

A Survey on Deep Learning: Approach for Task Offloading in Multi-UAV Aided Mobile Edge Computing

Mani Shankar S¹, Manoj CV², Navya RS³, Prajwal HV⁴, Deepak S Sakkri⁵

Department Of CS&E, Sri Krishna Institute of Technology, B'luru-560090

ABSTRACT

Mobile Edge Computing (MEC) combined with Unmanned Aerial Vehicles (UAVs) has emerged as a promising paradigm to enhance the capabilities of wireless networks by providing computation & storage resources at the edge. Task offloading, the process of allocating computing tasks to appropriate resources, plays a critical role in optimizing the performance of MEC systems. In multi-UAV scenarios, where multiple UAVs are deployed to support computing tasks, task offloading becomes more challenging due to the dynamic and distributed nature of the system. we propose a deep learning approach for task offloading in multi-UAV aided Mobile Edge Computing. We leverage the power of deep learning techniques, such as convolutional neural networks (CNNs) & recurrent neural networks (RNNs), to learn efficient task offloading decisions based on various context parameters, including UAV position, network conditions, and computational resources. We present a comprehensive review of existing deep learning-based task offloading approaches and evaluate their performance through simulations and experiments. The results demonstrate that our proposed deep learning approach outperforms traditional methods, achieving

better resource utilization and reduced latency in multi-UAV MEC scenarios. The research contributes to the optimization of task offloading decisions in dynamic and resource-constrained environments, enabling efficient utilization of UAVs in Mobile Edge Computing systems.

Keywords: Deep learning, deep reinforcement learning, Internet of Things, mobile edge computing, task offloading.

1. INTRODUCTION:

Computation offloading has proven to be an effective method for facilitating resource intensive tasks on IoT mobile edge nodes with limited processing capabilities. Additionally, in the context of Mobile Edge Computing (MEC) systems, edge nodes can offload its computation intensive tasks to a suitable edge server. Hence, they can reduce energy cost and speed up processing. Despite the numerous accomplished efforts in task offloading problems on the Internet of Things (IoT), this problem remains a research gap mainly because of its NP hardness in addition to the unrealistic assumptions in many proposed solutions. In order to accurately extract information from raw sensor data from IoT devices deployed in complicated contexts, Deep Learning (DL) is a potential method. Therefore,

here is an approach based on Deep Reinforcement Learning (DRL) will be presented to optimize the offloading process for IoT in MEC environments. This approach can achieve the optimal offloading decision. A Markov Decision Problem (MDP) is used to formulate the offloading problem. Delay time & consumed energy are the main optimization targets in this work. The proposed approach has been verified using extensive simulations. Simulation results demonstrate that the proposed model can effectively improve the MEC system latency, energy consumption, & significantly outperforms the Deep Q Networks (DQNs) and Actor Critic (AC) approaches. The 5G era networks has been realized based on networking technologies, innovations, and the new computing & communication paradigms. Mobile Edge Computing (MEC) is one of the key technologies for computation distribution that boosts the performance of 5G cellular networks. The main role of MEC is the minimization of communication latency between the user and the server. This behaviour has a great importance for Internet of Things (IoT) environments. IoT has become an important area of research due to its rapid use in our daily lives and in industry. Therefore, it faces numerous challenges, including latency reduction, storage management, energy consumption, task offloading, etc. Increasing the number of end devices in IoT environments leads to a corresponding increase in the number of possible actions. Consequently, it is crucial to enhance the availability and the

terminal-to-terminal delay. By offloading IoT tasks to resource-rich terminals in cooperative edge servers or clouds, mobile end devices can release intensive computation and storage. The end-to-end performance of IoT applications is nevertheless significantly impacted by the various service architectures and offloading techniques. Indeed, computing needs have a greater impact on the performance of IoT applications as compared with connectivity requirements. However, communication bandwidth represents the most important resource as the system expands to support more IoT devices. As a result, it becomes the primary component that directly affects performance as a whole. Building an orchestral IoT architecture is thus a must to include optimum solutions under various limitations for the best offloading location. Even though MEC has several benefits, it is still constrained by the positions of fixed towers. Consequently, it is difficult to deploy MEC servers at any time or location. Furthermore, there is a good chance that natural disasters could occasionally destroy the infrastructure. Additionally, mounting infrastructure in remote locations such as hotspots & mountains is nearly challenging. The IoT nodes are unable to completely service their users in the aforementioned conditions. Unmanned Aerial Vehicles (UAVs) with MEC servers installed on board can be used to support MEC systems by taking advantage of their flexibility and ease of deployment. This support is necessary for tasks that mobile users in hotspot

