REVIEW ARTICLE

SAMRIDDHI Volume 14, Issue 4, 2022 Print ISSN: 2229-7111 Online ISSN: 2454-5767

A Review on Performance Analysis of Deep Learning for Task Offloading in Edge Computing

S. Almelu*, S. Veenadhari, Kamini Maheshwar

Department of CSE, Rabindranath Tagore University, Bhopal, Madhya Pradesh, India

Abstract

Edge computing is a type of distributed computing that was designed especially for Internet of Things (IoT) users to provide computational resources and data management nearby to users' devices. Introducing edge computing for IoT networks has reduced the bandwidth and latency issue while handling real-time applications. The major benefit of edge computing is that it reduces the communication overhead between IoT user and server. Integrating IoT in our daily life has attracted researchers towards its performance management such as complexity minimization, latency minimization, memory management, energy consumption minimization, etc. The paper focuses on deep reinforcement learning to minimize the computational complexity at IoT user end. The task offloading decision process is designed using Q-Learning that minimizes the system cost and curse of high dimensional data.

Keywords: Edge Computing, Task Offloading, Deep Reinforcement, Q-learning.

SAMRIDDHI : A Journal of Physical Sciences, Engineering and Technology (2022); DOI: 10.18090/samriddhi.v14i04.26

INTRODUCTION

The cloud computing model is a performance management strategy that gives users the path to various dispersed functions in cloud data centers, including compute, networking, and data storage. Cloud service providers (CSPs) offer services such as infrastructure as a service (IaaS) to enable flexible and utility computing, software as a service (SaaS) and including Cloud platform services (PaaS), and Network administrators, for example, might customize programs to match their own requirements, change programs, and access cloud computing wherever on the internet.[1] Cloud-based solutions appropriate for enterprises with fluctuating and escalating network bandwidth requirements. Furthermore, using cloud services, business clients may deploy web apps to marketplace without worrying over managed network expenses, emergency preparedness management, or automatic updates. To leverage the benefits of the technology, several deployment types, such as public, private, hybrid cloud, are key features for stability analysis and scope for commercial purposes. Ever since inception, the cloud services industry has altered dramatically throughout all vertical sectors and in people's everyday lives. Furthermore, business IT investing in cloud-based applications will grow better than average IT, offering cloud computing among the most competitive capabilities in IT markets as it transitions from traditional software. As a result, cloud services are expected to become inescapable and pervasive in any professional or commercial market, comparable to the Web's current dominant position.[2]

While cloud computing may help organizations save money and enhance productivity by incorporating intricate cloud-based operational models, it even has drawbacks in a variety of situations, such as smart manufacturing, connected autonomous vehicles (CAVs), remote monitoring in residence and offices, and intelligent homes. For instance, cloud computing-based computation necessitates massive levels of information transmission between end sensors and devices, which consumes significant network bandwidth. Moreover, due to processing and memory constraints, cloud data service analytics is impracticable to have enormous data collected by millions or billions end nodes. As a result, cloud-based processing does not deliver quick response and low latency from massive IoT devices for big data analytics. In addition, cloud computing data center's are not confident in performing data analytics in certain cases where the security of our resources and privacy are the first concern. Protecting information privacy, on the other hand, permits data to be stored close to its source rather than in faraway data centers on the cloud. Edge computing is a

Corresponding Author: S. Almelu, Department of CSE, Rabindranath Tagore University, Bhopal, Madhya Pradesh, India. Email: almelu02@gmail.com

How to cite this article: Almelu, Veenadhari S, Maheshwar K. (2022). A Review on Performance Analysis of Deep Learning for Task Offloading in Edge Computing. *SAMRIDDHI : A Journal of Physical Sciences, Engineering and Technology,* 14(4), 159-164.

