A Review on Optimizing Offloading Performance in Heterogeneous IoT using Mobile Edge Devices as Nodes

Mardhani Riasetiawan
Department of Computer Science and Electronics
Faculty of Mathematics and Natural Sciences, Universitas Gadjah Mada
Yogyakarta, Indonesia

Ahmad Ashari
Department of Computer Science and Electronics
Faculty of Mathematics and Natural Sciences, Universitas Gadjah Mada
Yogyakarta, Indonesia

Roghib Muhammand Hujja
Department of Computer Science and Electronics
Faculty of Mathematics and Natural Sciences, Universitas Gadjah Mada
Yogyakarta, Indonesia

ABSTRACT
Offloading, the process of transferring data and tasks from one device to another has been identified as a promising approach for improving performance and reducing workload in the Internet of Things (IoT). However, offloading in a heterogeneous IoT environment, with a wide range of devices and technologies, can be challenging. Mobile edge devices, which provide low-latency connectivity and perform computation at the edge of the network, have been proposed to optimize offloading performance in such an environment.

In this literature review, we examine the existing research on using mobile edge devices as nodes for offloading in heterogeneous IoT. We present a comprehensive overview of the various approaches and techniques proposed for selecting the most appropriate device to handle offloaded tasks, including using machine learning algorithms for predicting performance and optimizing the offloading decision-making process. We also discuss the challenges and limitations of these approaches and provide directions for future research.

Our review highlights the potential of mobile edge devices as a solution for improving offloading performance in heterogeneous IoT and serves as a valuable resource for researchers and practitioners working in this field.

General Terms
Internet of Things

Keywords
Offloading, IoT, Mobile Edge Devices, Machine Learning

1. INTRODUCTION
The Internet of Things (IoT) is a rapidly expanding network of connected devices that can communicate with each other to exchange data and perform tasks. These devices, which can include everything from smart thermostats and security cameras to industrial control systems, often need more resources and capabilities. As a result, they may only sometimes be able to handle the workload required, leading to performance issues and reduced functionality[1]–[3]. One solution to this problem is offloading and transferring data and tasks from one device to another. This can be done to reduce the workload on a device, improve performance, or save resources such as battery life or data usage. For example, an intelligent thermostat might offload data processing tasks to the cloud to reduce its local workload and improve its performance, or a security camera might offload video streaming to a nearby edge device to save on data usage[2], [4], [5].

Offloading can be particularly challenging in a heterogeneous IoT environment with a wide range of devices and technologies. Mobile edge devices, which can provide low-latency connectivity and perform computation at the edge of the network, have been proposed to optimize offloading performance in such an environment[6].

There has been significant research on using mobile edge devices as nodes for offloading in heterogeneous IoT. Some existing approaches have focused on developing algorithms for selecting the most appropriate device to handle offloaded tasks based on factors such as network conditions, device capabilities, resource availability, and priority of tasks. Others have explored machine learning techniques to predict the performance of different offloading scenarios and optimize the offloading decision-making process[4], [7].

This paper reviews the existing research on using mobile edge devices as nodes for offloading in heterogeneous IoT. We present a comprehensive overview of the approaches and techniques proposed for selecting the most appropriate device to handle offloaded tasks. We also discuss the challenges and limitations of these approaches and provide directions for future research.

2. RESEARCH METHODOLOGY
This section introduces the systematic literature review methodology **[5], [8]** for the offloading schemes proposed in the edge computing-related literature. At first, to find review and survey articles in the offloading context, we employed the search strings "offloading edge computing" in the IEEE and Scopus. The results achieved from these searches are screened to find credible and original articles.

The remaining of these articles are used in conducting this study and will be reviewed in the next section. Furthermore, Table 1 describes the main research questions considered in this paper and why they are needed.

Table 1. Research Question

Index	Question	Reason
1	Which is the most optimal method to optimize offloading on a heterogenous IoT System?	Offloading can effectively improve the performance of devices in the IoT by reducing their workload and allowing them to delegate tasks to more capable devices or servers. By identifying the most optimal method for offloading, it is possible to maximize the

		benefits of offloading and improve the overall performance of the IoT system.
2	How effective is the Markov Decision Process in helping offload efficiency?	The MDP is a mathematical framework that can be used to model and optimize decision-making processes. Applying the MDP to the offloading problem in the IoT makes it possible to develop algorithms that can make optimal offloading decisions based on various factors such as network conditions, device capabilities, and resource availability. However, we need to know how good this method is in individual use or combined with other methods.

