1)})$;
    // Compute: $\sigma^2_{(i,t)}$
    $\sigma^2_{(i,t)} = \lambda \mathbf{g}(X_{(i,t)}; W_{(i,t-1)})^\top \mathbf{U}^{-1}_{t-1} \mathbf{g}(\mathbf{X}_{(i,t)}; W_{(i,t-1)})$
    // Compute expected reward
    $y_{(i,t)} = h_L(\mathbf{X}_{(i,t)}, \mathbf{W}_{(i,t-1)})$;
    // Sample reward
    $\hat{y}_{i,t} \sim \mathcal{N}(y_{(i,t)}, \sigma^2_{(i,t)})$
  end
  // Select the best action (offloading decision)
  $H_t^* = \arg\max_{i \in \mathcal{H}} \hat{y}_{i,t}$;
  // Execute offloading and observe actual reward
  Offloadng task to H_t^* & get the true reward $y^*_{(i,t)}$;
  // Update dataset and BDNN weight
  $\mathcal{D} \leftarrow (X_{(i,t)}, H_t^*, y^*_{(i,t)})$;
  // Updage BDNN weight
  $W_t = W_{t-1} - \eta \nabla L_W(W)$;
  // Update covariance matrix
  $\mathbf{U}_t = \mathbf{U}_{t-1} + \mathbf{g}(X_{(t,H_t^*)}; W_t) \mathbf{g}(X_{(t,H_t^*)}; W_t)^\top$
end

V. Simulation Results and Evaluation Analysis

A. Simulation Environment Configuration

To evaluate the performance of the proposed algorithm, we simulate a fog computing environment with a single TN and five HNs. The TN generates a computation task at each time slot which is then offloaded by a HN selected by the proposed BDNN-TS algorithm. The other parameters and corresponding values are listed in Table I.

B. Comparative Algorithms

To assess the performance of the proposed algorithm, we compare it with three baseline multi-armed bandit (MAB) algorithms commonly used for task offloading in fog computing. These include ϵ-Greedy Algorithms, upper confidence bound (UCB). Additionally, we include traditional TS without the BDNN component, to highlight the impact of incorporating deep learning for context-aware decision-making. The key metrics used for comparison include: (i) average offloading delay, which calculates the time taken for task completion after offloading (ii) task offloading success rate, which measures the percentage of tasks successfully offloaded within their deadlines, and (iii) average energy consumption per task.

TABLE I: Simulation Parameters and Values

Parameter	Value
# of TN	1
# of HNs	5
Task Size	0.5 - 5 MB
Task Deadline	0.5 - 2.0 sec
Task complexity	$C_t \sim U(500, 2000)$ cycles/bit
Computation Capacity of HNs	$f_{i,t} \sim U(1, 5)$ GHz
Available Bandwidth of HNs	$B_{i,t} \sim U(5, 20)$ MHz
Transmit power	$P_{i,t} = 100\text{mW}$
Hardware coefficient	$\kappa = 10^{-25}\text{J/cycle}$
Noise power density	$N_0 = 10^{-9}\text{W/Hz}$
Distance between TN and HNs	$d \sim U(10, 100)\text{m}$
Path loss	$PL(d) = 128.1 + 37.6 \log_{10}(d)$
BDNN Hidden Layers	$[64, 32, 16]$ neurons
Learning Rate	0.01
Training Iterations	1000

C. Evaluation Analysis

Fig. 4 compare the performance of four offloading algorithms in terms of delay characteristics. The CDF plot shows that BDNN-TS achieves the best delay performance, with over 95% of tasks completing within 2 seconds, while the tradition MAB learning based algorithm (i.e, ϵ-Greedy, UCB, TS) require longer delays to reach the same completion rate. This superior performance of BDNN-TS is further confirmed by the average delay plot, where it maintains consistently lower delays across all time slots compared to the others. The stability of BDNN-TS is particularly notable, showing minimal delay fluctuations over time, whereas the MAB learning algorithm exhibit more variability in their delay performance. These results suggest that our BDNN-TS approach provides more efficient and stable task offloading compared to the bandit-based methods, making it particularly suitable for time-sensitive applications.

