IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 11 October 2022, accepted 6 November 2022, date of publication 14 November 2022, date of current version 23 January 2023.

Digital Object Identifier 10.1109/ACCESS.2022.3221740

RESEARCH ARTICLE

Deep Reinforcement Learning Based Edge Computing Network Aided Resource Allocation Algorithm for Smart Grid

YINGYING CHI[1], YI ZHANG[2], YONG LIU[1], HAILONG ZHU[3], ZHE ZHENG[1], RUI LIU[1], AND PEIYING ZHANG[2]

[1]Beijing Smartchip Microelectronics Technology Company Ltd., Beijing 100192, China
[2]College of Computer Science and Technology, China University of Petroleum (East China), Qingdao 266580, China
[3]State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China

Corresponding authors: Hailong Zhu (zhuhl@bupt.edu.cn) and Yi Zhang (zhangyi.upc@qq.com)

This work was supported in part by the Academician Expert Open Fund of Beijing Smartchip Microelectronics Technology Company Ltd.; and in part by the Shandong Provincial Natural Science Foundation, China, under Grant ZR2020MF006.

ABSTRACT The dramatic increase in the volume of users and services makes scheduling network resources for smart grids a key challenge. Network slicing is an important technology to solve this problem. We introduce edge computing networks into the smart grid to intelligently allocate resources based on users' quality of service (QoS) and available resources. However, existing heuristic resource scheduling algorithms often lead to resource fragmentation and thus fall into local optima. To this end, we propose a deep reinforcement learning (DRL)-based virtual network embedding algorithm to optimize the resource allocation strategy of smart grids from a network virtualization perspective. We extract the network properties of the smart grid to construct a policy network as a training environment for DRL agents. Finally, DRL derives the probability of each node being embedded based on the extracted attributes of edge computing nodes and completes user request (UR) embedding based on this probability. The experimental results show that the algorithm proposed in this paper has excellent performance with guaranteed low latency, 21% improvement in long-term revenue and 5.6% improvement in UR success rate compared with the other two algorithms.

INDEX TERMS Deep reinforcement learning, delay sensitive, edge computing network, resource allocation, smart grid, user request.

NOMENCLATURE

m, n	Index of the physical nodes.
u, v	Index of the virtual node.
L_i^S	The ith edge link.
$BW(L_i^S)$	Bandwidth resources owned by link L_i^S).
$CPU(N_m^S)$	CPU resources owned by the mth edge node.
$CPU_r(N_m^S)$	Current remaining CPU resources of the m^{th} edge node.
$D(N_m^S)$	Delay attribute of the mth edge node.
$M(N_m^S)$	Memory resources owned by the mth edge node.
$P(CPU)$	Unit price of CPU resources.
$hop(L_i^r)$	Number of hops of link L_i^r.
$num(G_{succ}^R)$	Number of successfully embedded user requests.
γ	Discount on awards received by the agent.
G_k^R	The k^{th} user request.
O_k	Available resource vector of the kth edge node.
v_k	Feature vector of the kth node.

The associate editor coordinating the review of this manuscript and approving it for publication was Gab-Su Seo.

I. INTRODUCTION

Smart grids are a potential solution to the growing conflict between greenhouse gas emissions and growing energy demand. The smart grid uses modern information and

communication technology to achieve a secure, coordinated, clean, and intelligent development of the grid and to provide reliable power security for society [1]. In the process of transforming the traditional power grid to the smart grid, the power grid business develops in the direction of distributed automation, power internet of things, artificial intelligence, and the number of power grid business and user terminals is increasing [2]. It is a great challenge to meet the diverse service demands in the smart grid. Edge computing is an effective way to solve this problem, which can offload massive computing tasks to edge servers for efficient execution.

Edge computing is different from cloud computing [3]. Its computing power is deployed near the device side, so the device request can be responded with low delay. At the same time, the data acquisition terminal of edge computing can also reduce the impact of site bandwidth limitation. The emergence of edge computing technology alleviates the pressure of core network. Storage, computing and other services can be distributed to the edge network [4], [5]. The core network can be divided into several domains. The services of different domains can be distributed to the same edge network [6]. The nodes of edge computing are responsible for managing the transactions in the domain. By unified management of all nodes, the interconnection between various domains can be realized.

In the future, the network will have more stringent requirements for the storage of massive data, and will have higher and higher requirements for cost, bandwidth and delay [7]. Traditional heuristic algorithms are prone to fall into local optimal solutions when the data volume is large and cannot meet the requirements for differentiated quality of service in edge network-based smart grids [8], [9]. With the improvement of computer computing power, deep reinforcement learning (DRL) shows excellent performance in solving high-dimensional spatial problems. For this reason, many researchers have devoted themselves to finding the optimal resource allocation scheme with DRL. Using unmanned aerial vehicle (UAV)-assisted task offloading, the authors of literature [10] proposed a DRL-based collaborative computational offloading solution for UAVs to precisely match the service requirements of ground power devices. Considering the dynamic and continuous nature of computational tasks, a DRL-based dynamic resource management algorithm is proposed in literature [11], which effectively reduces the long-term average delay. Literature [12] and [13] explored the dynamic nature of mobile edge computing networks, using DRLs to solve the computational task offloading and resource allocation problem. To reduce energy consumption, literature [14] proposed a DRL-based online algorithm for time-varying channel scenarios to find a near-optimal task offloading scheme. However, for smart grid scenarios, existing edge computing technologies are difficult to satisfy the differentiated quality of service (QoS) requirements of massive devices.