3. HETEROGENOUS IoT

This section provides essential background knowledge about heterogeneous IoT and discusses various properties of mobile edge devices.

3.1 Device

An IoT (Internet of Things) device is a physical device connected to the internet and able to communicate with other devices. These devices can be used for various purposes, such as monitoring and control, data collection and analysis, and automation. Some examples of IoT devices include smart thermostats, security cameras, industrial control systems, and wearable fitness trackers**[9]**.

IoT devices are becoming more heterogenous or diverse in their characteristics due to the rapid expansion of the IoT and the increasing number of devices and technologies being integrated into it. This diversity can be seen in terms of the types of devices connected to the IoT, as well as the technologies and protocols used to enable communication and interoperability between devices[1], [10].

The increasing heterogeneity of IoT devices is driven by several factors, including the need for devices to be compatible with a wide range of platforms and technologies, the desire to incorporate new and emerging technologies into the IoT, and the demand for greater customization and specialization of devices for specific use cases.

3.2 Network

IoT (Internet of Things) devices in a heterogeneous network may use various technologies and protocols to communicate with each other and with other devices or servers. Some standard technologies and protocols that are used in IoT communication include:

- Wi-Fi: This wireless networking technology uses radio waves to transmit data over short distances. Wi-Fi is widely used in the IoT for connecting to the internet or local networks[2], [4].
- Bluetooth: This wireless networking technology uses radio waves to transmit data over short distances. Bluetooth is often used in the IoT for devices that need to communicate with each other over short distances, such as in a personal area network (PAN)[2], [4].
- Cellular networks are networks that use wireless communication to transmit data over long distances, such as those used by mobile phones. Some IoT devices, such as those that need to operate over vast areas or locations without access to Wi-Fi or other local networks, may use cellular networks for communication[4], [7].
- Zigbee: This wireless networking technology uses radio waves to transmit data over short distances. Zigbee is often used in the IoT for devices that need to communicate with each other over short distances and consume low amounts of power[2], [4].
- Ethernet: This wired networking technology uses cables to transmit data over longer distances. Ethernet is often used in the IoT for devices that need to connect to local networks or the internet through a physical connection[2], [4].

The choice of technology and protocol for communication in a heterogeneous IoT network will depend on the specific requirements and constraints of the devices and the overall system. Factors such as the distance over which communication is needed, the power consumption of the devices, the bandwidth requirements, and the cost of the technology may all be considered when selecting a communication technology or protocol for a heterogeneous IoT network.

3.3 Framework

The framework is a set of technologies, protocols, and standards that enable communication and interoperability between a diverse range of IoT devices. These frameworks typically include the following:

- Communication protocols: These are the rules and standards that define how devices can exchange data and communicate with each other. Examples of communication protocols used in the IoT include Wi-Fi, Bluetooth, cellular networks, and Zigbee[2], [4].
- Data management standards: These are the standards that define how data is collected, stored, and shared by IoT devices. Examples of data management standards used in the IoT include the IoT Protocol Stack, which defines standards for data representation, data modeling, and data access, and the OPC Foundation, which defines standards for interoperability between devices in industrial automation systems[2], [4].
- Security protocols: These are the protocols and standards used to protect the confidentiality, integrity, and availability of data and devices in the IoT. Examples of security protocols used in the IoT include encryption, authentication, and access control mechanisms[2], [4].
- Application programming interfaces (APIs): These are the interfaces that enable communication between devices and software applications. APIs expose the functionality of devices to software developers, allowing them to build applications that can interact with and control the devices[2], [4].

A heterogeneous IoT framework is designed to enable communication and interoperability between a wide range of devices and technologies in the IoT, regardless of their manufacturer or domain. This enables the creation of more complex and robust IoT systems that can deliver value to users through increased functionality, performance, and reliability.