locations or in emergent scenarios have been temporarily offloaded. In order to provide computing servers for mobile user's terminals in adaptable positions, UAV aided MEC is introduced. By adding additional compute resources to MEC servers, UAV-aided MEC speeds up calculations and increases the operating lifetime of mobile devices. Deep Learning (DL) has been widely used to learn and optimize a variety of issues for UAV aided MEC. Meanwhile, labelling the training data requires a significant amount of human effort. By engaging with MEC surroundings, Reinforcement Learning (RL) can learn and improve UAV-assisted MEC without training data. Therefore, to reduce overall energy consumption, Deep Reinforcement Learning (DRL) approaches can be used to provide effective task offloading, resource allocation, and UAV control. DRL uses RL and Deep Neural Networks (DNNs) for collecting the complicated states of MEC with UAV assistance. When developing distributed decision-making solutions for wireless task offloading problems, conventional approaches such as convex optimization and mixed integer programming are not always appropriate. This concept aims to develop a DRL model to solve task offloading MEC systems. The proposed approach avoids unrealistic assumptions such as ignoring user's device mobility. Hence, the transmitting channel noise is taken into consideration in addition to the coordination of mobile users and UAVs. This model seeks to maximize the

stability of the entire system while minimizing the time and energy it uses. Maximizing stability means balancing the computation power of the system workload, extending operation time, and maximizing the total number of completed tasks.

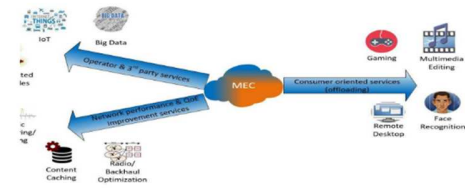


FIGURE-1 MEC and its role in 5G Implementation

Abbreviations and Acronyms

- MEC:** Mobile Edge Computing
- UAV:** Unmanned Aerial Vehicles
- DL:** Deep Learning
- RL:** Reinforcement Learning
- DQNs:** Deep Q Networks
- AC:** Actor Critic
- MDP:** Markov Decision Problem
- DRL:** Deep Reinforcement Learning
- IoT:** Internet of Things
- DNNs:** Deep Neural Networks

II. LITERATURE SURVEY

DRL approaches can autonomously extract features while minimizing human effort and domain expertise required to collect distinguishing characteristics. Hence, they play a key role against the heterogeneity of edge computing environments. Therefore, DRL models can efficiently optimize the task offloading strategy and determines offloading policies. Additionally, online heavy computation iterations can be avoided by offline training. Many

research efforts have been conducted in this direction. For cooperative UAV-enabled MEC networks, the study in presents a cooperative offloading strategy based on UAV-to-device interference mitigation. DRL-based optimization is investigated to obtain the optimal offloading decisions and resource management policies in order to maximize the long-term system utility. Here, the system utility of the DRL-based model is better than related solutions that use non-cooperative UAV edge computing methods. Meanwhile, the study in introduces the multi-objective ant colony optimization approach based on RL. It has been proposed for accurate resource allocation among end-users depending on the cost of creating Q-tables and optimal allocation in MEC. Additionally, fast responsive task offloading based on Meta Reinforcement Learning (MRL) is introduced in to overcome the low sample efficiency of the original RL-based algorithm. MRL enables learning and updating policies according to new environments. Additionally, it enables the user's equipment to run the training process by using its own data with little computing resources. Mobile applications are modelled as directed acyclic graphs and the dynamic offloading process are modelled as multiple MDPs. Moreover, the study in presents the task offloading problem in satellite-terrestrial edge computing networks, where tasks can be offloaded to the visible urban terrestrial cloud via satellite link. DRL-based task offloading is used to accelerate the