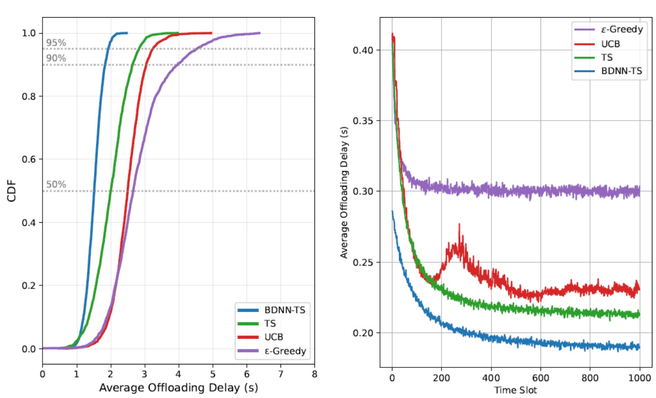

Fig. 4: Average offloading delay with different network size.

Fig. 5 presents the successful offloading ratio, which represents a ratio of tasks successfully offloaded within their deadlines.

The BDNN-TS approach achieves the highest success rate (94.7%), demonstrating its ability to make efficient offloading decisions under dynamic fog computing conditions. TS follows with 85.7%, benefiting from exploration-

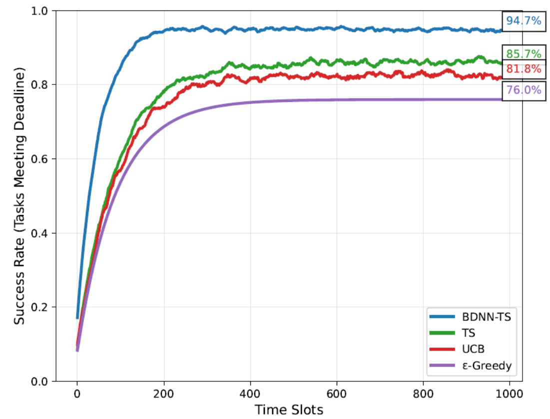

Fig. 5: Successful offloading ratio.

exploitation trade-offs but lacking the deep uncertainty modeling of BDNN-TS. UCB achieves 81.8%, leveraging confidence bounds but struggling with dynamic changes. ϵ-Greedy performs the worst at 76.0%, as its random exploration leads to suboptimal task assignments. The results highlight the advantage of BDNN-TS in maintaining high success rates while adapting to environmental uncertainties.

The comparative analysis of the algorithms in Fig. 6 highlights the energy efficiency of different approaches for task offloading.

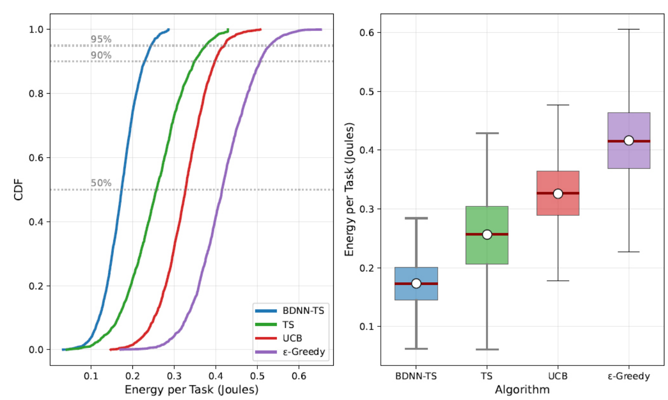

Fig. 6: The average energy per task for the comparative algorithms.

The cumulative distribution function (CDF) plot (left) shows that BDNN-TS achieves the lowest energy consumption per task, with its curve reaching 95% completion at a much lower energy value compared to the other algorithms. TS follows, exhibiting slightly higher energy consumption. UCB and ϵ-Greedy perform worse, with ϵ-Greedy displaying the highest energy consumption per task, as seen from its stretched CDF curve. The box plot (right) further reinforces this observation, where BDNN-TS has the lowest median energy consumption and the least variance, indicating stable and efficient performance. TS and UCB show moderate energy consumption but higher variability, while ϵ-Greedy has the highest median and spread, demonstrating inefficiency in energy consumption. This analysis suggests that BDNN-TS is the most energy-efficient algorithm, making it a strong candidate for energy-aware task offloading in edge computing environments.

VI. Conclusions

This paper presented BDNN-TS, a Bayesian Deep Neural Network integrated with Thompson Sampling, for intelligent task offloading in dynamic fog computing environments. Unlike conventional methods such as ϵ-greedy, UCB, and standard Thompson Sampling—which rely on simple linear models or fixed exploration strategies—BDNN-TS leverages the expressiveness of deep neural networks and Bayesian uncertainty quantification to model the complex and non-stationary reward function in real time.