3.4 Mobile Edge Devices

A mobile edge device is a device that is located at the edge of a network and can provide low-latency connectivity and perform computation. Mobile edge devices are often used in mobile networks and the Internet of Things (IoT) to provide connectivity and compute resources to devices that are located in the field or at the edge of the network. Mobile edge devices include:

- Mobile base stations: These devices are used to provide wireless connectivity to mobile devices in a network. Mobile base stations are typically located in areas with high demand for connectivity, such as urban areas, and can support many users[8], [10].
- Small cells are devices that provide wireless connectivity to small areas, such as a single building or a small group of buildings. Small cells are often used to increase the capacity and coverage of a mobile network in areas with high demand for connectivity[8], [10].
- Edge servers: These are located at the edge of a network and are used to provide computing resources and services to devices in the field. Edge servers are often used in the IoT to enable the processing of data and the execution of tasks closer to the source of the data, reducing latency and improving performance[8], [10].

Mobile edge devices play a crucial role in enabling connectivity and computation at the edge of a network and are an essential component of many mobile and IoT systems.

4. OFFLOADING SCHEMES

This section provides essential background knowledge about offloading and discusses various methods to help optimize the offloading problem.

4.1 Problem

Offloading is transferring data and tasks from one device to another to reduce the workload on the device, improve performance, or save resources such as battery life or data usage. In the context of the Internet of Things (IoT), offloading can address the challenges posed by the limited resources and capabilities of many IoT devices.

However, offloading in a heterogeneous IoT environment, with a wide range of devices and technologies, can be particularly challenging. This is because devices in a heterogeneous IoT may have different capabilities, protocols, and resources, making it difficult to determine the most appropriate device to handle offloaded tasks.[5], [8], [10]

Several factors can impact the effectiveness of offloading in a heterogeneous IoT environment:

- Network conditions: The availability and quality of the network connection between devices can impact offloading performance. If the network connection is poor or unavailable, it may not be possible to offload tasks effectively[5], [8], [10].
- Device capabilities: The capabilities of the devices that are involved in the offloading process, such as their processing power, memory, and storage, can impact the performance of offloading. If the device receiving the offloaded tasks does not have sufficient capabilities to handle them, the system's performance may suffer[5], [8], [10].
- Resource availability: The availability of resources such as battery life, data usage, and compute resources can impact the effectiveness of offloading. A device needs more resources to be able to offload tasks effectively[5], [8], [10].

In the journal that has been reviewed, the work discovers some main problems that need to be solved in recent research about offloading, summarized in Table 2. The table shows that the most widespread problem is Task Queue and Multi-user. Much research is trying to solve task queue problems, and we want to make the system closer to real-time. Then there is also a problem with the multi-user when requesting tasks and receiving results together. Also, the dynamic network and random arrival arise as problems when offloading is used in moving devices such as vehicles or mobile phones. Then the common IoT problem such as latency, long-term use, and scheduling still exist, and much research still wants to solve the problem[5], [8], [10].

Table 2. The recent problem in offloading

Index	Problem	Number of citation
1	Dynamic Network [4], [11]–[14]	5
2	Latency [5], [7], [8], [15], [16]	5
3	Long Term Use [11], [17]	2
4	Multi-User [1], [12], [18]–[21]	6
5	Random Arrival [20], [22]–[24]	4
6	Scheduling [2], [6], [25]	3
7	Task Queue [3], [9], [26]–[29]	6

The offloading problem in a heterogeneous IoT environment can be complex and challenging due to the diverse range of devices and technologies involved. Identifying the most appropriate device to handle offloaded tasks and optimizing the offloading process to take into account the various factors that can impact performance, is an essential aspect of maximizing the benefits of offloading in a heterogeneous IoT environment.

4.2 Variable

In the context of recent offloading research in the Internet of Things (IoT), several variables are often considered in the analysis and evaluation of offloading approaches. From the reviewed journal, these variables can include:

- Bandwidth: The amount of data transmitted over a network connection in a given period. In offloading research, bandwidth can be a variable of interest because it can impact offloading performance. For example, a device with a high bandwidth connection may be able to offload tasks more efficiently than a device with a low bandwidth connection [22], [23], [29].
- Location: The physical location of a device can impact the performance of offloading. For example, a device located in an area with poor network coverage may be unable to offload tasks effectively. In offloading research, location can be a variable of interest because it can be used to evaluate the suitability of a device for offloading based on its proximity to other devices or the network [14], [15], [25], [28].
- Task queue: The tasks waiting to be processed by a device

can impact the offloading performance. In offloading research, the size and characteristics of the task queue can be variables of interest because they can be used to evaluate the suitability of a device for offloading and to optimize the offloading process to reduce the size of the queue [5], [7], [8], [10], [11], [13], [14].