learning process by dynamically adjusting the number of candidate locations and the size of action space. The offloading problem is modelled as a mixed-integer programming problem where the offloading location & bandwidth allocation depend only on the current channel state. Furthermore, a reinforcement learning approach is presented in for computational offloading of energy harvesting for IoT devices. This approach uses DRL algorithm with a transfer learning strategy to compress the state space dimensions, accelerate the learning rate, and enhance the offloading system performance and system utility. A distributed offloading approach called best response based offloading algorithm has been introduced using game theory. In this approach, users' devices work together to reduce energy cost and latency cost. Moreover, the authors in investigate UAV-assisted MEC system. In this system, the UAV provides a complementary computation resource to the terrestrial MEC system. UAV tries to maximize the expected long-term computation performance. The study investigates a proactive model based on DRL techniques. MEC system is established for offline training of the proactive DRL model. Furthermore, DDPG based computation offloading algorithm has been introduced in to find the best offloading policy in a dynamic environment for UAV assisted MEC. Also, it can enable a continuous action space offloading decision and UAV mobility but with only one UAV server and one offloading layer. Hence, in this

work, DDPG algorithm is investigated for more complex and heterogeneous environment with more than one offloading layer

Abbreviations and Acronyms:

- MEC:** Meta-Reinforcement Learning
- UAV:** Unmanned Aerial Vehicle
- DRL:** Deep Reinforcement Learning
- DDPG:** Deep Deterministic Policy Gradient
- RL:** Reinforcement Learning

III. METHODOLOGY

1) MEC SYSTEM ARCHITECTURE:

The proposed offloading system architecture which is composed of N end-user devices, M edge servers, K UAVs, one cloud server, and a Central Offloading Controller (COC). The COC is deployed in MEC layer hence, it can be a master MEC server with special and higher efficiency storage and computing resources. This COC is a DDPG-based task offloading agent that is mainly responsible for responding to task computing requests of the end user’s devices. Agent application can get environment information through monitoring devices. These devices are deployed on the user’s device, MEC, and UAV. Furthermore, COC works as an orchestrator that manipulates numerous user offloading requests and collects information to select the optimal computing terminal. A task offloading environment based on DDPG is distributed and linked to the four-tier hierarchy, which includes IoT device layer, UAV server layer, MEC server

layer, and cloud server layer. The following subsections describe the main characteristics of each layer:

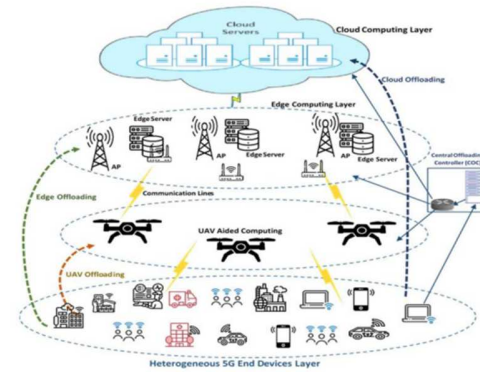


FIGURE 2: MEC System Architecture

- a) **IoT Layer:** A network of interconnected IoT devices is present at this layer. Through wireless access points, each device can link to UAV, MEC, and cloud servers. The IoT user must make dynamic task offloading decisions for each cycle of offloading based on QoS requirements and the state of the network (transmission bandwidth, task size, available resources, etc.).
- b) **UAV Layer:** This layer contains lightweight MEC servers on UAVs, which can provide high mobility and flexible deployment. Hence, the processing delay can be reduced since this layer can offer computing support for jobs that mobile users offload in locations with temporary hotspots. For instance, sports stadiums or communities that have been devastated by natural disasters
- c) **MEC Layer:** MEC servers for real-time task processing are present in this layer. They can offer lower latency computation services at the edge of the network. MEC servers may send

complicated computational jobs to resource-rich cloud servers. In order to guarantee the security of offloading operations, MEC servers provide dependable communication with the IoT device layer and the cloud server layer.

d). **Cloud Server Layer:** This layer consists of numerous powerful virtual machines with higher storage and computational capacity. It is mostly utilized by IoT devices to do complicated computing tasks. Each cloud node in this layer is securely connected to MEC terminals and IoT nodes and runs in a decentralized safe manner.