Through extensive simulations, BDNN-TS consistently outperformed baseline algorithms across key performance metrics, including average task offloading delay, task success rate, and adaptability under uncertain conditions. In particular, BDNN-TS achieved faster convergence and better long-term performance by sampling actions from a learned posterior distribution rather than relying on heuristic exploration. The probabilistic reward modeling in BDNN enables more accurate decision-making under uncertainty, especially when network conditions and helper node resources vary dynamically.

These results validate the effectiveness of combining Bayesian learning with Thompson Sampling in fog computing scenarios. Future work will extend this framework to multi-task settings, incorporate decentralized coordination among task nodes, and explore hybrid strategies that integrate BDNN-TS with other exploration-exploitation techniques for enhanced performance under strict QoS requirements.

Acknowledgments

This work was supported in part by the Ministry of Science and ICT (MSIT), Korea, under the Innovative Human Resource Development for Local Intellectualization program, supervised by the Institute for Information and Communications Technology Planning and Evaluation (IITP) (IITP-2025-RS-2020-II201612 (25%), IITP-2025-RS-2024-00438430 (25%)) and Korea Research Fellowship Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (2018R1A6A1A03024003 (25%), RS-2023-00249687 (25%)).

References

[1] A. V. Dastjerdi and R. Buyya, "Fog computing: Helping the internet of things realize its potential," *Computer*, vol. 49, no. 8, pp. 112–116, 2016.

[2] L. Liu, Z. Chang, X. Guo, S. Mao, and T. Ristaniemi, "Multiobjective optimization for computation offloading in fog computing," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 283–294, 2018.

[3] H. Tran-Dang, S. Bhardwaj, T. Rahim, A. Musaddiq, and D.-S. Kim, "Reinforcement learning based resource management for fog computing environment: Literature review, challenges, and open issues," *Journal of Communications and Networks*, pp. 1–16.

[4] ——, "Reinforcement learning based resource management for fog computing environment: Literature review, challenges, and open issues," *Journal of Communications and Networks*, vol. 24, no. 1, pp. 83–98.

[5] M. M. Razaq, S. Rahim, B. Tak, and L. Peng, "Fragmented task scheduling for load-balanced fog computing based on q-learning," *Wireless communications and mobile computing*, vol. 2022, no. 1, p. 4218696.

[6] S. T. Sumona, S. S. Hasan, A. Y. Tamzid, P. Roy, M. A. Razzaque, and R. Mahmud, "A deep q-learning framework for enhanced qoe and energy optimization in fog computing," in *2024 20th International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT)*. IEEE, pp. 669–676.

[7] J. Shi, J. Du, J. Wang, J. Wang, and J. Yuan, "Priority-aware task offloading in vehicular fog computing based on deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 16 067–16 081.

[8] V. Jain and B. Kumar, "Qos-aware task offloading in fog environment using multi-agent deep reinforcement learning," *Journal of Network and Systems Management*, vol. 31, no. 1, p. 7.

[9] J. Xiong, P. Guo, Y. Wang, X. Meng, J. Zhang, L. Qian, and Z. Yu, "Multi-agent deep reinforcement learning for task offloading in group distributed manufacturing systems," *Engineering Applications of Artificial Intelligence*, vol. 118, p. 105710.

[10] Z. Zhao, H. Zhang, L. Wang, and H. Huang, "A multimodel edge computing offloading framework for deep-learning application based on bayesian optimization," *IEEE Internet of Things Journal*, vol. 10, no. 20, pp. 18 387–18 399.

[11] A. Asheralieva and D. Niyato, "Bayesian reinforcement learning and bayesian deep learning for blockchains with mobile edge computing," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 1, pp. 319–335.

[12] Z. Zhu, T. Liu, Y. Yang, and X. Luo, "Blot: Bandit learning-based offloading of tasks in fog-enabled networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 12, pp. 2636–2649.

[13] S. Misra, S. P. Rachuri, P. K. Deb, and A. Mukherjee, "Multiarmed-bandit-based decentralized computation offloading in fog-enabled iot," *IEEE Internet of Things Journal*, vol. 8, no. 12, pp. 10 010–10 017.