- Energy: The amount of energy consumed by a device can impact offloading performance. In offloading research, energy can be a variable of interest because it can be used to evaluate the impact of offloading on a device's energy consumption and optimize the offloading process to conserve energy [11], [12], [14], [17].
- Channel quality: The communication channel quality between devices can impact offloading performance. In offloading research, variables such as signal strength and error rate can be used to evaluate the channel's quality and optimize the offloading process accordingly [11], [21].
- Delay: The time it takes to transmit data between devices can impact offloading performance. In offloading research, delay can be a variable of interest because it can be used to evaluate the impact of offloading on the system's latency and optimize the offloading process to reduce delay [9], [13], [14], [19].
- Data loss: Data loss during transmission between devices can impact offloading performance. In offloading research, data loss can be a variable of interest because it can be used to evaluate the offloading process's reliability and optimize the offloading process to reduce data loss [12], [14], [27].
- Computing resource: The number of computing resources available on a device, such as processing power, memory, and storage, can impact offloading performance. In offloading research, computing resources can be a variable of interest because they can be used to evaluate the suitability of a device for offloading and to optimize the offloading process to maximize the use of available resources [4], [5], [7], [28].

These variables can all play a role in the performance of offloading in the Internet of Things (IoT) and can be used to evaluate and optimize offloading approaches in different scenarios. The specific variables used in a given offloading research study will depend on the research question being addressed and the goals of the offloading process. And for the detailed data. The summary of the variable used in the reviewed journal is shown in table 3.

Table 3. variable used in recent offloading research

Index	Problem	Number of citation
1	Bandwidth [1], [3], [9], [10], [15], [18], [19], [22], [23], [29]	11
2	Channel [9], [11], [21], [24]	4
3	Computation [4], [5], [7], [28]	5
4	Delay [2]–[4], [7], [9], [13], [14], [19], [21], [24]–[26], [30], [31]	13
5	Energy [3], [7], [11], [12], [14], [17], [19]–[23], [25], [28], [31]	14
6	Location [14], [15], [25], [28]	4
7	Loss [12], [14], [27]	3
8	Task queue [1]–[3], [5], [7], [8], [10], [11], [13], [14], [16], [18]–[21], [23]–[26], [28], [28], [29]	20

4.3 Method

The reviewed journal has proposed several approaches to improve offloading performance in a heterogeneous Internet of Things (IoT) environment. These approaches can be grouped into several categories, including:

- Algorithms for selecting the most appropriate device to handle offloaded tasks: These approaches use various metrics, such as device characteristics, network conditions, and resource availability, to identify the most suitable device for a given offloading scenario. Examples of these approaches include the Markov Decision Process (MDP) and the Value Iteration Algorithm [22], [23].
- Machine learning techniques for predicting the performance of different offloading scenarios: These approaches use machine learning algorithms, such as deep neural networks, to learn the optimal offloading strategy based on observed data. Examples of these approaches include Deep Q-Networks (DQN) and Deep Reinforcement Learning (DRL) [1], [25], [30], [31].
- Frameworks for optimizing the offloading process: These approaches provide a structured approach for optimizing the offloading process based on various factors, such as the characteristics of the devices and tasks, the network conditions, and the available resources. Examples of these approaches include the Virtual Continuous Time System (VCTS) and the Proximal Policy Optimization (PPO) Algorithm [15], [29].

The summary of the methods that have been proposed to improve offloading performance in a heterogeneous IoT environment is shown in table 4.