2) DRL BASED OFFLOADING OPTIMIZATION ALGORITHM:

DRL introduces a deep neural network to replace the Q-table in the RL algorithm. DDPG is an improved version of DQN to make DRL agent efficiently deal with continuous action space. As explained in Figure-3, DDPG uses two separate DQNs for approximating the actor-network (policy-network) and the critic-network (Qvalue network).

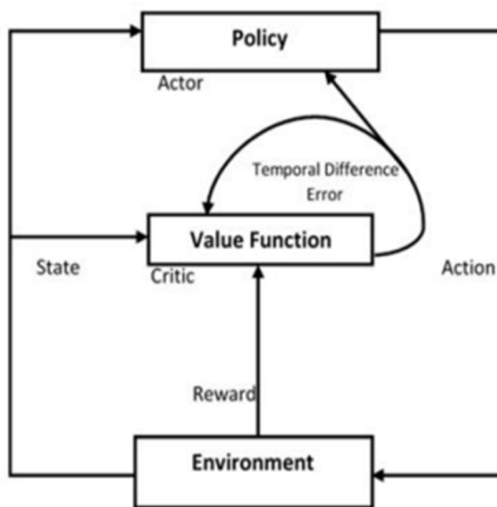


FIGURE 3: AC algorithm

a. State Space: In UAV-aided MEC environments, the state space is jointly described by N UDs, K UAVs, M MECs, and cloud and their surrounding environment. For a given state S_t , the agent adopts action a_t according to the selected policy.

b. Action Space: Based on the current state S_t of the observed system environment parameters, the agent chooses a certain action a_t to offload the requested tasks of all mobile devices nodes to the available computing terminal servers.

c. Reward Function: The behaviour of DDPG agent is based on rewards. Hence, the effectiveness of the DDPG framework is greatly influenced by the selection of a suitable reward function. In order to optimize the reward, it is important to reduce the total processing time and the energy consumption as stated in the equation.

$$R_t(s_t, a_t) = (1 - \epsilon) F_t - \rho_e \log_5(E_t^{total}) - \rho_t \log_3(T_t^{total}) + C$$

3) PROPOSED DDPG-OFFLOADING ALGORITHM:

The DDPG algorithm is the efficient enhanced version of AC algorithm. This is mainly because DDPG uses four neural networks: a Q network (critic), a deterministic policy network (actor), a target Q network, and a target policy network.

The Q network and policy network are much similar to actor and critic networks.