Source of support: Nil

Conflict of interest: None

framework that enables critical information to be analyzed or securely stored before being transmitted to a centralized on-premises facilities or cloud source repository.[3] In IoT scenarios, edge systems capture information by sensed data and analyze this immediately or transfer that to on-premise data centre or Off-premise cloud computing for analysis if processing and analysis are insufficient. It required, the edge computing paradigm will move part of the burden from centralized on-premises facilities to network edge devices outside of cloud data centers, reducing or even removing the pressure on the centralized site. Likewise, as compared to conventional cloud computing models, fog computing reduces capability while introducing new data interchange, setting the framework for developing new, highly heterogeneous computer and network architecture as well as fog cloud integration.[4,5] A primary impetus for edge computing is the need for a scalable extremely large data volume or real-time data processing in the IoT ecosystem. Edge computing is concerned with data creation and consumption at the network's edge in applications such as the smart home, autonomous city, and mobile internet of things. Computational offloading is crucial in edge computing since it helps to reduce latency and ensure user satisfaction. Tasks can be delegated from cloud computing to appropriate technology to bridge the gap between operating overheads, energy consumption, and efficiency. Game and edge-cloud cooperation, heuristic offloading, and other methodologies and techniques for compute offloading have been suggested by Wei et al.[6] Additionally, computationally offloading-oriented optimization is recommended, including edge-cloud collaboration optimal computation offloading and even strategic planning resource allocation.[7,8]

Edge Computing Architecture

Edge computing, the modern computing concept that euphemistically describes the data collection phase, points on a computer network that connects directly to the internet, called Network Edge. The network edge for such scenarios might be a location where data storage, processing, or networking sites are given. Edge computing is consolidating and shrinking the distance between different edge devices and centralized cloud computing resources, thanks to the Internet, 5G connectivity, driverless cars, and intelligent homes. Edge computing may improve information distribution and software product offerings by shortening the data transmission path between the endpoint or its location as it aims at bringing implementation through centralized On-premises data centre to end-users and application-generated data. Moreover, if security and privacy are critical, edge computing ensures data protection by retaining network information edge instead of transferring it to centralized cloud computing platforms.

Furthermore, because there are hundreds of thousands of smart edge objects spread across the globe, and data analysis is carried out on a variety of edge servers, it is critical to develop new scheme frameworks appropriate in edge cloud computing, as the data created from several applications or systems operating in the edge computing framework are significant. Figure 2 depicts the edge computing paradigm. With the number of IoT devices grows, many smartphones or apps need stricter service quality management's requirements and a method where requests for data entry must be handled swiftly called real-time data processing activity.

The edge computing architecture, in contrast to the traditional cloud computing approach, performs only those mathematical calculations and storage at the channel's edge, and it extends information technology, connectivity, and collection capabilities pass through the cloud computing framework into cable network edge to utilize completely the processing of computational power in end-end devices. Shi et al. use an analysis of a smart city device's edge architecture to address problems like durability, insulation, flexibility, and uniqueness in the production of innovative designs and software platforms in edge-related computing environments. Numerous alternative computing approaches have been established for cloud computing model just to adapt to various application conditions and suit various service demands. The Fog computing paradigm was suggested and known as a virtual server computing service model for cloud-related processes outside the scope of a network.

Fog Computation centralizes storage, computing, and applications on network edge devices, eliminating the need for everyone to be stored in a cloud computing environment and addition of a layer present in the end system and data center. The intermediate layer employs fog servers used upon that network edge to minimize the distance of off-premises cloud data centers with edge PCs, the internet backbone bandwidth, and power consumption. Furthermore, cloud-edge collaboration generates considerable delays in network communications and facilities, and during data transmission, data privacy and energy consumption should be addressed.

Subsequently, there is a need for optimal offloading of task that can handle the major issues and challenges of the IoT edge computing networks. Traditional offloading schemes are not much efficient to give optimal decisions for dynamic environment.[9,10] Many researchers proposed dynamic task offloading algorithms, such as Markov decision process (MDP), by interaction with environmental parameters. But MDP can handle single task at the same time. Further, reinforcement learning showed its effectiveness in resolving the issues of MDP but still needs to make the learning process more adaptable to channel environment.[11,12]

Computational Offloading using Reinforcement Learning

Reinforcement learning is an optimal learning process that learns the channel state by cooperating with atmosphere. In Edge-IoT network, every IoT user is considered to be as agent and their requirements are considered as environment.