Table 4. the variable used in recent offloading research

Method	Description
Markov Decision Process (MDP) [1]–[3], [3], [5]–[7], [9], [12]–[16], [18], [19], [19], [20], [20]–[25], [27], [28]	A mathematical framework for modeling decision-making problems and identifying the optimal strategy based on different actions' expected rewards or utilities.
Value Iteration Algorithm [15], [21]	An algorithm for solving MDPs by iteratively updating an estimate of the value of each state based on the expected rewards or utilities of the available actions.
Q Function [11], [14], [17], [27], [30]	A function used to represent the expected reward or utility of taking a specific action in a specific state in an MDP.
Deep Q-Network (DQN) [11], [28], [31]	A type of neural network used to approximate the Q function in an MDP and learn the optimal offloading strategy.

Post Decision State [17]	A state in an MDP represents the system's state after an action has been taken.
Virtual Continuous Time System (VCTS) [30]	A mathematical model represents a system's behavior over time as a set of differential equations.
Deep Reinforcement Learning (DRL) [4], [7]–[10], [12], [13], [20], [23]–[25], [30]	A reinforcement learning algorithm using deep neural networks to learn the optimal offloading strategy.
Convolutional Neural Network (CNN) [25]	A type of neural network used to process data with a grid-like topology, such as images.
Stochastic Gradient Descent (SGD) [23]	An optimization algorithm is used to update the parameters of a neural network by calculating the gradient of the loss function concerning the parameters.
Asynchronous Advantage Actor Critic (A3C) [13]	A reinforcement learning algorithm using multiple parallel agents to learn the optimal offloading strategy.
Lagrangian Transformation [2]	A mathematical technique used to transform a problem into a new form that is easier to solve.
Proximal Policy Optimization (PPO) Algorithm [16], [25]	A type of reinforcement learning algorithm is used to learn the optimal offloading strategy by updating the Q function based on observed rewards or utilities.
Deep Deterministic Policy Gradient (DDPG) Algorithm [3], [31]	A reinforcement learning algorithm using deep neural networks to learn the optimal offloading strategy. Designed to handle continuous action spaces.

The method shown in table 5 is used in a different case to optimize offloading performance in heterogeneous IoT networks. The detailed use of the method is explained in the paragraph below:

- Markov Decision Process (MDP): This mathematical framework is used to model decision-making problems in which an agent takes actions in a sequence of states to maximize some reward or utility. In the context of offloading in the Internet of Things (IoT), MDPs can be used to model the offloading process as a sequence of states and actions and to identify the optimal offloading strategy based on the expected rewards or utilities.

- Value Iteration Algorithm: This algorithm is used to solve MDPs by iteratively updating an estimate of each state's value based on the available actions' expected rewards or utilities. In the context of offloading in the IoT, the value iteration algorithm can be used to identify the optimal offloading strategy by iteratively updating estimates of the value of each offloading decision.

- Q Function: This is a function that is used to represent the expected reward or utility of taking a specific action in a specific state in an MDP. In the context of offloading in the IoT, the Q function can be used to evaluate the expected rewards or utilities of different offloading decisions and to identify the optimal offloading strategy.

- Deep Q-Network (DQN): This type of neural network is used to approximate the Q function in an MDP. DQN is trained using reinforcement learning algorithms to learn the optimal offloading strategy by iteratively updating the Q function based on the observed rewards or utilities of different offloading decisions.

- Post-Decision State: This is a state in an MDP that represents the system's state after an action has been taken. In the context of offloading in the IoT, post-decision states can be used to model the impact of offloading decisions on the system's state and identify the optimal offloading strategy.

- Virtual Continuous Time System (VCTS): This is a mathematical model that represents the behavior of a system over time as a set of differential equations. In the context of offloading in the IoT, VCTSs can be used to model the offloading process's dynamics and identify the optimal offloading strategy.

- Deep Reinforcement Learning (DRL): This reinforcement learning algorithm uses deep neural networks to learn the optimal offloading strategy by iteratively updating the Q function based on the observed rewards or utilities of different offloading decisions.

- Convolutional Neural Network (CNN): This type of neural network is used to process data with a grid-like topology, such as images. In the context of offloading in the IoT, CNNs can be used to process data from sensors and other devices and to identify patterns that can be used to optimize the offloading process.

- Stochastic Gradient Descent (SGD): This is an optimization algorithm that is used to update the parameters of a neural network by calculating the gradient of the loss function concerning the parameters. In the context of offloading in the IoT, SGD can be used to optimize the performance of DRL algorithms by updating the parameters of the neural network based on the observed rewards or utilities of different offloading decisions.

