# A REVIEW OF EDGE COMPUTING AND RESOURCE MANAGEMENT: CHALLENGES AND STATE OF THE ART SOLUTIONS

**Y. Kefas*[1], I. O. Oyefolahan[2] and B. Sulaimon[3]**
[1]Nasarawa State University, Keffi, Nigeria.
[2, 3] Federal University of Technology, Minna, Nigeria.

*Email of Corresponding Author: *kefasyunana@nsuk.edu.ng*

***ABSTRACT***
*The Internet of Things with its discovery for linking billions of devices and static devices to aid with numerous applications in real time has make cloud computing paradigms encounter major challenges which includes jitter, extreme latency, non-encouraging location- cognizance and mobility. In addressing such challenges, edge computing paradigm and related architecture have been developed to move digital services from the central cloud to network edge. Observing its disruption, IoT connected devices are forecasted to reach 500 billion by the year 2030, even as the world mobile traffic is anticipated to be in the rise by 2022. Given the advantages of the edge computing paradigm, the edge devices are to some extend resource constrained were an efficient resource management is essential to make edge computing a reality. The study proposed a framework for efficient resource management at the network edge based on deep reinforcement learning. The experimental results reveals that the EECFRM framework based on the TD3 algorithm converges in comparison with the DDPG based algorithm as presented.*

**Keywords:** Internet of Things, Edge Computing, Resource Management, Scheduling and Deep Reinforcement Learning.

## 1.0 INTRODUCTION

Several embedded gadgets that can interact with the environment are already connected to internet where it is estimated that the number will increase to 75 billion by the year 2025 Hassan *et al*., (2019). The Institute of Electrical and Electronics Engineering (IEEE) defines the internet of things (IoT) as a framework in which all things have a depiction and existence in the Internet that allows the communication between things and applications in the cloud. IoT is actually the notion of daily objects which ranges from industrial machines to wearable gadgets utilizing embedded sensors to collect data and act on it via the network Safavat *et al.,* (2020).

IOT as an emerging technology allows numerous objects to be linked to the Internet which consists of wearable equipment, actuators, sensors, smart vehicles, and smart buildings. It links pervasive computing and allows things to be sensed and controlled remotely. IoT is envisioned to improve efficiency and accuracy, and as well as decrease costs, and it is presently influencing basic roles in a number of promising services which includes smart cities, smart grids and intelligent transportation systems Majd S. Ahmed (2021) and Tang (2020).

A number of this gadgets already have sensing capabilities that is use to sense and collect data from the environment around us and as well share the data across the Internet where it can be processed for different purposes. The connection of such a huge number of physical objects with sensing abilities to the Internet have given rise to the idea of big data which needs efficient and smart storage where its analysis can be the requirement for the design and planning of sustainable smart cities Dechouniotis *et al*., (2020) and Tang (2020). Looking at its disruption, IoT enable devices is predicted to reach 500 billion by the year 2030 with increase in global data also to reach 175ZB by 2025 Zeyu *et al.,* (2020), even as the world mobile traffic is anticipated to be in the rise by 2022 Hyungsup *et al.,* (2020).

However, with the major breakthroughs in aspects of hardware advances and processing capabilities of the devices, the IoT devices cannot assure the needed high performance as regard mission or time critical IoT applications Pratap *et al.,* (2020).

Since its introduction, Cloud computing has play major role in rising the visibility and abilities of computing, storage, and networking infrastructure to the applications. It has been described as a model that encourages universal, on demand network access to shared computing resources worldwide Zahmatkesh and Al-turjman (2020).

Over the years, the computation load and data volume in cloud have been on the rise where results of computations and control data are reliably moved to the centralize data centers and network hubs. With this achievement, cloud computing encounters challenges such as increasing latency Muniswamaiah *et al*, (2021). Nevertheless, this makes the latency even worse as a result of the overhead caused by inter-cloud communications.

Edge computing as a paradigm has become a crucial solution to address the problems of emerging technologies such as cloud and IOT by decreasing data transmission rate, service latency enhancement and decreasing the pressure linked with cloud computing. Edge computing paradigm has become a vital complement to the cloud by substituting the cloud role in certain scenarios Wang (2020). In edge computing scenarios, the raw data produced by edge devices are usually process at the edge rather than been send to the remote cloud where they are processed locally at the edge and only selected data with perceptions are transferred to the cloud server Mohammed et al., (2021) and Wang (2020).