To cope with the differentiated QoS requirements of users, resource allocation based on network slicing has become a

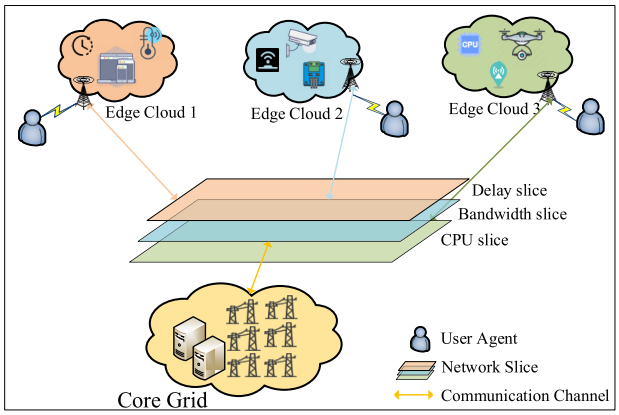

FIGURE 1. Network resource allocation in smart grid aided by edge computing networks.

popular research topic [15]. A network slice is a collection of logical network functions that provides end-to-end connectivity for a specific service by virtually slicing the physical network [16], [17]. As shown in Figure 1, edge service providers divide different types of access network slices according to the characteristics and requirements of application scenarios. For example, delay slicing is used to provide services for users with strict requirements on delay, which can provide user terminals with fast content access and retrieval functions and reduce content delivery latency and network transmission load [18]. The core issue of network slicing is virtual network embedding (VNE), which is essentially the orchestration of network resources. Therefore, the resource allocation problem of the smart grid can be converted into the study of the VNE algorithm.

For the appeal problem, we design a DRL-based edge network resource scheduling scheme for smart grid scenarios. The specific contributions of this paper can be summarized as follows:

1) To address the differentiated QoS requirements of users, we introduce edge computing and network slicing into the smart grid, and model the edge computing-assisted smart grid as a multi-domain virtual network model to solve the smart grid resource allocation problem from the perspective of virtual network embedding.
2) We describe the virtual network embedding as a Markov decision process (MDP), use the self-built policy network as an agent, extract the network resource attributes to construct the feature matrix, and derive the embedding probabilities of the nodes to complete the whole VNE algorithm.
3) We experimentally validate the performance of the proposed algorithm, and the experimental results show that the algorithm performs well in terms of long-term average revenue and user request (UR) success rate.

The reminder of this article is arranged as follows. In the second section, the research status of edge computing networks, multi-domain virtual networks and resource allocation

algorithms based on machine learning are reviewed. The third section introduces the model of the system, including network model and user request model. The fourth section introduces the reinforcement learning model based on policy network and its algorithm implementation. In the fifth section, the performance of the algorithm is evaluated and the results are analyzed. The sixth section summarizes the full text.

II. RELATED WORK

Since the concept of edge computing was put forward, it has attracted extensive scholars' attention [19]. Edge computing can be integrated with many fields, such as Internet of Thing (IoT), cloud computing, blockchain [20], etc. Integrating edge computing with other fields for resource allocation has always been a hot topic.

A. EDGE COMPUTING NETWORK

As the number of QoS-critical applications grows, edge computing faces a shortage of infrastructure resources. If additional infrastructure resources are needed to ensure QoS, further optimization of infrastructure management and resource management solutions is required. In the literature [21], to alleviate the high latency and network congestion problems in IoT, a method is proposed to periodically distribute incoming tasks in an edge computing network, increasing the upper limit of the number of tasks that can be processed simultaneously. In the literature [22], the authors proposed a resource allocation and task scheduling scheme based on service urgency priority to prioritize high-priority tasks and reduce task processing delays. In literature [23], a model for allocating computing resources in edge computing platform is proposed, which can ensure the quality of service and has high efficiency. The authors of the literature [24] proposed a power migration scaling algorithm that assigns user requests to the optimal server, which can better handle the dynamic load of the server while reducing energy consumption. The authors of the literature [25] proposed an analytical model for cloud-only, edge-only, and hybrid edge cloud processing. By using this model on edge hardware, web services can avoid geographic constraints and improve resource utilization.

B. VIRTUAL NETWORK EMBEDDING

With the development of virtual network technology, researchers have studied proposed virtual network embedding algorithms for resource allocation. Multi-domain virtual networks (VNs) need to span multiple infrastructure domains and the embedding cost is expensive. The literature [26] proposed a centralized virtual network embedding architecture for dynamically managing the network resources of each autonomous region, which can improve the division efficiency and reduce the embedding cost of the network. Literature [27] proposed a multi-controller-based multi-domain virtual network embedding algorithm that applies a particle swarm optimization algorithm to divide virtual network requests, which has a great advantage in reducing embedding cost. The authors of the literature [28] proposed a machine learning-based algorithm for virtual network functional resource demand prediction, which effectively improves the utilization of network resources. Literature [29] proposed a heuristic and machine learning-based approach to virtual machine allocation that offers significant advantages in terms of reduced energy consumption and execution time.

C. MACHINE LEARNING

Machine learning has a wide range of application scenarios [30]. The resource allocation between devices in cellular networks is studied in literature [31]. A joint resource allocation and power control algorithm based on Q-Learning is proposed in literature [32], which has good results in improving throughput. The authors of papers [33] and [34] integrate blockchain with network slicing techniques to ensure the security of network resource sharing. In addition, deep reinforcement learning methods are used to find the optimal resource pricing strategy to ensure the fairness of resource sharing. In the literature [35], the authors use reinforcement learning to allocate dynamic channel resources to address the vehicle communication delay problem.

We summarize the results obtained from the above research and make innovations. We model the edge computing network-assisted smart grid as a multi-domain virtual network model to address the resource allocation problem from a virtual network embedding perspective. The goal is to be able to meet the differentiated user requirements for QoS such as latency and bandwidth while improving internet service provider (ISP) revenue and user request success rates.

III. SYSTEM MODEL

In this part, we introduce the edge computing aided smart grid model, user request model and algorithm model in detail. In addition, the constraint conditions and evaluation metrics of the network resource allocation problem are presented.