- Asynchronous Advantage Actor Critic (A3C): This reinforcement learning algorithm uses multiple parallel agents to learn the optimal offloading strategy by updating the Q function based on the observed rewards or utilities of different offloading decisions.

- Lagrangian Transformation: This mathematical technique is used to transform a problem into a new form that is easier to solve. In the context of offloading in the IoT, Lagrangian transformation can be used to transform the offloading problem into a more tractable form that can be solved more efficiently.

- Proximal Policy Optimization (PPO) Algorithm: This reinforcement learning algorithm is used to learn the optimal offloading strategy by updating the Q function based on the observed rewards or utilities of different offloading decisions. The PPO algorithm is designed to be more stable and efficient than other reinforcement learning algorithms, making it well-suited for use in the IoT.

- Deep Deterministic Policy Gradient (DDPG) Algorithm: This reinforcement learning algorithm uses deep neural networks to learn the optimal offloading strategy by updating the Q function based on the observed rewards or utilities of different offloading decisions. The DDPG algorithm is designed to handle continuous action spaces, making it well-suited for use in the IoT.

These methods can optimize offloading in a heterogeneous IoT environment by identifying the most appropriate devices to handle offloaded tasks, predicting the performance of different offloading scenarios, and optimizing the overall system performance.

5. CONCLUSION

Based on the data provided from the review, it is clear that offloading optimization in a heterogeneous Internet of Things (IoT) environment has been an active area of research in recent years. The number of publications on this topic has steadily increased, with a particularly significant increase in the number of publications in 2022. This trend suggests that offloading optimization is an important and growing research area, with many researchers developing new approaches and techniques to improve offloading performance in the IoT.

A wide variety of methods have been proposed to optimize offloading in a heterogeneous IoT environment, including algorithms for selecting the most appropriate device to handle offloaded tasks, machine learning techniques for predicting the performance of different offloading scenarios, and frameworks for optimizing the offloading process. These methods can improve offloading performance in the IoT by enabling more efficient resource utilization and reducing the burden on individual devices.

Overall, the literature review conclusion suggest that offloading optimization in a heterogeneous IoT environment is a promising research area with significant potential to improve the performance of the IoT. Further research is needed to continue developing and refining these approaches and better understand their limitations and potential applications.

6. ACKNOWLEDGMENTS

We want to thank all of the researchers who have contributed to the field of offloading optimization in a heterogeneous Internet of Things (IoT) environment. Their work has greatly informed our understanding of this complex and significant problem, and their insights have been invaluable in developing this paper.

7. REFERENCES

[1] X. Wang, Z. Lu, S. Sun, J. Wang, L. Song, and M. Nicolas, “Optimization Scheme of Trusted Task Offloading in IIoT Scenario Based on DQN,” *Comput. Mater. Contin.*, vol. 74, no. 1, Art. no. 1, 2022, doi: 10.32604/cmc.2023.031750.

[2] R. Li, Q. Ma, J. Gong, Z. Zhou, and X. Chen, “Age of Processing: Age-Driven Status Sampling and Processing Offloading for Edge-Computing-Enabled Real-Time IoT Applications,” *IEEE Internet Things J.*, vol. 8, no. 19, pp. 14471–14484, Oct. 2021, doi: 10.1109/JIOT.2021.3064055.

[3] Y. Lyu, Z. Liu, R. Fan, C. Zhan, H. Hu, and J. An, “Optimal Computation Offloading in Collaborative LEO-IoT Enabled MEC: A Multi-agent Deep Reinforcement Learning Approach,” *IEEE Trans. Green Commun. Netw.*, pp. 1–1, 2022, doi: 10.1109/TGCN.2022.3186792.

[4] B. Dai, J. Niu, T. Ren, and M. Atiquzzaman, “Toward Mobility-Aware Computation Offloading and Resource Allocation in End–Edge–Cloud Orchestrated Computing,” *IEEE Internet Things J.*, vol. 9, no. 19, pp. 19450–19462, Oct. 2022, doi: 10.1109/JIOT.2022.3168036.