Given their advantages, edge computing are to some degree resource constrained where an efficient resource management is vital making edge computing a reality. Devices in the edge usually have less resources in comparison to the cloud computing services which comprises energy, processing power, and storage resources Fu et al., (2020) and Giancarlo et al., (2021). The aim of this paper is to propose a framework for efficient resource management at the network edge based on deep reinforcement learning.

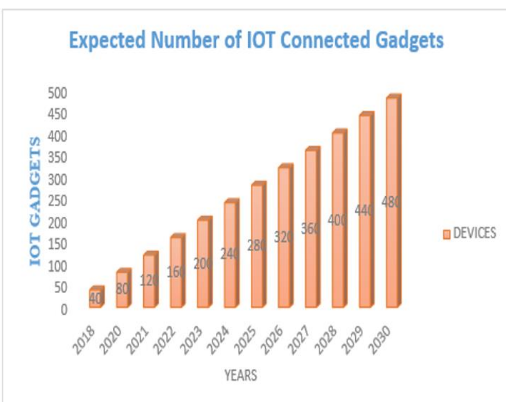



**Figure 1: Expected Number of IOT Connected Gadgets Dechouniotis *et al*., (2020)**

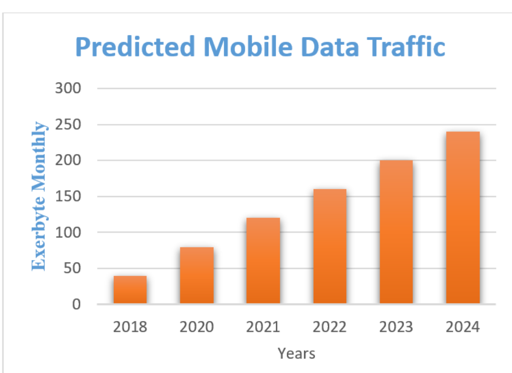



**Figure 2: Mobile Data Traffic Prediction Toczé and Nadjm (2018)**

## 2.0 RELATED WORKS

Edge computing, as an emerging paradigm offers alternative and promising platform to process tasks by exploring various resources at the network edge. Unlike centralized cloud, edge are distributed in different positions, primarily allowing performance and cost efficiency enhancement by computing data where it is generated. Due to the limited resources associated with edge computing devices several techniques have been studied to manage resources at the network edge.

Notable work in that regard includes that of Li *et al.,* (2019), Wang *et al.,* (2019), Zhang et al., (2019), Alqerm and Pan (2020). Although several of the research interest was towards resource allocation but task scheduling is rarely studied. In Tan et al., (2017), an online scheduling model was proposed to reduce response time of task when it is offloaded to the edge servers. Similarly, a technique based on Lyapunov optimization, was proposed by Zhang *et al.,* (2019) to reduce the communication delay and computation delay in the network. Chen et al., (2019) proposed a dual-scheduling mechanism for heterogeneous vehicular edge computing system of servers and arrival rate of task. Chiang *et al.,* (2019) proposed a parallel task offloading and scheduling scheme based on first-come-first serve (FCFS) technique to lessen the average completion via a mixed integer non-linear programming (MINLP) algorithm. Li et al., (2019) proposed a scheme based on shortest job first (SJF) scheduling technique where the task with less delay is scheduled earliest. In Saleem *et al.,* (2021) the authors proposed a Mobility Aware

Joint Task Scheduling and Resource Allocation framework for Cooperative Mobile Edge Computing for task offloading. In Meng *et al.,* (2020), a task scheduling and dispatching of networking and computing resources was proposed with the aim of maximizing the number of completed tasks. These methods are generally based on mathematical model enhanced by mixed-integer non-linear programming (MINLP) and some heuristic algorithms.

Even thou the model based algorithms can realise good performance, they are not well suited for the dynamic nature of several IOT environment where the state space is large and continuous in nature.