A. EDGE COMPUTING NETWORK MODEL

A large number of edge computing servers are deployed in the network, forming an "edge" layer to provide nearest-end services to users. In addition, in the edge layer, resources such as computational resources, storage resources, and network bandwidth must be allocated for distributed data processing. To this end, we model the edge computing network as a weighted undirected graph $G^S = \{N^S, L^S, A^S\}$. N^S denotes the set of edge computing nodes, L^S denotes the set of links between nodes. In addition, it is worth noting that the node set $N^S = \{N_G^S, N_U^S\}$ contains two different types nodes, N_G^S and N_U^S represent power grid nodes and remote user nodes respectively, and the two types of nodes have different characteristics. $A^S = \{CPU(N^S), M(N^S), D(N^S), BW(L^S), P\}$ denotes the set of attributes of the edge computing network, where $CPU(N^S)$ denotes the computing resources of nodes, $M(N^S)$ denotes the storage resources of nodes, $D(N^S)$ denotes the processing delay of nodes, $BW(L^S)$ denotes the bandwidth resources

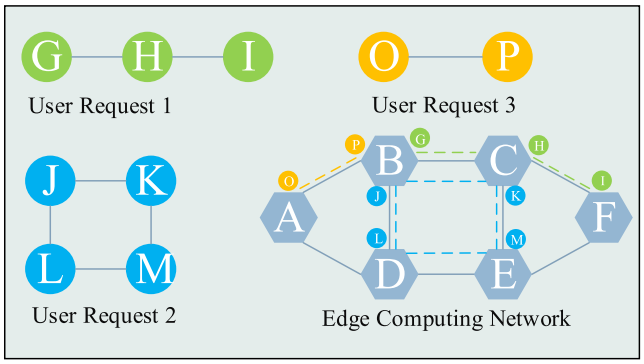

FIGURE 2. Example of an edge computing network and three user requests.

of links, and P represents the set of corresponding resource unit prices. $P = \{P(CPU), P(M), P(BW)\}$, $P(CPU)$, $P(M)$ and $P(BW)$ represent the unit price of CPU, memory, and bandwidth, respectively.

B. USER REQUEST MODEL

The edge network can accept service requests sent by multiple users at the same time. There is a communication relationship between different user requests, thus forming a user request network. Consequently, we model the user request network as a weighted undirected graph $G^R = \{N^R, L^R, A^R\}$. N^R represents the device node in the user request, and L^R represents the link set between the device nodes in the user request. $A^R = \{CPU(N^R), M(N^R), D(N^R), BW(L^R)\}$ represents the attribute set requested by users, where $CPU(N^R)$ represents the computing resources required by nodes, $M(N^R)$ represents the storage resources required by nodes, $D(N^R)$ represents the maximum acceptable delay of nodes, and $BW(L^S)$ represents the bandwidth resources required by links. Figure 2 shows three user requests with the different numbers of nodes and their embedding schemes.

C. MDP MODEL

We model the resource allocation problem as an MDP. In an MDP, the future state depends only on its current state and the action taken, independent of the previous state. We denote the MDP by the five-tuple $\{S, A, P, R, \gamma\}$.

1) STATE SPACE S

State ($s \in S$) represents the state space observable by the agent, including the usage of physical network compute, storage, and bandwidth resources, the network resources and performance limits required by current user requests, and the mapping relationships between deployed user requests and the physical network. The state $s_t \in S$ at moment t can be denoted as

$$S_t =< CPU_t(N_m^S), M_t(N_m^S), D_t(N_m^S), BW_t(L_i^S), P(R), G^R \uparrow G^S > . \quad (1)$$

In Equation (1), the network resource states of nodes $N_m^S \in N^S$ and links $L_i^S \in L^S$ change dynamically with time.

2) ACTION SPACE A

A indicates that the agent can select the action space. Action a ($a \in A$) indicates that the agent selects a physical node in the physical network that satisfies the constraints to deploy the next UR node according to a specific policy, thus completing the UR embedding. The action $a_t \in A$ at moment t can be denoted as

$$a_t =< A_{m,u}^{cpu}, A_{m,u}^{mem}, A_{m,u}^{delay}, A_{l_i^S,(u,v)}^{bw} >$$
$$m \in N^S, u \in N^R, l_i^S \in L^S, (u,v) \in L^R. \quad (2)$$

In Equation (2), $A_{m,u}^r$ denotes the action of deploying UR nodes in the edge computing network by consuming the corresponding resources $r \in CPU, M, BW$, and $A_{m,u}^{delay}$ indicates the edge computing node that processes the computational tasks of the UR nodes and generates delay.

3) REWARD FUNCTION R

Rewards ($r \in R$) represent the benefits obtained after completing one UR embedding. DRL agent aims to obtain the maximum cumulative reward R. The maximum cumulative reward is calculated by the following formula.

$$R = \sum_{i=1}^{|UR|} \sum_{t=1}^{T} \gamma \cdot r_t(s_t, a_t), \quad (3)$$

where γ represents a reward discount. This paper describes the reward function with the cost of resources.

The agent interacts with the network environment to complete the UR embedding. After getting the state space of the environment, the agent takes an action to complete the UR node embedding, and then returns the action taken to the environment and receives the corresponding reward. After getting the action from the agent, the environment updates the network resources to the next state and then passes the next state back to the agent.

D. CONSTRAINT CONDITIONS

Due to the limited network resources, the completion of user requests needs to satisfy certain resource constraints and performance constraints. In the edge network, physical nodes need to provide sufficient computational and storage resources. The resource constraints of UR embedding are expressed as follows.

$$CPU(N_u^R) < CPU_r(N_m^S) \quad N_u^R \in N^R, N_m^S \in N^S, \quad (4)$$

where $CPU_r(N_m^S)$ denotes the current available computing resources of edge computing node N_m^S. Equation (4) is the computational resource constraint, the computational resources required by the UR cannot exceed the computational resources of the edge nodes, otherwise, the processing will fail.

$$M(N_u^R) < M_r(N_m^S) \quad N_u^R \in N^R, N_m^S \in N^S, \quad (5)$$

where $M_r(N_m^S)$ denotes the current available memory resources of edge computing node N_m^S. Equation (5) is the

memory resource constraint of the node, the amount of data of the UR cannot exceed the memory space of the edge node to ensure that the UR can be processed. In addition to the network resources, the delay performance of the nodes is also conditionally constrained. Equation (6) shows the delay constraints of the nodes.

$$\sum_{i=1}^{|N^S|} D(N^S) < D(N^R) \quad N_u^R \in N^R, N_m^S \in N^S. \tag{6}$$

To guarantee the delay, we set the delay attribute for the node. The total time consumed by the edge node for processing computational tasks is less than the maximum delay that is acceptable for user requests. If the node latency cannot meet the QoS requirements, the node cannot be used for processing user requests.