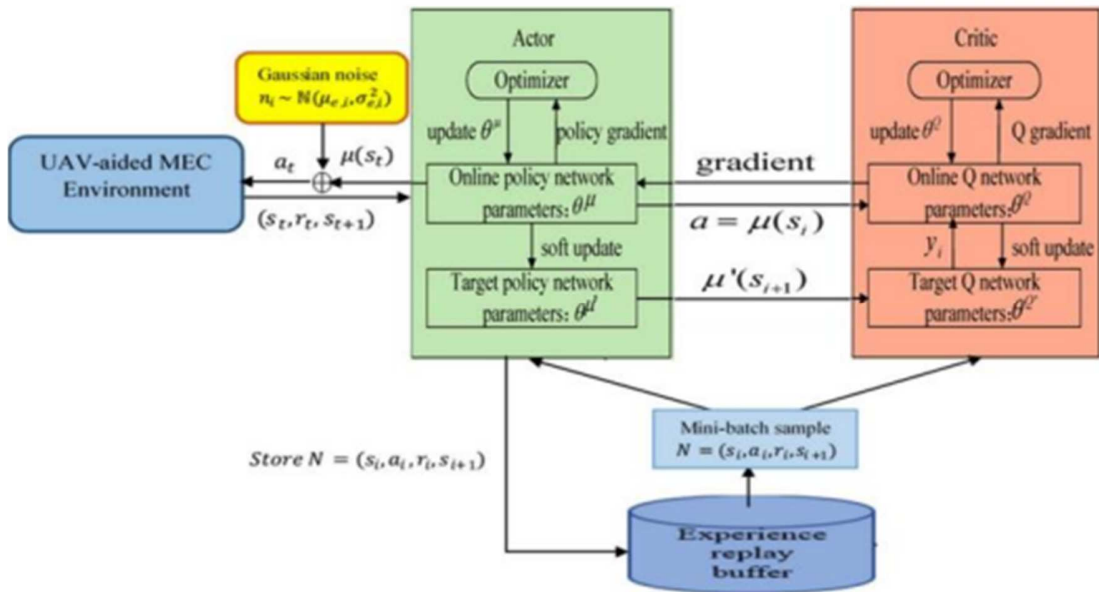


FIGURE-4: Deep deterministic policy gradient algorithm.

However, in DDPG, the Actor is used to create a unique action by directly mapping states to actions instead of outputting the probability distribution across a discrete action space. On the other hand, critic is used to approximate the Q-value action function. The target networks are always time-delayed copies of their original networks that slowly track the learned networks. Therefore, using target value networks can greatly improve stability in the learning process. The improvement of equation in the proposed DDPG algorithm is that critic is updated by minimizing the sum of gradient update loss for each experience sample N. This improvement makes the DDPG model more effective and practical than AC based model. The improved loss equation is described as:

$$Loss(\theta^Q) = \frac{1}{N} \sum_i (y_t - Q(s_t, a_t/\theta^Q))^2$$

IV. IMPLEMENTATION

The details of the simulation study are presented. PyTorch is adopted for developing the proposed DDPG based offloading environment. The adopted simulator in the experiments has three main components. These components are system environment, UAV-aided MEC, and the DDPG controller agent. The entire UAV-aided MEC offloading environment is described as MDP environment. The MDP environment is the focus of the DDPG model actions. Consequently, the proposed DDPG approach is compared with DQN and AC approaches. a. Simulation Setting: Simulation adopts 2D square areas, each of them has N = 10UDs randomly distributed in 300 × 300m² area. Additionally, it is assumed that the UAVs fly at a fixed height H = 100m. Each UAV has a unique mass Muav = 9.65KG and a maximum flight speed 9 = 50m/s. During the training phase, the batch and the buffer sizes are set to be 64 and 10⁵,

respectively. b. Performance Evaluation: The results of the evaluation of DDPG-based computational offloading model is presented in this section. Adam optimizer is adopted to train the DDPG agent, which is an adaptive learning optimization method. To optimize the offloading decision, the state of UAVs, MECs, cloud, users, and the other environment parameters are used as inputs to the actor- network. Meanwhile, the output is the UAV new position and the offloading decision. The input environment state parameters should include UAV current position, UAVs and MECs frequencies, the coordinates and frequencies of the users being served at the moment, and the parameters of the offloaded tasks such as required computation cycles, data size, and expiration time.

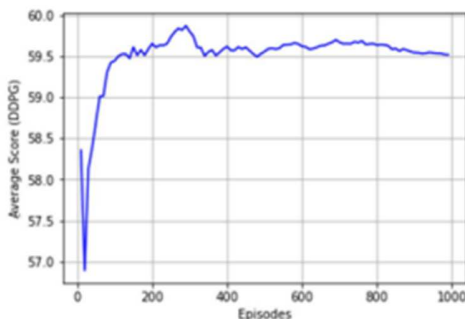


FIGURE-5: Average Score of DDPG agent.