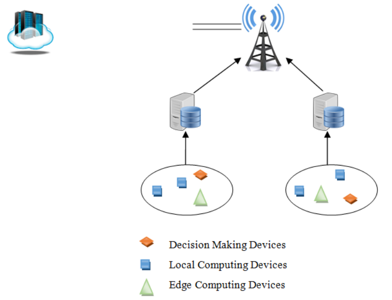

Fig.1: IoT Network with Edge Computing with offloading Scheme

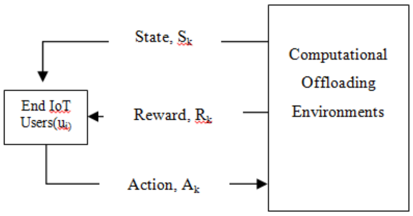

Fig 2: Reinforcement Learning for IoT Users

As shown in Figure 1, three-layer architecture is given for designing Cloud-Edge-IoT computing network. There are many IoT networks connected with Edge server. Each IoT network composed of large number of IoT users, gateway devices (or edge devices) that can collect data from IoT users. These gateways units collect information from Users connected in a specific coverage area and analyze it on an edge server with limited storage and processing capabilities. As each IoT user has a distinct sort of job of varying size, these devices, whether gateway or edge server, may enable limited processing concurrently. Because these computing resources have limited resources and battery capacity, optimum job offloading is required to increase the overall network's performance and lifespan.

In Figure. 2, Edge-IoT task offloading scheme is illustrated with reinforcement learning. In this architecture, each agent, IoT user (u_i), having state (S_k), at time instance (k) with action (A_k) to determine the computing mode based on the decision policy given by environmental parameters. Therefore, if thereis change in environment new state (S_{k+1}) is observed with new reward R_k.

A reinforcement learning consists of following elements:

- Agent: These are the end users, IoT devices, that can perform some actions.
- Environment: These are the determining parameters that decides the reward of agents.
- State: The states explores the environment status such as channel gain, queue status, computation capacity, etc.
- Policy: Mapping by environment features to actions to the activities the agents performs in the environment, alternatively it may be stated that a policy is a mapping by the features of the environmental to the matter, end user does in the environment.
- Reward: A reward can determine whether an agent can offload the task or not. For this, reinforcement learning is implemented. For task offloading each agent maximizes the rewards during learning. In Edge computing, to minimize the system computation cost with respect to power, time and space, negative reward is adopted make right decisions.
- Action: The action is performed by each IoT operator for choosing local computing as well as sometimes the offloading.

Related Work

Liu et al. (2020) studied a machine learning technique for resource allocation in Internet-of-Things (IoT) networks using edge computing. In this work, centralized clustering of IoT users is proposed is by assigning user's priorities. The highest priority cluster is selected and assigned to offload their task at the edge server whereas the task having lowest priority is can compute their task locally. The distributed task offloading is performed by Markov decision process which consider all the IoT user as agent and makes a series of task offloading decision. This policy is designed with respect to cost effectiveness based on channel dynamics. In this approach, deep Q-network was used for high dimensional tasks to learn the optimal policy.

Alfakih et al. (2020) suggested a method using deep reinforcement earning approach for designing state-action-reward policy for task offloading from IoT users to edge server. The optimal offloading decision was performed to minimize system cost in respect to time delay and energy.

Wang et al. (2019) proposed task offloading algorithm for fog-cloud network design of Internet of Vehicles (IoV). The objective was to minimize the power consumption and computational resources of the vehicles.NP-hard problem was formulated as optimization of offloading issues.

Alelaiwi (2019) presented studied how to utilize a DL focused reaction time predictive framework to select if to offload in the cloud hub, close haze/edge hub otherwise neighbor haze/edge hub, close fog/edge node otherwise neighbor fog/edge node. A limited Boltzmann ML is also used to cope with resource availability irregularities.