In addition to the nodes, the links similarly require meeting the corresponding bandwidth resource constraints.

$$BW(L_{(u,v)}^R) < BW_r(L_i^S) \quad L_{(u,v)}^R \in L^R, L_i^S \in L^S, \tag{7}$$

where $BW_r(L_i^S)$ denotes the current available bandwidth resources of the edge network link L_i^S. The edge network link L_i^S needs to allocate sufficient bandwidth resources for the user request link $L_{(u,v)}^S$.

E. EVALUATION INDICATORS

The goal of an ISP in providing services to its subscribers is to generate as much revenue as possible. We use Equation (8) to calculate the revenue earned by ISPs after processing user requests.

$$Re(G_k^R) = \sum_{i=1}^{|L^R|} BW(L_i^V) \cdot P(BW) + \sum_{i=1}^{|N^R|} [CPU(N_i^S) \cdot P(CPU) + M(N_i^S) \cdot P(M)]. \tag{8}$$

The above equation shows that the total revenue earned by the ISP is determined by the network resources consumed and the corresponding unit price of the resources.

Accordingly, the cost to the ISP is positively related to the resources allocated to the user request. The cost is calculated by the following equation.

$$Cost(G_k^R) = \sum_{N_i^r \in N^R} [CPU(N_i^r) + M(N_i^r)] + \sum_{L_i^r \in L^R} BW(L_i^r) \cdot hops(L_i^r), \ N^R \subset G_k^R, L^R \subset G_k^R, \tag{9}$$

where $hops(L_i^r)$ denotes the hop count of link L_i^r. The increase in the number of hops passed by the link will lead to the consumption of more bandwidth resources.

Increasing the success rate of user request embedding can bring more revenue. The user request success rate is calculated using the following equation.

$$SR = \lim_{T \to \infty} \frac{\sum_{t=0}^{T} num(G_{succ}^R)}{\sum_{t=0}^{T} num(G^R)}, \tag{10}$$

where G_{succ}^R denotes the successfully embedded UR.

The long-term average revenue-cost ratio indicates the merits of the network resource allocation scheme, and we calculate it using the following formula. The Equation (11) is derived from Equation (8) and (9).

$$LRC = \lim_{T \to \infty} \frac{\sum_{t=0}^{T} Re(G_k^R)}{\sum_{t=0}^{T} Cost(G_k^R)}. \tag{11}$$

$$Delay = \sum_{N_u \in N^R} \frac{CPU(N_m^S) - CPU_r(N_m^S)}{CPU_r(N_m^S)} \cdot D(N_m^S). \tag{12}$$

Equation (12) illustrates the delay to complete a UR, and the edge computing node N_m^S is used to host the UR node N_u. Node delay will be affected by CPU utilization, and $CPU_r(N_m^S)$ will change dynamically with the number of embedded nodes. Therefore, the allocation of the latest arriving URs will affect the already embedded URs.

IV. INTELLIGENT EDGE NETWORK RESOURCES ALLOCATION BASED ON REINFORCEMENT LEARNING

In this section, We will detail model of the intelligent edge network resources allocation algorithm based on reinforcement learning, which is based on a policy network. The agent extracts a feature matrix from the substrate network which we use it as the input of the policy network, and the output of the policy network is the probability that the node is embedded to the substrate network.

A. FEATURE MATRIX

Network nodes have many properties from which we wish to extract feature matrices as the training environment for the agent. Each node has attributes such as CPU, the total bandwidth of adjacent links, etc. To select higher-quality nodes, we need to extract the attributes of each edge node to construct a feature matrix.

We extract four attributes from the edge network:

(1) *CPU*: CPU resources for each substrate network node. Large CPU resources mean more user request nodes can be hosted.

(2) *BW*: The total bandwidth of all links connected to each node of the substrate network. Large bandwidth means more embedded links can be carried.

(3) *DL*: The sum of the delays of all links connected to each node of the substrate network. The larger the delay we have, the greater the probability of the embedded node.

(4) *DIS*: Average distance from each node to other nodes in the substrate network. A small average distance means that the distance between the node and other nodes is short, which will allow us to reduce our overhead when embedding nodes.

There are many features that can be extracted from nodes. The more feature attributes are extracted, the closer the training environment is to the real application scene. However, extracting too many features from the network increases computational complexity. Therefore, we only extract the above four important attributes. We need to normalize the feature matrix for better learning and faster convergence. After extracting the attributes of nodes, we normalize them and form the feature vector V_i:

$$V_i = (CPU_i, BW_i, DL_i, DIS_i) \tag{13}$$

We combine the feature vectors of all nodes to form the feature matrix M:

$$M = \begin{bmatrix} CPU_1 & BW_1 & DL_1 & DIS_1 \\ CPU_2 & BW_2 & DL_2 & DIS_2 \\ \cdots & \cdots & \cdots & \cdots \\ CPU_k & BW_k & DL_k & DIS_k \end{bmatrix}. \tag{14}$$

Each row in M represents the feature vector corresponding to a node. The feature matrix is used as input to the DRL agent and is updated as the edge computing network changes.

B. DEEP REINFORCEMENT LEARNING

We use a self-built artificial neural network called a policy network as a deep reinforcement learning agent. As shown in Figure 3, the policy network consists of an input layer, a convolutional layer, a softmax layer, and an output layer. The input to the policy network is a feature matrix and the output is the probability of the edge nodes being embedded. We train the policy network using historical user request data.