[5] M. Masdari and H. Khezri, “Efficient offloading schemes using Markovian models: a literature review,” *Computing*, vol. 102, no. 7, pp. 1673–1716, Jul. 2020, doi: 10.1007/s00607-020-00812-x.

[6] J. Ren *et al.*, “Collaborative task offloading and resource scheduling framework for heterogeneous edge computing,” *Wirel. Netw.*, 2021, doi: 10.1007/s11276-021-02768-y.

[7] Y. Li, L. Liang, J. Fu, and J. Wang, “Multiagent Reinforcement Learning for Task Offloading of Space/Aerial-Assisted Edge Computing,” *Secur. Commun. Netw.*, vol. 2022, p. e4193365, May 2022, doi: 10.1155/2022/4193365.

[8] A. Shakarami, M. Ghobaei-Arani, and A. Shahidinejad, “A survey on the computation offloading approaches in mobile edge computing: A machine learning-based perspective,” *Comput. Netw.*, vol. 182, p. 107496, Dec. 2020, doi: 10.1016/j.comnet.2020.107496.

[9] W. Wu, P. Yang, W. Zhang, C. Zhou, and X. Shen, “Accuracy-Guaranteed Collaborative DNN Inference in Industrial IoT via Deep Reinforcement Learning,” *IEEE Trans. Ind. Inform.*, vol. 17, no. 7, pp. 4988–4998, Jul. 2021, doi: 10.1109/TII.2020.3017573.

[10] J. Ren *et al.*, “Collaborative task offloading and resource scheduling framework for heterogeneous edge computing,” *Wirel. Netw.*, Sep. 2021, doi: 10.1007/s11276-021-02768-y.

[11] X. Chen, H. Zhang, C. Wu, S. Mao, Y. Ji, and M. Bennis, “Optimized Computation Offloading Performance in Virtual Edge Computing Systems Via Deep Reinforcement Learning,” *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4005–4018, Jun. 2019, doi: 10.1109/JIOT.2018.2876279.

[12] Y. Li and B. Shi, “A Deep Reinforcement Learning based Mobile Device Task Offloading Algorithm in MEC,” in *2020 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*, Dec. 2020, pp. 200–207. doi: 10.1109/ISPA-BDCloud-SocialCom-SustainCom51426.2020.00051.

[13] J. Zou, T. Hao, C. Yu, and H. Jin, “A3C-DO: A Regional Resource Scheduling Framework Based on Deep Reinforcement Learning in Edge Scenario,” *IEEE Trans. Comput.*, vol. 70, no. 2, pp. 228–239, Feb. 2021, doi: 10.1109/TC.2020.2987567.

[14] K. Lin, B. Lin, and X. Shao, “Reinforcement Learning-Based Computation Offloading Approach in VEC,” in *Computer Supported Cooperative Work and Social Computing*, Singapore, 2022, pp. 563–576. doi: 10.1007/978-981-19-4546-5_44.

[15] G. Yang, L. Hou, X. He, D. He, S. Chan, and M. Guizani, "Offloading Time Optimization via Markov Decision Process in Mobile-Edge Computing," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2483–2493, Feb. 2021, doi: 10.1109/JIOT.2020.3033285.

[16] S. Chen, L. Rui, Z. Gao, W. Li, and X. Qiu, "Cache-Assisted Collaborative Task Offloading and Resource Allocation Strategy: A Metareinforcement Learning Approach," *IEEE Internet Things J.*, vol. 9, no. 20, pp. 19823–19842, Oct. 2022, doi: 10.1109/JIOT.2022.3168885.

[17] X. Zheng, M. Li, M. Tahir, Y. Chen, and M. Alam, "Stochastic Computation Offloading and Scheduling Based on Mobile Edge Computing," *IEEE Access*, vol. 7, pp. 72247–72256, 2019, doi: 10.1109/ACCESS.2019.2919651.

[18] J. Chen, Y. Yang, C. Wang, H. Zhang, C. Qiu, and X. Wang, "Multitask Offloading Strategy Optimization Based on Directed Acyclic Graphs for Edge Computing," *IEEE Internet Things J.*, vol. 9, no. 12, pp. 9367–9378, Jun. 2022, doi: 10.1109/JIOT.2021.3110412.