Chen et al. (2020) proposed an algorithm for offloading task between IoT and edge server and termed it as intelligent Task Offloading Algorithm (iTOA). After evaluating to existing algorithms, iTOA decides offloading activities on current state of network by using Monte Carlo Tree Search (MCTS). To provide quick search facilities, the MCTS algorithm was merged with Deep Neural Network (DNN).

Wei et al. (2018) analyzed a system comprising of multiple mobiles which are intended to carry out uploading action through Multi-Access Edge Computing server and by a single cell, hence its proper allocation to restricted server with communication networks channels (wireless) emerges out as a problem. The author has hence designed an optimization problem for energy and tasks on the mobile so that they are efficiently divided. A Select Maximum Saved Energy First

(SMSEF) framework being the key algorithm utilized.

Higuchi et al. (2019) investigated a virtual edge server and its compatibility with dealing it with multiple vehicles. However, he concluded that during its early development stage, its feasibility requires more concern. The concluding simulations of the paper suggested that for vehicles the horizontal form of offloading can reduce the peak power in edge computing structure nearly by 53%. The penetration rate of vehicle-to-vehicle V2V communication technology was maintained less during this study.

Khayyat et al. (2020) Focused on multilevel vehicular edge cloud networks and presented a powerful DL algorithm for it. The main focus is on achieving energy conservation and effectively utilizing the resources shared by the vehicles, achieving an integration model for computational constrained offloading. The reduction in time and energy consumption is attained by achieving the binary formulation for resource allocation. However, it was found that due to problem of dimensionality, this type of solution is NP difficult and is thus very complex/prohibited to solve them. Thus, the authors have developed a similar reinforcement learning method and proposed a distributed deep learning algorithm along with neural network-based learning soas to finally achieve an optimum solution for taking offloading decisions.

Li et al. (2020) suggested a Deep RL a reinforcement learning calculation for taking care of intricate calculation offloading issue for the Edge Computing Server(ECS) that is heterogeneous in nature with combined computing resources. The main concern in this work that was focused on is network condition and task characteristics. The actor gradient policy was designed to settle on enhanced choices of offloading of tasks. Considering performing multiple tasks, due to the variation of edge subnets and the diversity of edge operations, the suggested calculation may get acquainted with the network and produce a calculation offloading option to reduce the allotment latency.

Guo et al. (2019) focused on improving the operation earning by dropping the charge of the MDs. As the computation demands vary according to regions and there is difference in the availability of the computational edge servers, the traditional method to achieve the objective cannot be chosen. To address this, the article found that in UDN situations with temporal variation, we may build an effective computational strategy and offloading job profile. The author came up with a deep Q-network-focused solution to the issue.

In different IoT's networking modes, the emergence of edge computing has resulted in numerous issues for the researchers and developers, including dynamic computation offloading scheme designing, resources such as computing resource, spectrum useful resource and their allocation, and transmit electricity controlling. It has been found that these are very difficult to solve independent as the designing of and computational offloading scheme for the edge server has to learn about the resources of the gateways as well as the transmitting power capability of an end user. There has been focused study on computational offloading schemes capable of handling multiple resources in the edge servers, specifically in the MEC frameworks. Power consumption and latency are the two main area of concern while working in IoT network with offloading scheme of task. So, many researchers have proposed an optimized task offloading scheme that can handle energy requirements and latency such as MEC system. But MEC-based task offloading can handle single user at a time. Then, multi-user framework was implanted using deep reinforcement learning using MEC framework. However, these algorithms are not much efficient in handling multi-user multi-task at the same time, optimally.

Along with this in order to achieve threshold structure of the system, an optimal task offloading policy is validated. The clustering of the IOT users into various groups has been proposed by making use of clustering optimizing algorithm which is framed as initial step for the task offloading scheme designing. The reduction in the power utilization, time and system cost along with considerable decrement in the execution latency is achieved by making use of distribution computation task offloading algorithm. Diverse contributions from various authors towards the work have been described in the Table 1.