At the input layer, we compute the feature matrix and pass it to the policy network. Then, the policy network passes the feature matrix to the convolutional layer with one convolution kernel, where the policy network evaluates the resources of each substrate node:

$$O_k = \omega \cdot v_k + b, \tag{15}$$

where O_k is the k^{th} output of the convolutional layer, ω is the convolution kernel weight vector, v_k is the k^{th} input of the convolutional layer, and b is the bias.

Then the vector is passed to the softmax layer in order to produce the probability of each node. The softmax layer is only used in the last layer to normalize the values, so we have:

$$p_k = \frac{e^{O_k}}{\sum_{k=1}^{|N^S|} e^{O_k}}, \tag{16}$$

where p_k denotes the probability that the k^{th} node is successfully embedded, and γ_k^i is the resource contained in the k^{th} node. Finally, the output layer outputs the probability of each node being embedded.

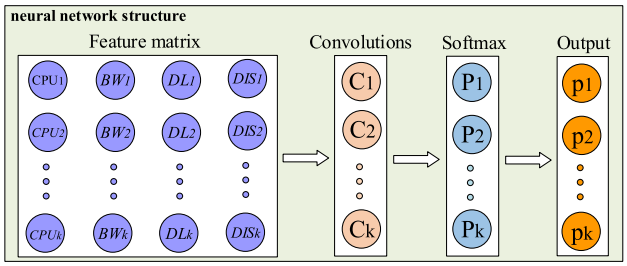

FIGURE 3. Composition of the neural network structure of the agent.

TABLE 1. Network Properties.

Parameter names	Values
Bandwidth resources of edge computing links	U[30, 100]
Computational resources of edge computing nodes	U[50, 100]
Delay of edge computing nodes	U[5, 50]
Memory resources of edge computing nodes	U[30, 100]
Number of edge computing nodes	80
Number of edge computing links	500
Bandwidth resources requirements of the requesting link	U[1, 30]
Computational resource requirements of the requesting node	U[1, 50]
Delay requirements of request node	U[10, 60]
Memory resource requirements of the requesting node	U[1, 30]
Number of URs	2000
Request nodes in each UR	U[2, 7]

V. PERFORMANCE EVALUATION

In this section, we have conducted some simulation experiments to prove the performance of the algorithm proposed in this paper, and compared it with other algorithms.

A. SIMULATION ENVIRONMENT AND PARAMETERS

We use Tensorflow [36] to build the policy network, and the computer used in the simulation is 64-bit win10 operating system. To evaluate our proposal, we used GT-ITM [37] to generate topologies of the physical network and the user request network. All experimental parameter settings are shown in Table 1. We generate physical network with 80 nodes and about 500 links. And the bandwidth of substrate network links are uniformly distributed between 30 and 100. Not only the substrate network but also the user request network were generated on a 250 × 250 grid. For the user request network, we generated 2000 URs with 2 to 7 nodes per UR. The bandwidth requirement of every requested link follows a uniform distribution between 1 and 50. We set the arrival of URs to follow a Poisson distribution with an average of 5 URs in 100 time units. The duration of each UR follows an exponential distribution with an average of 1000 time units. In addition, we divide UR into training set and test set on average. The training set is used for model training to aggregate the model. The test set is used to evaluate the performance of the trained model.

B. POLICY NETWORK TRAINING RESULTS

We transform the resource allocation problem of edge networks into a multi-domain virtual network mapping problem. Virtual network resource allocation includes node embedding

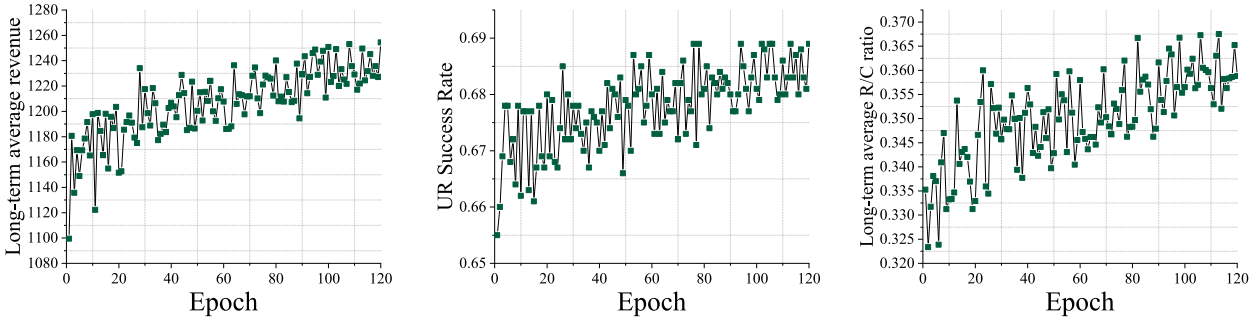

FIGURE 4. Training process.

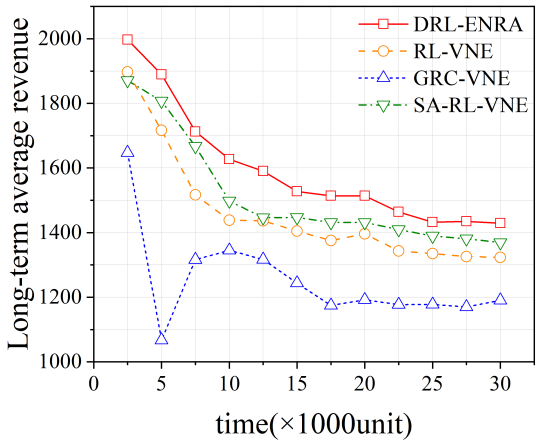

FIGURE 5. Long-term average revenue comparison.