Furthermore, relevant research efforts have been expended in order to develop DRL methods for continuous control such as deep policy gradient (DPG) method that was introduced in Mocanu *et al.,* (2019) and Meng *et al.,* (2020). DPG applies a DNN to precisely estimate the action selection probability for a given state instead of approximating the Q-value function of taking an action in a given state. The DPG is often criticized for its low sampling efficiency and the high variance in its gradient estimates, which lead to slow convergence. To overcome this drawback, Wan *et al.,* (2018) and Ye et al., (2020) utilised the deep deterministic policy gradient (DDPG) algorithm in order to approximate the schedules of diverse electronics. The DDPG algorithm is an actor critic method that approximates the policy and Q-value during the training process. As such, the DDPG primarily lessens the variance in the gradient approximation and also contributes good convergence performance. Never the less, the main problem of the DDPG algorithm is the overestimation of the Q-value function resulting to sub-optimal policies Fujimoto *et al*., (2018).

## 3.0 DESIGN

In achieving the objectives of an edge computing system, an efficient edge computing framework for resource management (EECFRM) is proposed based on Deep Reinforcement Learning for efficient resource scheduling and management so as to achieve low latency and optimize the energy constraint of edge devices. The architecture in figure 3 consist of three layers, the least layer is the layer with the various IOT delay sensitive applications which usually consists of sensors and actuators. The middle layer consists of edge servers that carries out the analytics and also running the DRL algorithms for effective and efficient management of the edge resources, and the cloud layer is the last layer usually for long time storage.

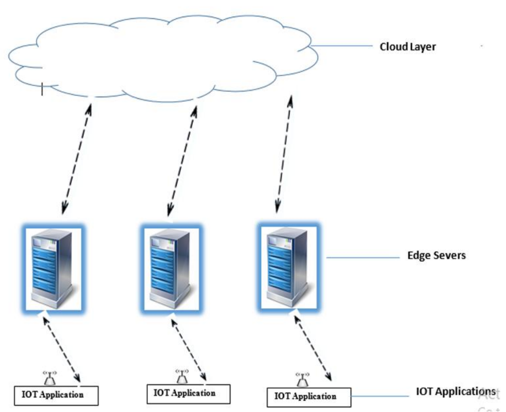



**Figure 3. EECFRM Architecture**

### 3.1 Deep Reinforcement Learning

Deep reinforcement learning (DRL) is a branch of artificial intelligence designed to imitate human thinking. DL acquires the system information by perceiving the environment and make available system state information for the present environment. The RL then links the present state information to the corresponding action and then approximate the values using the predictable accumulated return Li *et al.*, (2018). The basic structure of Deep RL is displayed in Figure.4.

The Deep reinforcement learning architecture given in figure 4 shows an agent that relates with an environment where at each time step t, the agent observes some state $s_t$, and is request to take an action $a_t$. In taking the action, the state of the environment transitions to $s_{t+1}$ and thereby resulting in the agent receiving a reward $r_t$. The state transitions and rewards are said to be stochastic and are assumed to have the Markov property, which indicates that the state transition probabilities and rewards depend only on the state of the environment $s_t$ and the action taken by the agent $a_t$. It is crucial to note that the agent can only control its actions and has no prior knowledge of which state the environment would transition to or what the reward may be.

The goal of learning is to maximize the expected cumulative discounted reward where the agent picks actions based on a policy which is defined as a probability distribution over actions $\pi : \pi(s, a) \rightarrow [0, 1]$; where $\pi(s, a)$ is the probability that action a is taken in a state Mao *et al*., (2016).

The Deep reinforcement learning as a machine learning approach links artificial neural networks alongside reinforcement learning techniques to enable agents in software form learn the best possible actions in a simulated environment so as to achieve their aim. Even though neural networks are liable for current AI innovations in areas such as time series prediction, computer vision and machine translation, they can as well be join with reinforcement learning algorithms to generate innovative games such as the Deep mind's AlphaGo game, which is an algorithm that thrashed the Go board game world champions Varghese, (2019).