In view of the previous studies, following main issues were being listed:

- One of the main issues of Markov Decision Problem (MDP) is that it can be only utilized with single user with task scheduling.
- MDP can lead to most effective values however creates time-constraint troubles.
- Q-learning is a method that can be utilized for resolving the MDP issue however it may face huddles such as overestimated function. For some stochastic environments the famous reinforcement studying algorithm Q-getting is known to perform very poorly. This poor performance results from massive overestimations of action values. These overestimations due to a positive bias that is because the largest action value is used as an estimate for the most expected action value in Q-learning.

ENERGY AND QOS TRADEOFF

Ever more programs are operating on mobile applications these days, and the level of the service or product is perhaps the most essential metric for determining the success of apps and gadgets. Nevertheless, smartphones near the network's edge typically have scarce resources, such as processing power, storage capacity, and fast charging, making it challenging to fulfill mobile users' rising demands. Resources are distributed and planned according to user needs and contractual arrangements to implement comprehensive and advanced services. As a consequence, delay-sensitive applications must take precedence, while computation-intensive applications must take precedence. Towards this aim, a customer's subjective assessment of the service

Table 1: A comparative overview of different task offloading methods in an iot network.

Author	*Description*	*Results*	*Drawbacks*
Liu et al. (2020)	Deep reinforcement learning approach with markov decision process	Optimal clustering of IoT users and results in optimal energy cost.	Not adaptable to multiple users processing simultaneously.
Li et al. (2020)	Deep Deterministic Policy Gradient	minimize the task delay	Energy cost consumption was not considered.
Alelaiwiet al. (2019)	Deep learning for prediction of response time	Improves the offloading computational performance	Time-constraint issues
Xiaolan et al. (2019)	Greedy Q learning algorithm for optimal offloading t=of task	Task offloading performance was improved in terms of energy usage and overall network delay requirements.	Not adaptable to multiple users processing simultaneously.
Chen et al. (2019)	Deep Reinforcement Learning Approach	Improved service latency performance	There is no predictive model as well not adaptable to massive IoT scenario
Min et al. (2019)	Fast deep Q-network (DQN) based offloading approach	After a significant amount of learning time, the best offloading policy is determined.	Local power control as well as offloading decision-making problem
Zhang et al. (2018)	Cost as well as energy aware offloading task is proposed using deep reinforcement learning.	Energy and cost were estimated	Multiple users cannot be served optimally

quality and efficiency of gadgets, systems, services, and apps is referred to as quality of experience (QoE). Offloading computation to edge servers and subsequently returning calculation results to smartphones can considerably reduce the smart mobile device's resource usage.[13-15] It is required to establish and execute QoS and QoE criteria, construct an acceptable TO (Task offloading) sequence during the compute offloading process, and evaluate the offloaded timings of each task.

Furthermore, while the quantity of edge nodes grows, so does the power consumption of edge and data centres. Furthermore, high energy consumption might result in increased operational costs and decreased system dependability. As a result, resource awareness is an important component in making better computation offloading decisions.[16-18] Computing offloading to mobile devices has advantages such as energy savings and increased processing capabilities, however communication delays among portable devices, edge nodes, and cloud servers might impair system performance.[19,20] As a result, for computation offloading, the right mix of computing and communication is crucial. Wang et al.[21] Combined optimizing issues, which employs compute offloads to minimize the power consumption when reducing programme executing delays on mobile devices. They framed the challenge as a Min-ED and a 0-1.0 integers linear programming (ILP) problem, then used weighted double matching to find the best solution in polynomial time for specific circumstances when the mobile device has enough remaining energy so each program demands a same amount of wealth Customer mobility and low latency must be addressed in the development and execution of compute offloading techniques in mobile cloud computing and service execution processes, according to Deng et al.[22] Using genetic algorithm optimization, they presented a new compute offloading technique. Wang and Li[23] suggested In mobile cloud computing (MCC) systems, a context-aware offloading approach and a compute offloading mechanism have been developed. They provided a simple price calculation framework for cloud computing resources to estimate task execution costs, including implementation effort and timeperiod usage. Wang's offloading method could provide impartial judgments and locations during runtime by conforming to depending on network state, equipments data, and the range of different cloud computing resources. To supply flexible mobile cloud computing services, the recommended offloading framework analyzes several cloud resources, including a wireless ad hoc network, cloud resources, and public, utilizing the suggested offloading approach and simulation methodologies. Similarly, the suggested strategy ignores interactions with other cloud services, which might significantly impact the example program's component failure-tolerant capabilities. It has been recommended that high-density, low-cost, and high-write nonvolatile data storage, such as nonvolatile memory, be installed upon that edge system to guarantee that the edge node media storing easy save and retrieve many times, with no interrupting in real data. The addition of power evaluation and financial reporting to the command line and runtime library could even help the edge computing OS (i.e. operating system) which traditionally conducted energy consumption evaluation decomposition technique as well as estimation primarily while in the system software level, enabling edge computing to become fully programmable and reduce run - time code energy consumption.