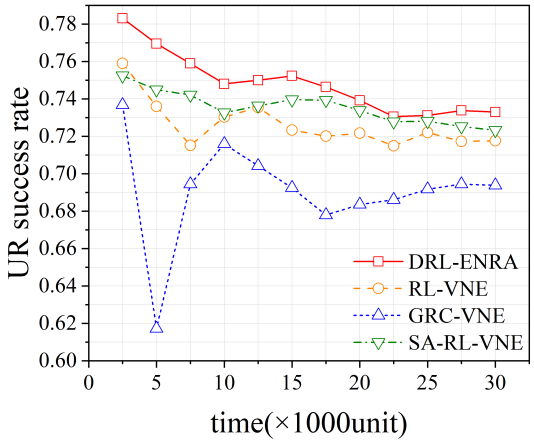

FIGURE 6. Percentage of completed URs.

and link embedding. We use machine learning to perform node embedding.

To improve the quality of the selected nodes, we train the DRL agent on a training network set containing 1000 URs. We performed 100 rounds of training on the training set and observed the performance in terms of long-term average revenue, UR success rate, and long-term revenue-cost ratio. Figure 4 shows the changes in the above three indicators during the 120 rounds of training. As can be seen from the figure, the values of the three indicators gradually increased with the number of training rounds and finally leveled off in an interval.

At the beginning of the training phase, since each parameter has just been initialized, the agent has no experience in executing UR embedding. Agent embeds UR randomly, so the performance of the three indicators is general and unstable. Then, the agent continuously adjusts the parameter weight according to the reward obtained after completing the UR embedding. When the agent receives a large reward, similar actions will be taken to accumulate more rewards after adopting the UR embedding strategy. Therefore, the performance of the algorithm is gradually improved. At the later stage of training, the algorithm converges gradually and the three evaluation indexes tend to be stable.

C. TEST RESULTS

We use another 1000 user requests as a test set for testing the performance of the trained algorithm. To validate the performance of our proposed DRL-based edge network resource allocation (DRL-ENRA) algorithm, we test both RL-VNE proposed in literature [38], the SA-RL-VNE algorithm proposed in the literature [39] and GRC-VNE proposed in literature [40] as comparison algorithms. Both RL-VNE and SA-RL-VNE are reinforcement learning-based network resource allocation algorithms that use gradient descent to achieve automatic optimization of resource allocation. GRC-VNE is a heuristic algorithm which makes embedding selection according to the global resource capacity of nodes to maximize the revenue of Infrastructure providers. Unlike the above two algorithms, we consider the delay factor in the process of UR and use DRL to optimize the algorithm to prevent the algorithm from falling into the local optimum.

As shown in Figure 5, we compare the performance of the four algorithms in terms of long-term average revenue. At the early stage of algorithm operation, the network resources are abundant, the conditions for processing URs are satisfied, and large revenues can be obtained. With the arrival of URs, the network resources are quickly occupied, some of the URs cannot be embedded, and the average revenue rapidly

decreases and remains at a certain level. The DRL-ENRA algorithm extracts multiple network resource attributes to form a training environment and continuously optimizes the resource allocation strategy by considering the revenue and delay factors, thus obtaining higher revenue. RL-VNE only considers the revenue-cost ratio when setting the agent's reward function and does not fully consider the allocation of multiple resources, resulting in a higher decrease in the long-term average revenue. The SA-RL-VNE algorithm mainly focuses on security with respect to network resource allocation, and therefore performs poorly when a delay constraint is introduced. GRC-VNE preferentially embeds user nodes to nodes with more global capacity resources, which leads to resource fragmentation and thus performs worse than the other two algorithms.

Figure 6 shows the comparison of four algorithms in UR success rate. Similar to the trend of long-term average revenue, the UR success rates of the four algorithms gradually decreased and stabilized at a certain level. The success rate of UR embedding is related to the number of network resources. The reason for the gradual decline in the success rate of UR is that UR continues to occupy edge network resources, and the available network resources continue to decrease, which cannot meet more UR needs. Our algorithm trains the cooperative allocation of multiple network resources and continuously optimizes the resource allocation strategy. Therefore, our algorithm has a high UR success rate.

Based on the analysis of the above experimental results, our algorithm has good performance in terms of revenue and UR success rate and can provide multi-objective edge network resource allocation with excellent performance.

VI. CONCLUSION

Edge computing makes it more convenient for smart grids to provide services to users. In this paper, we propose a DRL-based edge computing network-assisted smart grid network resource allocation algorithm to achieve multi-objective optimization based on user QoS requirements. We use the DRL approach to extract the feature attributes of the underlying network to construct a policy network to train the model, which can be more adaptable to the dynamically changing resource environment of the smart grid. The experimental results show that our proposed algorithm has excellent performance in terms of long-term average revenue and UR success rate.

In future work, we will extract more network features and build a more realistic training environment to enhance the capability of the DRL agent. Meanwhile, we will work on increasing the number of layers of the policy network to improve the learning ability of the DRL agent. In addition, we will investigate how to automatically predict and classify the QoS requirements of users.

REFERENCES

[1] D. Jiang, L. Huo, P. Zhang, and Z. Lv, "Energy-efficient heterogeneous networking for electric vehicles networks in smart future cities," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1868–1880, Mar. 2021, doi: 10.1109/TITS.2020.3029015.

[2] P. Zhang, C. Wang, G. S. Aujla, and R. S. Batth, "ReLeDP: Reinforcement-learning-assisted dynamic pricing for wireless smart grid," *IEEE Wireless Commun.*, vol. 28, no. 6, pp. 62–69, Dec. 2021, doi: 10.1109/MWC.011.2000431.

[3] P. Sharma and V. Jadhao, "Molecular dynamics simulations on cloud computing and machine learning platforms," in *Proc. IEEE 14th Int. Conf. Cloud Comput. (CLOUD)*, Sep. 2021, pp. 751–753, doi: 10.1109/CLOUD53861.2021.00101.