[19] B. Qu, Y. Bai, Y. Chu, L. Wang, F. Yu, and X. Li, "Resource allocation for MEC system with multi-users resource competition based on deep reinforcement learning approach," *Comput. Netw.*, vol. 215, p. 109181, Oct. 2022, doi: 10.1016/j.comnet.2022.109181.

[20] J. Wang, H. Ke, X. Liu, and H. Wang, "Optimization for computational offloading in multi-access edge computing: A deep reinforcement learning scheme," *Comput. Netw.*, vol. 204, p. 108690, Feb. 2022, doi: 10.1016/j.comnet.2021.108690.

[21] B. Yao, B. Wu, S. Wu, Y. Ji, D. Chen, and L. Liu, "An Offloading Algorithm based on Markov Decision Process in Mobile Edge Computing System," *Int. J. Circuits Syst. Signal Process.*, vol. 16, pp. 115–121, 2022, doi: 10.46300/9106.2022.16.15.

[22] S. Huang, B. Lv, and R. Wang, "MDP-Based Scheduling Design for Mobile-Edge Computing Systems with Random User Arrival," in *2019 IEEE Global Communications Conference (GLOBECOM)*, Dec. 2019, pp. 1–6. doi: 10.1109/GLOBECOM38437.2019.9013178.

[23] S. Huang, B. Lv, R. Wang, and K. Huang, "Scheduling for Mobile Edge Computing With Random User Arrivals—An Approximate MDP and Reinforcement Learning Approach," *IEEE Trans. Veh. Technol.*, vol. 69, no. 7, pp. 7735–7750, Jul. 2020, doi: 10.1109/TVT.2020.2990482.

[24] J. Li, Z. Chen, and X. Liu, "Deep reinforcement learning for partial offloading with reliability guarantees," in *2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*, Sep. 2021, pp. 1027–1034. doi: 10.1109/ISPA-BDCloud-SocialCom-SustainCom52081.2021.00143.

[25] W. Zhan *et al.*, "Deep-Reinforcement-Learning-Based Offloading Scheduling for Vehicular Edge Computing," *IEEE Internet Things J.*, vol. 7, no. 6, pp. 5449–5465, Jun. 2020, doi: 10.1109/JIOT.2020.2978830.

[26] X. Meng, W. Wang, Y. Wang, V. K. N. Lau, and Z. Zhang, "Closed-Form Delay-Optimal Computation Offloading in Mobile Edge Computing Systems," *IEEE Trans. Wirel. Commun.*, vol. 18, no. 10, pp. 4653–4667, Oct. 2019, doi: 10.1109/TWC.2019.2926465.

[27] Z. Gao, W. Hao, Z. Han, and S. Yang, "Q-Learning-Based Task Offloading and Resources Optimization for a Collaborative Computing System," *IEEE Access*, vol. 8, pp. 149011–149024, 2020, doi: 10.1109/ACCESS.2020.3015993.

[28] B. Lin, K. Lin, C. Lin, Y. Lu, Z. Huang, and X. Chen, "Computation offloading strategy based on deep reinforcement learning for connected and autonomous vehicle in vehicular edge computing," *J. Cloud Comput.*, vol. 10, no. 1, p. 33, Jun. 2021, doi: 10.1186/s13677-021-00246-6.

[29] X. Yang, H. Luo, Y. Sun, and M. Guizani, "A Novel Hybrid-ARPPO Algorithm for Dynamic Computation Offloading in Edge Computing," *IEEE Internet Things J.*, vol. 9, no. 23, pp. 24065–24078, Dec. 2022, doi: 10.1109/JIOT.2022.3188928.

[30] H. Zhang, W. Wu, C. Wang, M. Li, and R. Yang, "Deep Reinforcement Learning-Based Offloading Decision Optimization in Mobile Edge Computing," in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, Apr. 2019, pp. 1–7. doi: 10.1109/WCNC.2019.8886332.

[31] Z. Qinghua, C. Ying, Z. Jingya, and L. Yong, "Computation offloading Optimization in Edge Computing based on Deep Reinforcement Learning," in *2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE)*, Dec. 2020, pp. 1552–1558. doi: 10.1109/ICMCCE51767.2020.00340.