CONCLUSION

It is known that edge computing provides the computational capability to the IoT users as IoT computing requires large resource spectrum. Also, transmission of computational tasks to an edge server consumes more power. So, conserving energy locally at IoT user end is quite complicated. This can be done by optimizing network design or routing policies. In this paper a brief review of different task offloading scheme are discussed with their limitations. Further proposed algorithm depends on modified Q-learning dependent deep reinforcement approach to handle multiple users and multiple task simultaneously at minimum computational cost.

REFERENCES

[1] Alfakih, T., Hassan, M. M., Gumaei, A., Savaglio, C., & Fortino, G. (2020). Task offloading and resource allocation for mobile edge computing by deep reinforcement learning based on SARSA. IEEE Access, 8, 54074–54084. https://doi.org/10.1109/ACCESS.2020.2981434

[2] Alelaiwi, A. (2019). An efficient method of computation offloading in an edge cloud platform. Journal of Parallel and Distributed Computing, 127, 58–64. https://doi.org/10.1016/J.JPDC.2019.01.003

[3] Chen, J., Chen, S., Luo, S., Wang, Q., Cao, B., & Li, X. (2020). An intelligent task offloading algorithm (iTOA) for UAV edge computing network. Digital Communications and Networks, 6(4), 433–443. https://doi.org/10.1016/J.DCAN.2020.04.008

[4] Chen, J., Chen, S., Wang, Q., Cao, B., Feng, G., & Hu, J. (2019). IRAF: A Deep Reinforcement Learning Approach for Collaborative Mobile Edge Computing IoT Networks. IEEE Internet of Things Journal, 6(4), 7011–7024. https://doi.org/10.1109/JIOT.2019.2913162

[5] Chen, X., Zhang, H., Wu, C., Mao, S., Ji, Y., & Bennis, M. (2019). Optimized computation offloading performance in virtual edge computing systems via deep reinforcement learning. IEEE Internet of Things Journal, 6(3), 4005–4018. https://doi.org/10.1109/JIOT.2018.2876279

[6] Deng, S., Huang, L., Taheri, J., & Zomaya, A. Y. (2015). Computation Offloading for Service Workflow in Mobile Cloud Computing. IEEE Transactions on Parallel and Distributed Systems, 26(12), 3317–3329. https://doi.org/10.1109/TPDS.2014.2381640

[7] Guo, H., Lv, J., & Liu, J. (2019). Smart Resource Configuration and Task Offloading with Ultra-Dense Edge Computing. International Conference on Wireless and Mobile Computing, Networking and Communications, 2019-October. https://doi.org/10.1109/WIMOB.2019.8923227

[8] Higuchi, T., Ucar, S., & Altintas, O. (2019). Offloading Tasks to Vehicular Virtual Edge Servers. Proceedings - 2019 IEEE 16th International Conference on Mobile Ad Hoc and Smart Systems Workshops, MASSW 2019, 162–163. https://doi.org/10.1109/MASSW.2019.00040

[9] Jiang, C., Fan, T., Qiu, Y., Wu, H., Zhang, J., Xiong, N. N., & Wan, J. (2018). Interdomain I/O Optimization in Virtualized Sensor Networks. Sensors (Basel, Switzerland), 18(12). https://doi.org/10.3390/S18124395