[4] Z. Jin, C. Zhang, Y. Jin, L. Zhang, and J. Su, "A resource allocation scheme for joint optimizing energy consumption and delay in collaborative edge computing-based industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 18, no. 9, pp. 6236–6243, Sep. 2022, doi: 10.1109/TII.2021.3125376.

[5] Y. Li and S. Wang, "An energy-aware edge server placement algorithm in mobile edge computing," in *Proc. IEEE Int. Conf. Edge Comput. (EDGE)*, Jul. 2018, pp. 66–73, doi: 10.1109/EDGE.2018.00016.

[6] S. Zhang, G. Cui, Y. Long, and W. Wang, "Joint computing and communication resource allocation for satellite communication networks with edge computing," *China Commun.*, vol. 18, no. 7, pp. 236–252, Jul. 2021, doi: 10.23919/JCC.2021.07.019.

[7] L. Liu, B. Sun, X. Tan, and D. H. K. Tsang, "Energy-efficient resource allocation and subchannel assignment for NOMA-enabled multiaccess edge computing," *IEEE Syst. J.*, vol. 16, no. 1, pp. 1558–1569, Mar. 2022, doi: 10.1109/JSYST.2021.3064919.

[8] L. V. Rueden, S. Mayer, K. Beckh, B. Georgiev, S. Giesselbach, R. Heese, B. Kirsch, M. Walczak, J. Pfrommer, A. Pick, R. Ramamurthy, J. Garcke, C. Bauckhage, and J. Schuecker, "Informed machine learning—A taxonomy and survey of integrating prior knowledge into learning systems," *IEEE Trans. Knowl. Data Eng.*, early access, May 12, 2021, doi: 10.1109/TKDE.2021.3079836.

[9] Y.-H. Wang, Y. Ou, X.-D. Deng, L.-R. Zhao, and C.-Y. Zhang, "The ship collision accidents based on logistic regression and big data," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Jun. 2019, pp. 4438–4440, doi: 10.1109/CCDC.2019.8832686.

[10] Y. Liu, S. Xie, and Y. Zhang, "Cooperative offloading and resource management for UAV-enabled mobile edge computing in power IoT system," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 12229–12239, Oct. 2020, doi: 10.1109/TVT.2020.3016840.

[11] Y. Chen, Z. Liu, Y. Zhang, Y. Wu, X. Chen, and L. Zhao, "Deep reinforcement learning-based dynamic resource management for mobile edge computing in industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 17, no. 7, pp. 4925–4934, Jul. 2021, doi: 10.1109/TII.2020.3028963.

[12] J. Wang, L. Zhao, J. Liu, and N. Kato, "Smart resource allocation for mobile edge computing: A deep reinforcement learning approach," *IEEE Trans. Emerg. Topics Comput.*, vol. 9, no. 3, pp. 1529–1541, Jul. 2021, doi: 10.1109/TETC.2019.2902661.

[13] Y. Liu, H. Yu, S. Xie, and Y. Zhang, "Deep reinforcement learning for offloading and resource allocation in vehicle edge computing and networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 11158–11168, Nov. 2019, doi: 10.1109/TVT.2019.2935450.

[14] L. Qian, Y. Wu, F. Jiang, N. Yu, W. Lu, and B. Lin, "NOMA assisted multi-task multi-access mobile edge computing via deep reinforcement learning for industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 17, no. 8, pp. 5688–5698, Aug. 2021, doi: 10.1109/TII.2020.3001355.

[15] K. Schwenk, S. Meisenbacher, B. Briegel, T. Harr, V. Hagenmeyer, and R. Mikut, "Integrating battery aging in the optimization for bidirectional charging of electric vehicles," *IEEE Trans. Smart Grid*, vol. 12, no. 6, pp. 5135–5145, Nov. 2021, doi: 10.1109/TSG.2021.3099206.

[16] J. Ma, "Research on the application of financial intelligence based on artificial intelligence technology," in *Proc. 2nd Int. Conf. Artif. Intell. Educ. (ICAIE)*, Jun. 2021, pp. 72–75, doi: 10.1109/ICAIE53562.2021.00022.

[17] C. Lee and A. Fumagalli, "Internet of Things security–multilayered method for end to end data communications over cellular networks," in *Proc. IEEE 5th World Forum Internet Things (WF-IoT)*, Apr. 2019, pp. 24–28, doi: 10.1109/WF-IoT.2019.8767227.

[18] D. Apostolopoulou, R. Poudineh, and A. Sen, "Distributed vehicle to grid integration over communication and physical networks with uncertainty effects," *IEEE Trans. Smart Grid*, vol. 13, no. 1, pp. 626–640, Jan. 2022, doi: 10.1109/TSG.2021.3119776.

[19] V. De Nitto Persone and V. Grassi, "Architectural issues for self-adaptive service migration management in mobile edge computing scenarios," in *Proc. IEEE Int. Conf. Edge Comput. (EDGE)*, Jul. 2019, pp. 27–29, doi: 10.1109/EDGE.2019.00020.