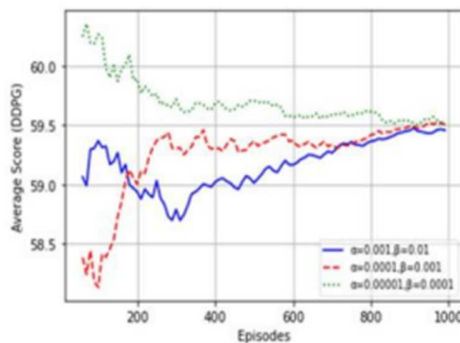


Figure-6: DDPG Convergence under different learning rates

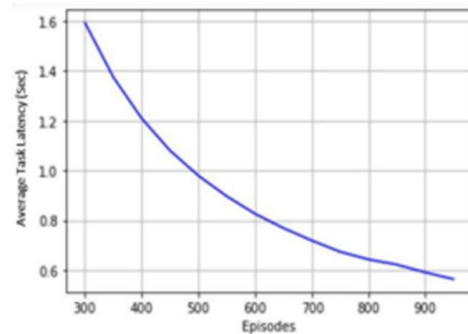


FIGURE-7: DDPG average time delay (Sec) for 50 tasks

Figure-5 shows the average score of the DDPG-based agent. As shown in this figure, the average score increases during the training interval T. This indicates that the performance of DDPG learning improves with the increase of training steps and achieves higher rewards in each episode. Moreover, the proposed DDPG-based agent can efficiently explore the MEC environment action space, which demonstrates that efficient task offloading policies can be successfully learned. Based on the comparison among various learning rates for actor and critic networks, the convergence performance of the proposed algorithm is studied with different learning rates as shown in Figure-6. It can be noticed that when $\alpha = 0.0001$ and $\beta = 0.001$, the proposed DDPG algorithm can have the best convergence. Figure-7 and Figure-8 show the offloading time cost and energy cost of the proposed DDPG-based model. These figures show that both time cost and energy cost decrease over the interval T. These decays of both costs over time prove the efficiency of the proposed model.

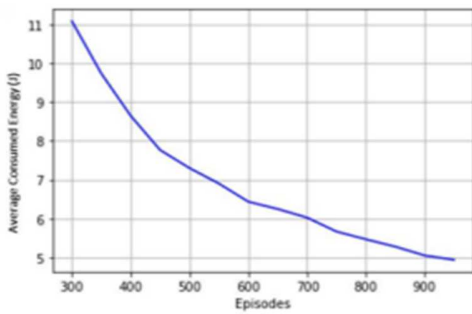


Figure-8: DDPG average consumed energy for 50 tasks.

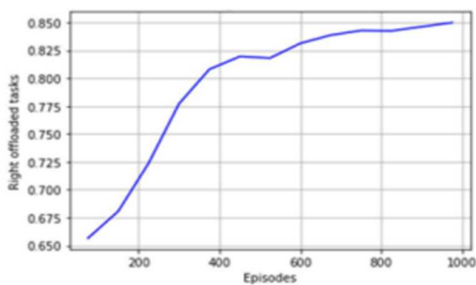


FIGURE-9: Correct offloaded tasks ratio

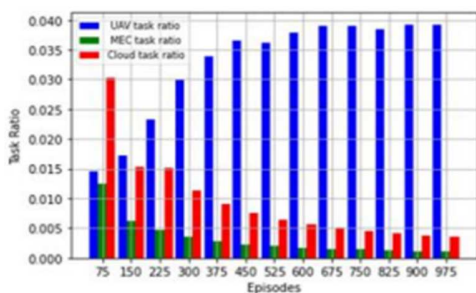


FIGURE-10: Offloaded tasks ratio to UAV, MEC and Cloud Layer.

Figure-9 shows the correct offloaded task ratio of the DDPG-based model with respect to all requested tasks. Figure-10 shows the percentage of tasks that have been offloaded to each system layer. This figure indicates that the number of tasks transferred to cloud and MEC layers is much lower than the number of tasks transferred to UAV layer. Also, it shows that the cloud task ratio decreases over time while the UAV task ratio increases. These observations proves that the DDPG

can efficiently learn over time how to take the best offloading action that minimizes the offloading cost.