[10] Jiang, C., Han, G., Lin, J., Jia, G., Shi, W., & Wan, J. (2019). Characteristics of Co-Allocated Online Services and Batch Tasks in Internet Data Centers: A Case Study from Alibaba Cloud. IEEE Access, 7, 22495–22508. https://doi.org/10.1109/ACCESS.2019.2897898

[11] Jiang, C., Wang, Y., Ou, D., Li, Y., Zhang, J., Wan, J., Luo, B., & Shi, W. (2019). Energy efficiency comparison of hypervisors. Sustainable Computing: Informatics and Systems, 22, 311–321. https://doi.org/10.1016/J.SUSCOM.2017.09.005

[12] Khayyat, M., Elgendy, I. A., Muthanna, A., Alshahrani, A. S., Alharbi, S., & Koucheryavy, A. (2020). Advanced Deep Learning-Based Computational Offloading for Multilevel Vehicular Edge-Cloud Computing Networks. IEEE Access, 8, 137052–137062. https://doi.org/10.1109/ACCESS.2020.3011705

[13] Li, Y., Qi, F., Wang, Z., Yu, X., & Shao, S. (2020). Distributed Edge Computing Offloading Algorithm Based on Deep Reinforcement Learning. IEEE Access, 8, 85204–85215. https://doi.org/10.1109/ACCESS.2020.2991773

[14] Liu, X., Qin, Z., & Gao, Y. (2019). Resource Allocation for Edge Computing in IoT Networks via Reinforcement Learning. IEEE International Conference on Communications, 2019-May. https://doi.org/10.1109/ICC.2019.8761385

[15] Liu, X., Yu, J., Wang, J., & Gao, Y. (2020). Resource Allocation With Edge Computing in IoT Networks via Machine Learning. IEEE Internet of Things Journal, 7(4), 3415–3426. https://doi.org/10.1109/JIOT.2020.2970110

[16] Min, M., Xiao, L., Chen, Y., Cheng, P., Wu, D., & Zhuang, W. (2019). Learning-Based Computation Offloading for IoT Devices with Energy Harvesting. IEEE Transactions on Vehicular Technology, 68(2), 1930–1941. https://doi.org/10.1109/TVT.2018.2890685

[17] Qiu, Y., Jiang, C., Wang, Y., Ou, D., Li, Y., & Wan, J. (2019). Energy Aware Virtual Machine Scheduling in Data Centers. Energies 2019, Vol. 12, Page 646, 12(4), 646. https://doi.org/10.3390/EN12040646

[18] Wang, C., & Li, Z. (2004). A computation offloading scheme on handheld devices. Journal of Parallel and Distributed Computing, 64(6), 740–746. https://doi.org/10.1016/J.JPDC.2003.10.005

[19] Wang, X., Han, Y., Wang, C., Zhao, Q., Chen, X., & Chen, M. (2019). In-edge AI: Intelligentizing mobile edge computing, caching and communication by federated learning. IEEE Network, 33(5), 156–165. https://doi.org/10.1109/MNET.2019.1800286

[20] Wang, X., Wang, J., Wang, X., & Chen, X. (2017). Energy and Delay Tradeoff for Application Offloading in Mobile Cloud Computing. IEEE Systems Journal, 11(2), 858–867. https://doi.org/10.1109/JSYST.2015.2466617

[21] Wang, X., Wei, X., & Wang, L. (2019). A deep learning based energy-efficient computational offloading method in Internet of vehicles. China Communications, 16(3), 81–91. https://doi.org/10.12676/J.CC.2019.03.008

[22] Wei, F., Chen, S., & Zou, W. (2018). A greedy algorithm for task offloading in mobile edge computing system. China Communications, 15(11), 158–170. https://doi.org/10.1109/CC.2018.8543056

[23] Zhang, C., Liu, Z., Gu, B., Yamori, K., & Tanaka, Y. (2018). A Deep Reinforcement Learning Based Approach for Cost- and Energy-Aware Multi-Flow Mobile Data Offloading. IEICE Transactions on Communications, E101B(7), 1625–1634. https://doi.org/10.1587/transcom.2017CQP0014