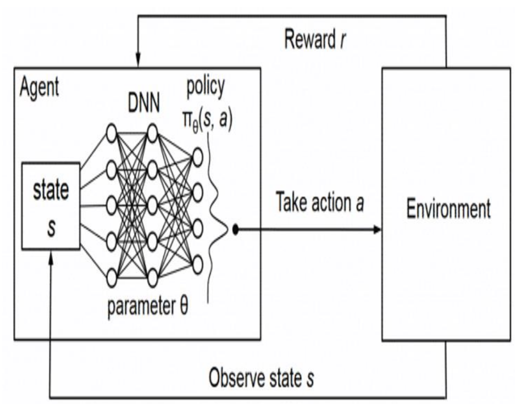



**Figure 4: Deep Reinforcement Learning Architecture Philip, (2018)**

A typical deep reinforcement learning approach is characterised by the following terminologies:

**Action** (A): This comprises all the likely moves that the agent can receive

**State** (S): The present condition returned by the environment.

**Reward** (R): This is the response by which we measure the success or failure of an agent's actions

in a given state. The response could be either positive or negative as the case maybe.
**Policy** ($\pi$): This is the approach use by the agent to decide the next action based on the present state.
**Value** (V): This is the predictable long term return with discount. It is given as V π(s) been the expected long term return of the present state under policy π.
**Q-value** or action-value (Q): The Q value is similar to Value, except that it takes an additional parameter, which is the current action a. Q π (s, a) refers to the long term return of the current state (s), taking action a under policy π.
**Environment**: This is the surrounding in which the agent acts, and which reacts to the agent. The agent's present state and action is normally taken as input by the environment and give back as output the agent's reward and its next state Varghese, (2019).

The main objective of RL in an environment is to enable an agent take the best action in the current state so as to maximize the long term goal, whereby the interaction between the agent's action and state via the environment is often modelled as a Markov Decision Process (MDP) Toczé and Nadjm **(2018)**.

The adopted algorithm for this research is the twin delayed deep deterministic policy gradient (TD3) algorithm which is an enhancement of deep deterministic policy gradient (DDPG) involving three critical parameters. This includes Clipped double *Q*-learning for actor-critic, Target networks and delayed policy updates and Targ **Equation (6)** smoothing regularization: Add noise to the target action to smooth the *Q*-value function and avoid overfitting.

In the first technique, The TD3 algorithm incorporates the idea of double *Q*-learning in DDPG where it establishes two *Q*- **Equation (7)** to compute the value of the next state as seen in equation 1 and 2 (Hao *et al*., 2020).

$$Q_{\theta_1'}\left(s', a'\right) = Q_{\theta_1'}\left(s', \pi_{\phi_1}(s')\right) \quad \textbf{Equation (1)}$$

$$Q_{\theta_2'}\left(s', a'\right) = \quad \textbf{Equation (8)} \quad s')) \quad \textbf{Equation (2)}$$

The minimum of any of equation one or two is then used to compute the Bellman equation given in equation (3)

$$Y_1 = r + \gamma \min_{i=1,2} Q_{\theta_i'}\left(s', \pi_{\phi_1}(s')\right) \quad \textbf{Equation (3)}$$

As deep function approximators needs multiple gradient updates to converge, target networks offers a stable objective in the learning procedure and enables better coverage of the training data. TD3 algorithm only updates the policy and target networks after a fixed number of updates *d* to the critic. The less frequent policy updates can make the update of *Q*-value function has a smaller variance, and thus a higher quality policy can be obtained (Hao *et al*., 2020).

In solving the problems associated with the DDPG overestimation issues as given in equation **(4)**,

$$y = r + \mathbb{E}_{\epsilon}\left[Q_{\theta'}\left(s', \pi_{\phi'}(s') + \epsilon\right)\right] \quad \textbf{Equation (4)}$$

In adding a truncated normal distribution noise to each action as a regularization, the computation of *Q*-values can be smoothed to avoid overfitting resulting in an enhanced target update given by **Equation 5**

$$y = r + \gamma Q_{\theta'}\left(s', \pi_{\phi'}(s') + \epsilon\right), \epsilon \sim clip(N(0,\sigma), -c, c \quad \textbf{Equation (5)}$$

## 3.2 EECFRM and DRL Formulation

The problem of scheduling and offloading decisions by the edge servers can be modelled as a Markov Decision Process scheme where the aim is to find the optimal policy denoted as π which in turn reduces the tasks penalties. In line with this, we define the following terminologies:
**State Space**: The state space is characterised by the state of each edge server and the corresponding set of tasks given as $S_e = \{s_1, s_2, s_3 \ldots\ldots\ldots s_k\}$, where k is the number of edge servers in the network. The task queue length of the edge server is $q_i$ with the task rank given as $T_r = \{n1, n_2, n_3 \ldots\ldots\ldots n_i\}$

where i is the total number of tasks.
**Action Space**: usually on arrival of a task i, actions needs to be taken to determine which edge server is free and appropriate to carry out the task, and is given as a ={1,2,3…………k)

**Reward:** At any point in time in DRL, the agent always gets a reward which can be positive or negative. The goal is to minimize the average timeout and slowdown of the task in the queue length. As such, the reward is given as R= $\frac{1}{ft(S,a,\pi)} + \frac{1}{ln(EC,a,\pi)}$

where $f_t$ and $E_C$, are the feedback time and energy consumption rate respectively.