V. CONCLUSION

Edge computing is evolving rapidly toward the fundamental infrastructure and facilitating the future of IoT. Efficient coordination mechanisms and task offloading models are leveraged to enable mobile devices and edge-cloud to cooperatively work. This paper investigated one of the efficient deep reinforcement learning algorithms, which is the DDPG algorithm. It proposed a DDPG-based offloading system to improve the efficiency of offloading decision strategy. It assessed the UAV benefits in 5G IoT environments to maximize the percentage of offloaded tasks from the total requested task. The paper proposed a DDPG model to tackle the offloading optimization problem for making correct decisions regarding offloading to one of the previously mentioned layers to reduce energy and time. It was demonstrated that DDPG performed better than DQN. Moreover, offloading to UAVs in cooperation with MECs and cloud servers resolves incomplete offloaded task requests. Hence, the three-layer offloading system and the deep learning-based algorithms can serve most of the task offloading requests and can achieve effective offloading decisions and resource management. In future work, an improvement of the proposed model will be made to maximize the offloading system stability in dynamic and uncontrollable networking environments.

VI. ACKNOWLEDGEMENT

We would like to thank, Dr. Deepak S Sakkari & Dr. Shantharam Nayak for their valuable suggestion, expert advice and moral support in the process of preparing this paper.

REFERENCES

- [1] T. Dragičević, P. Siano, and S. R. S. Prabakaran, "Future generation 5G wireless networks for smart grid: A comprehensive review," *Energies*, vol. 12, no. 11, p. 2140, Jun. 2019, doi:10.3390/en12112140.
- [2] I. Al Ridhawi, M. Aloqaily, Y. Kotb, Y. Al Ridhawi, and Y. Jararweh, "A collaborative mobile edge computing and user solution for service composition in 5G systems," *Trans. Emerg. Telecommun. Technol.*, vol. 29, no. 11, p. e3446, Nov. 2018, doi:10.1002/ett.3446.
- [3] S. Nižetić, P. Šolić, D. López-de-Ipiña González-de-Artaza, and L. Patrono, "Internet of Things (IoT): Opportunities, issues and challenges towards a smart and sustainable future," *J Cleaner Prod*, vol. 274, Nov. 2020, Artno.122877, doi:10.1016/j.jclepro.2020.122877.
- [4] M. S. Hossain, C. I. Nwakanma, J. M. Lee, and D.-S. Kim, "Edge computational task offloading scheme using reinforcement learning for IIoT scenario," *ICT Exp.*, vol. 6, no. 4, pp. 291–299, Dec. 2020, doi:10.1016/j.ict.2020.06.002.
- [5] J. Almutairi and M. Aldossary, "Modeling and analyzing offloading strategies of IoT applications over edge computing and joint clouds," *Symmetry*, vol. 13, no. 3, p. 402, Mar. 2021, doi:10.3390/sym13030402.
- [6] W. Zhang, L. Li, N. Zhang, T. Han, and S. Wang, "Air-ground integrated mobile edge networks: A survey," *IEEE Access*, vol. 8, pp. 125998–126018, 2020, doi:10.1109/ACCESS.2020.3008168.
- [7] L. Zhang, Z.-Y. Zhang, L. Min, C. Tang, H.-Y. Zhang, Y.-H. Wang, and P. Cai, "Task offloading and trajectory control for UAV-assisted mobile edge computing using deep reinforcement learning," *IEEE Access*, vol. 9, pp. 53708–53719, 2021, doi:10.1109/ACCESS.2021.3070908.
- [8] Z. Zhou, H. Liao, B. Gu, K. M. S. Huq, S. Mumtaz, and J. Rodriguez, "Robust mobile crowd sensing: When deep learning meets edge computing," *IEEE Netw.*, vol. 32, no. 4, pp. 54–60, Jul. 2018, doi:10.1109/MNET.2018.1700442.
- [9] F. Jiang, K. Wang, L. Dong, C. Pan, W. Xu, and K. Yang, "Deep-learning based joint resource scheduling algorithms for hybrid MEC networks," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6252–6265, Jul. 2020, doi:10.1109/JIOT.2019.2954503.
- [10] A. M. Andrew, "Reinforcement learning: An introduction," *Kybernetes*, vol. 27, no. 9, pp. 1093–1096, 1998, doi:10.1108/k.1998.27.9.1093.3.
- [11] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K.

- Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015, doi:10.1038/nature14236.
- [12] H. Mei, K. Yang, Q. Liu, and K. Wang, "Joint trajectory-resource optimization in UAV-enabled edge-cloud system with virtualized mobile clone," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 5906–5921, Jul. 2020, doi:10.1109/JIOT.2019.2952677.
- [13] Q. Wang, A. Gao, and Y. Hu, "Joint power and QoE optimization scheme for multi-UAV assisted offloading in mobile computing," *IEEE Access*, vol. 9, pp. 21206–21217, 2021, doi:10.1109/ACCESS.2021.3055335.
- [14] O. Alagoz, H. Hsu, A. J. Schaefer, and M. S. Roberts, "Markov decision processes: A tool for sequential decision making under uncertainty," *Med. Decis. Making*, vol. 30, no. 4, pp. 474–483, Jul. 2010, doi: 10.1177/0272989X09353194.
- [15] M. McClellan, C. Cervelló-Pastor, and S. Sallent, "Deep learning at the mobile edge: Opportunities for 5G networks," *Appl. Sci.*, vol. 10, no. 14, p. 4735, Jul. 2020, doi: 10.3390/app10144735.
- [16] Q.-V. Pham, F. Fang, V. N. Ha, M. J. Piran, M. Le, L. B. Le, W.-J. Hwang, and Z. Ding, "A survey of multi-access edge computing in 5G and beyond: Fundamentals, technology integration, and state-of-the-art," *IEEE Access*, vol. 8, pp. 116974–117017, 2020, doi: 10.1109/ACCESS.2020.3001277.
- [17] N. Kiran, X. Liu, S. Wang, and C. Yin, "VNF placement and resource allocation in SDN/NFV-enabled MEC networks," in *Proc. IEEE Wireless Commun. Netw. Conf. Workshops*, Apr. 2020, pp. 1–6, doi: 10.1109/WCNCW48565.2020.9124910.
- [18] M. McClellan, C. Cervelló-Pastor, and S. Sallent, "Deep learning at the mobile edge: Opportunities for 5G networks," *Appl. Sci.*, vol. 10, no. 14, p. 4735, Jul. 2020, doi: 10.3390/app10144735.
- [19] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing— A key technology towards 5G," *IEEE Internet Things J.*, vol. 11, no. 11, pp. 1–16, 2015.
- [20] M. A. Abdelaal, G. A. Ebrahim, and W. R. Anis, "High availability deployment of virtual network function forwarding graph in cloud computing environments," *IEEE Access*, vol. 9, pp. 53861–53884, 2021, doi: 10.1109/ACCESS.2021.3068342.
- [21] M. A. Abdelaal, G. A. Ebrahim, and W. R. Anis, "Efficient placement of service function chains in cloud computing environments," *Electronics*, vol. 10, no. 3, p. 323, Jan. 2021, doi: 10.3390/electronics10030323.
- [22] K. Antevski, C. J. Bernardos, L. Cominardi, A. de la Oliva, and A. Mourad, "On the integration of NFV and MEC technologies: Architecture analysis and benefits for edge robotics," *Comput.*

Netw., vol. 175, Jul. 2020, Art. no. 107274, doi: 10.1016/j.comnet.2020.107274.

[23] Z. Ullah, F. Al-Turjman, and L. Mostarda, "Cognition in UAV-aided 5G and beyond communications: A survey," *IEEE Trans. Cognit. Commun. Netw.*, vol. 6, no. 3, pp. 872–891, Sep. 2020, doi: 10.1109/TCCN.2020.2968311.

[24] A. Nakao and P. Du, "Toward in-network deep machine learning for identifying mobile applications and enabling application specific network slicing," *IEICE Trans. Commun.*, vol. E101.B, no. 7, pp. 1536–1543, Jul. 2018, doi: 10.1587/transcom.2017CQI0002.

[25] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali, and M. Guizani, "A survey of machine and deep learning methods for Internet of Things (IoT) security," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1646–1685, 2020, doi: 10.1109/COMST.2020.2988293.