## 3.3 EECFRM Algorithm Based on TD3

In this section, we develop an algorithm based on the Twin delayed Deep Deterministic Policy Gradient for the EECFRM. It is expected that a given edge server completes a task within a specified time frame as soon as possible leading to less latency. As such, a delay in task computation will result in the agent getting a negative reward based on equation **(8)**

## 4. Results and Discussion

This section examines the results of the simulation carried out to verify the performance of the proposed framework based on deep reinforcement learning algorithm for efficient resource management at the network edge. The experiment was carried out on a system with Core i5 CPU with speed of 2.60GHz, 8GB of RAM and 64bit operating system based processor. The reinforcement learning tool used for agent training was matlab reinforcement learning tool box. Other parameters used for the experiment includes number of Episodes set at 5000, max steps per Episode set at 300, discount Factor was given as 0.995, average Window Size set at 100, training termination value set at 250. After the training of about 5000 episode the proposed EECFRM framework based on the TD3 algorithm converges in comparison with the DDPG based algorithm as shown in figure 5.

**EECFRM Algorithm Based on TD3**

1. Generate Task sets
2. Initialised critic Networks $Q_{\theta 1}$, $Q_{\theta 2}$ and actor network $\pi\phi$ with parameters $\theta_1$ ,$\theta_2$ and $\phi$
3. Initialise target networks $\theta'_1 \leftarrow \theta_1, \theta'_2 \leftarrow \theta_2$ $\phi' \leftarrow \phi$
4. Initialise Replay Buffer (B)
5. For each episode j = 1 to M do
6. Select action with exploration noise $a \sim \pi\phi(s) + \varepsilon$, $\varepsilon \sim N(0,\sigma)$ and observe reward r (**Equation 8**) and new state s′ (**Equation 7**)
7. Store transition (s, a, r,s′) in B
8. Sample minibatch of n transitions (s, a, r,s′) from B
   $\tilde{a} \leftarrow \pi\phi'(S') + \varepsilon$, clip $(N(0,\tilde{\sigma}\text{-c}, c)$
   $Y \leftarrow \gamma + \gamma \min_{i=1,2} Q_{\theta 1}(s',\tilde{a})$ e (Using minimum of **equation** 1 or |**2**)
   Update critics $\theta_i \leftarrow \mathrm{argmin}_{\theta i}\, N^{-1} \Sigma(y - Q_{\theta i}(s, a))^2$
   If j mod d then
9. Update $\phi$ by the deterministic policy gradient
   $\nabla\phi J(\phi) = N^{-1} \Sigma \nabla_a Q_{\theta i}(s, a)|_{a=\pi\phi(s)} \nabla_\phi \pi\phi(s)$
   Update target networks
10. $\theta^{1}_{i} \leftarrow T\theta_i + (1 - T)\,\theta'_i$
    $\theta' \leftarrow T\phi + (1 - T)\,\phi'$
11. end if
12. end for

**Figure 5. Training convergence of proposed Framework EECFRM based TD3 Algorithm**

For the task scheduling, a set of tasks to be processed by the edge servers was considered where each task has an identified processing time. The goal of which is to minimise the computation time by edge servers taken to process the different tasks. It can be observed from the figure 6 that the task cumulative measure based scheduling and offloading proposed algorithm increases with increment of the task arrival measure. This is as a result of the fact that the higher the arrival measure shows that more task will be in queue to be processed by the edge servers at a specified period thereby raising the task awaiting period. Thus, with the increase in the number of the edge servers shows that more tasks can be assigned to available servers resulting in the waiting time reduction.

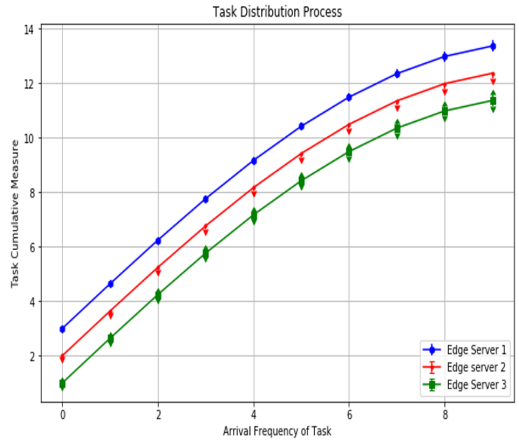
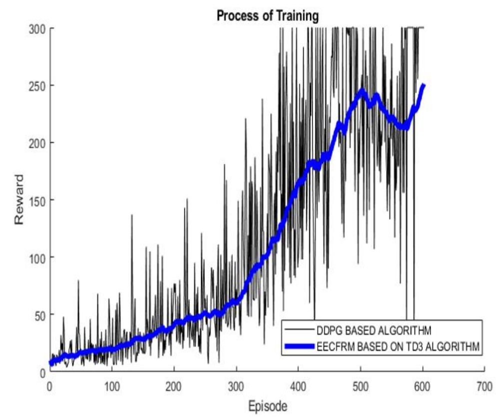



**Figure 6. Task Cumulative Measure against Arrival Frequency and Edge servers**




## 5. RECOMMENDATIONS

Edge computing as a paradigm has become a crucial solution to address the problems of emerging technologies such as cloud and IOT by decreasing data transmission rate, service latency enhancement and decreasing the pressure linked with cloud computing. Edge computing has become a vital complement to the cloud by substituting the cloud role in certain scenarios. In edge computing scenarios, the raw data produced by edge devices are usually process at the edge rather than been send to the remote cloud where they are processed locally at the edge and only selected data with perceptions are transferred to the distant cloud server.

Given their advantages, edge computing are to some degree resource constrained where an efficient resource management is vital making edge computing a reality. Devices in the edge usually have less resources in comparison to the cloud computing services which comprises energy, processing power, and storage resources. To efficiently manage the resources at the network edge, it is recommended that organisations with internet of things based delay sensitive data that required immediate transmission employed the use of deep reinforcement learning for optimal performance at the network edge.

## 6. CONCLUSION AND FUTURE WORK

IOT with its emergence enables huge number of devices to be linked to the Internet which comprises wearable equipment, sensors, actuators, smart vehicles, and smart buildings. It links pervasive computing and allows devices to be sensed and controlled remotely. Normally, the Internet has transformed the way and manner we see and consume information and also how we interact with one another. A number of household devices are being connected to the Internet via state of the art innovations in processing power and speed, broadband network, storage and sensor technologies.

Cloud computing in the other hand since its beginning has perform critical role in rising the visibility and abilities of computing, storage and networking infrastructure to the applications. It has been considered as a model that stages universal, on demand network access to shared computing resources worldwide.

The edge computing given their advantages are to some extend resource constrained, as such, an efficient resource management is essential to make edge computing a reality.

The study hence, proposed a framework for efficient resource management at the network edge based on deep reinforcement learning. The experimental results reveals that the EECFRM framework based on the TD3 algorithm converges in comparison with the DDPG based algorithm as presented. As a future work, the various security challenges of the edge computing applications and their possible mitigation strategies will be carried out using reinforcement learning approach.

## REFERENCES

Alqerm, I.; Pan, J. (2020). Enhanced Online Q-Learning Scheme for Resource Allocation with Maximum Utility and Fairness in Edge-IoT Networks. *IEEE Trans. Netw. Sci. Eng*, 7, 3074–3086.

Chen, X. Thomas, N. Zhan, T. and Ding, J. A (2019). Hybrid Task Scheduling Scheme for Heterogeneous Vehicular Edge Systems. *IEEE Access*, 7, 117088–117099

Chiang, Y., Zhang, T., and Ji, Y. (2019). Joint Cotask-Aware Offloading and Scheduling in Mobile Edge Computing Systems. *IEEE Access*, 7, 105008–105018

Dechouniotis, D. Athanasopoulos, N. and Leivadeas, A. (2020). "Edge Computing Resource Allocation for Dynamic Networks : The DRUID-NET Vision and Perspective," *Sensors*. 20(2191), pp. 1–18.

Fu, S. Yang, F. and Xiao, Y. (2020). AI inspired intelligent resource management in future wireless network, *IEEE Access*, vol. 8, pp. 22425–22433.

Fujimoto, S. Van Hoof, H. and Meger, D. (2018). Addressing function approximation error in actor-critic methods. In Proceedings of the Machine Learning Research.

Giancarlo Fortino, MengChu Zhou, Mohammad Mehedi Hassan, Mukaddim Pathan, Stamatis Karnouskos, (2021). Pushing Artificial Intelligence to the Edge: Emerging trends, issues and challenges, Engineering Applications of Artificial Intelligence, Volume 103, 2021, 104298, ISSN 0952-1976, https://doi.org/10.1016/j.engappai.2021.104298.

Hao Dong, Zihan Ding and Shanghang Zhang (2020). Deep Reinforcement Learning: Fundamentals, Research and Applications Springer Nature Singapore Pte Ltd. 2020. All Rights Reserved.

Hassan, M.U., Rehmani, M.H., and Chen, J. (2019). Privacy preservation in blockchain based IoT systems: integration, issues, prospects, challenges, and future research directions. *Future Gener. Comput. Syst.* **97**, 512–29.

Hyungsup Shin, Jiyeon Jung, Yoonmo Koo (2020). Forecasting the video data traffic of 5 G services in south korea, *Technological Forecasting and Social Change*,153, 119948, https://doi.org/10.1016/j.techfore.2020.119948.

Li, C. Tang, J. Tang, H. and Luo, Y. (2019). Collaborative cache allocation and task scheduling for data-intensive applications in edge computing environment. *Future Gener. Comput. Syst.*, 95, 249–264.

Li, Y.J. Chang, H.Y. Lin, Y.J. Wu, P W. FrankWang, Y.C (2018). Deep reinforcement learning for playing 2.5 D fighting games, in: 2018 25th IEEE International Conference on Image Processing, ICIP, 2018, pp.3778–3782.

Li, Z., Yang, Z., Xie, S., Chen, W., and Liu, K. (2019). Credit-Based Payments for Fast Computing Resource Trading in Edge-Assisted Internet of Things. *IEEE Internet Things J*, 6, 6606–6617.

Majd S. Ahmed (2021). Designing of internet of things for real time system, Materials Today: Proceedings, 2021, 2214-7853, https://doi.org/10.1016/j.matpr.2021.03.527.

Mao, H. Alizadeh, M. Menache, I. and Kandula, S (2016). Resource Management with Deep Reinforcement Learning, pp. 50–56.

Meng, F. Chen, P. Wu, L. Cheng, J. (2020). Power Allocation in Multi-User Cellular

Networks: Deep Reinforcement Learning Approaches. IEEE Trans. *Wirel. Commun*., 19, 6255–6267.

Meng, J. Tan, H. Li, X. Han, Z. Li, B. (2020). Online Deadline-Aware Task Dispatching and Scheduling in Edge Computing. *IEEE Trans. Parallel Distrib. Syst.*, 1270–1286.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature*, 518, 529–533.

Mocanu, E. Mocanu, D.C. Nguyen, P.H. Liotta, A. Webber, M.E. Gibescu, M. Slootweg, J.G.(2019). On-line building energy optimization using deep reinforcement learning. IEEE Trans. *Smart Grid*, 10, 3698–3708.

Mohammed Laroui, Boubakr Nour, Hassine Moungla, Moussa A. Cherif, Hossam Afifi, Mohsen Guizani (2021). Edge and fog computing for IoT: A survey on current research activities & future directions, Computer Communications,180, pp. 210-231, https://doi.org/10.1016/j.comcom.2021.09.003.

Muniswamaiah, M. Agerwala T. and Tappert, C.C (2021). A Survey on Cloudlets, Mobile Edge, and Fog Computing," *2021 8th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2021 7th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)*, pp. 139-142, doi: 10.1109/CSCloud-EdgeCom52276.2021.00034

Philip Ossenkopp (2018). An article on Deep Reinforcement Learning, Available at https://www.novatec-gmbh.de/en/blog/deep-q-networks/

Pratap, R Javaid, M. Haleem, A. and Suman, R.(2020). Internet of things ( IoT ) applications to fight against COVID-19 pandemic, *Diabetes Metab. Syndr. Clin. Res. Rev.*, 14(4), 521–524, 2020.

Remani, T. Jasmin, E. Ahamed, T.I. (2019). Residential load scheduling with renewable generation in the smart grid: A reinforcement learning approach. *IEEE Syst. J.* **2019**, 12, 3283–3294.

Safavat, S. Sapavath, N. N. and Rawat, D. B.(2020). Recent advances in mobile edge computing and content caching, *Digit. Commun. Networks*, 6(2), pp. 189–194.

Saleem, U. Liu, Y. Jangsher, S. Li, Y. and Jiang, (2021). Mobility-Aware Joint Task Scheduling and Resource Allocation for Cooperative Mobile Edge Computing. *IEEE Trans. Wirel. Commun*, 20, 360–374.

Tan, H. Han, Z. Li, X. and Lau, F.C.M. (2017). Online job dispatching and scheduling in edge-clouds. In Proceedings of the IEEE INFOCOM, *IEEE Conference on Computer Communications*, Atlanta, GA, USA, pp. 1–9.

Tang L. and Hu, H. (2020). Computation Offloading and Resource Allocation for the Internet of Things in Energy-Constrained MEC-Enabled HetNets, *IEEE Access*, 8, pp. 47509–47521.

Technological Forecasting and Social Change, Volume 153, 2020, 119948, ISSN 0040-1625, https://doi.org/10.1016/j.techfore.2020.119948.

Toczé K. and Nadjm-Tehrani, S. (2018). A Taxonomy for Management and Optimization of Multiple Resources in Edge Computing, *Wirel. Commun. Mob. Comput.*, pp. 1–23.

Tsang, N.; Cao, C.;Wu, S.; Yan, Z.; Yousefi, A.; Fred-Ojala, A.; Sidhu, I. Autonomous Household Energy Management Using Deep Reinforcement Learning. In Proceedings of the 25th IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC), Valbonne Sophia-Antipolis, France, 17–19 June 2019; pp. 1–7.

Varghese, B (2019). Resource Management in Fog / Edge Computing : A Survey on Architectures , Infrastructure , and Algorithms, 52(5), 2019.

Wan, Z. Li, H. and He, H. (2018). Residential energy management with deep reinforcement learning. In Proceedings of the 2018 *International Joint Conference on Neural Networks* (IJCNN), Rio de Janeiro, Brazil, 8–13, 2018; pp. 1 - 7.

Wang, K (2020). Migration strategy of cloud collaborative computing for delay-sensitive industrial IoT applications in the context of intelligent manufacturing," *Comput. Commun.*, vol. 150, no. October 2019, pp. 413–420.

Wang, P., Yao, C., Zheng, Z. Sun, G. and Song, L. (2019). Joint Task Assignment, Transmission, and Computing Resource Allocation in Multilayer Mobile Edge Computing Systems. *IEEE Internet Things J*., 6, 2872–2884.

Ye, Y. Qiu, D. Wu, X. Strbac, G. and Ward, J. (2020). Model-Free Real-Time Autonomous Control for A Residential Multi-Energy System Using Deep Reinforcement Learning. IEEE Trans. *Smart Grid* **2020**, 11, 3068–3082.

Zahmatkesh H. and Al-turjman F (2020). Fog computing for sustainable smart cities in the IoT era : Caching techniques and enabling technologies - an overview, *Sustain. Cities Soc.*, 59. *Cities Soc.*, vol. 59, p. 102 - 139.

Zeyu, H. Geming, X. Zhaohang W. and Sen, Y (2020). Survey on Edge Computing Security, *International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE)*, Fuzhou, China, pp. 96-105, doi: 10.1109/ICBAIE49996.2020.00027.

Zhang, J. Hu, X. Ning, Z. Ngai, E.C. Zhou, L. Wei, J. Cheng, J. Hu, B. Leung, V.C.M. (2019). Joint Resource Allocation for Latency-Sensitive Services over Mobile Edge Computing Networks with Caching. *IEEE Internet Things J*., 6, 4283–4294.

Zhang, Y. Du, P. Wang, J. Ba, T. Ding, R. Xin, N. (2019). Resource Scheduling for Delay Minimization in Multi-Server Cellular Edge Computing Systems. *IEEE Access*, 7, 86265–86273.