[20] P. Frauenthaler, M. Sigwart, C. Spanring, M. Sober, and S. Schulte, "ETH Relay: A cost-efficient relay for ethereum-based blockchains," in *Proc. IEEE Int. Conf. Blockchain (Blockchain)*, Nov. 2020, pp. 204–213, doi: 10.1109/Blockchain50366.2020.00032.
[21] Y. Song, S. S. Yau, R. Yu, X. Zhang, and G. Xue, "An approach to QoS-based task distribution in edge computing networks for IoT applications," in *Proc. IEEE Int. Conf. Edge Comput. (EDGE)*, Jun. 2017, pp. 32–39, doi: 10.1109/IEEE.EDGE.2017.50.
[22] J. X. Liao and X. W. Wu, "Resource allocation and task scheduling scheme in priority-based hierarchical edge computing system," in *Proc. 19th Int. Symp. Distrib. Comput. Appl. Bus. Eng. Sci. (DCABES)*, Oct. 2020, pp. 46–49, doi: 10.1109/DCABES50732.2020.00021.
[23] J. Xu, B. Palanisamy, H. Ludwig, and Q. Wang, "Zenith: Utility-aware resource allocation for edge computing," in *Proc. IEEE Int. Conf. Edge Comput. (EDGE)*, Jun. 2017, pp. 47–54, doi: 10.1109/IEEE.EDGE.2017.15.
[24] Z. Ali, S. Khaf, Z. H. Abbas, G. Abbas, F. Muhammad, and S. Kim, "A deep learning approach for mobility-aware and energy-efficient resource allocation in MEC," *IEEE Access*, vol. 8, pp. 179530–179546, 2020, doi: 10.1109/ACCESS.2020.3028240.
[25] D. Loghin, L. Ramapantulu, and Y. M. Teo, "Towards analyzing the performance of hybrid edge-cloud processing," in *Proc. IEEE Int. Conf. Edge Comput. (EDGE)*, Jul. 2019, pp. 87–94, doi: 10.1109/EDGE.2019.00029.
[26] W. Yi, W. Muqing, and H. Xiaolan, "An effective strategy of centralized multi-domain virtual network embedding," in *Proc. IEEE 5th Int. Conf. Comput. Commun. (ICCC)*, Dec. 2019, pp. 1186–1191, doi: 10.1109/ICCC47050.2019.9064364.
[27] Y. Ni, G. Huang, S. Wu, C. Li, P. Zhang, and H. Yao, "A PSO based multi-domain virtual network embedding approach," *China Commun.*, vol. 16, no. 4, pp. 105–119, Apr. 2019, doi: 10.12676/j.cc.2019.04.008.
[28] H.-G. Kim, D.-Y. Lee, S.-Y. Jeong, H. Choi, J.-H. Yoo, and J. W.-K. Hong, "Machine learning-based method for prediction of virtual network function resource demands," in *Proc. IEEE Conf. Netw. Softwarization (NetSoft)*, Jun. 2019, pp. 405–413, doi: 10.1109/NETSOFT.2019.8806687.
[29] A. Pahlevan, X. Qu, M. Zapater, and D. Atienza, "Integrating heuristic and machine-learning methods for efficient virtual machine allocation in data centers," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 8, pp. 1667–1680, Aug. 2018, doi: 10.1109/TCAD.2017.2760517.
[30] S. Wang, Y.-C. Wu, M. Xia, R. Wang, and H. V. Poor, "Machine intelligence at the edge with learning centric power allocation," *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7293–7308, Nov. 2020, doi: 10.1109/TWC.2020.3010522.
[31] F. Jiang, L. Zhang, C. Sun, and Z. Yuan, "Clustering and resource allocation strategy for D2D multicast networks with machine learning approaches," *China Commun.*, vol. 18, no. 1, pp. 196–211, Jan. 2021, doi: 10.23919/JCC.2021.01.017.
[32] B. Tan, Y. Peng, and J. Lin, "A local path planning method based on Q-learning," in *Proc. Int. Conf. Signal Process. Mach. Learn. (CONF-SPML)*, 2021, pp. 80–84, doi: 10.1109/CONF-SPML54095.2021.00024.
[33] G. O. Boateng, D. A.-Mensah, D. M. Doe, A. Mohammed, G. Sun, and G. Liu, "Blockchain-enabled resource trading and deep reinforcement learning-based autonomous RAN slicing in 5G," *IEEE Trans. Netw. Service Manage.*, vol. 19, no. 1, pp. 216–227, Mar. 2022, doi: 10.1109/TNSM.2021.3124046.
[34] G. O. Boateng, G. Sun, D. A. Mensah, D. M. Doe, R. Ou, and G. Liu, "Consortium blockchain-based spectrum trading for network slicing in 5G RAN: A multi-agent deep reinforcement learning approach," *IEEE Trans. Mobile Comput.*, early access, Jul. 19, 2022, doi: 10.1109/TMC.2022.3190449.
[35] H. Ding and K.-C. Leung, "Resource allocation for low-latency NOMA-V2X networks using reinforcement learning," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, May 2021, pp. 1–6, doi: 10.1109/INFOCOMWKSHPS51825.2021.9484529.
[36] D. Sierra-Sosa, M. Telahun, and A. Elmaghraby, "TensorFlow Quantum: Impacts of quantum state preparation on quantum machine learning performance," *IEEE Access*, vol. 8, pp. 215246–215255, 2020, doi: 10.1109/ACCESS.2020.3040798.
[37] M. O'Sullivan, L. Aniello, and V. Sassone, "A methodology to select topology generators for WANET simulations (extended version)," 2019, *arXiv:1908.09577*.
[38] H. Yao, X. Chen, M. Li, P. Zhang, and L. Wang, "A novel reinforcement learning algorithm for virtual network embedding," *Neurocomputing*, vol. 284, pp. 1–9, Apr. 2018.
[39] P. Zhang, C. Wang, C. Jiang, and A. Benslimane, "Security-aware virtual network embedding algorithm based on reinforcement learning," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 2, pp. 1095–1105, Apr. 2021, doi: 10.1109/TNSE.2020.2995863.
[40] L. Gong, Y. Wen, Z. Zhu, and T. Lee, "Toward profit-seeking virtual network embedding algorithm via global resource capacity," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2014, pp. 1–9.

YINGYING CHI received the master's degree in microelectronics from Tsinghua University, in 2013. She is currently the Deputy Manager with the Energy Efficiency Monitoring Division, Beijing Smartchip Microelectronics Technology Company Ltd. She has been engaged in integrated circuit technology innovation, product development, and application in the fields of artificial intelligence and communications. She has participated in the research work of more than ten national and government-level science and technology projects, and she has published two SCI-retrieved papers and 16 EI-retrieved papers, 11 of which she signed as the first author. In addition, she has obtained more than ten authorized patents.

YI ZHANG is currently a Graduate Student with the College of Computer Science and Technology, China University of Petroleum (East China). His research interests include network virtualization and artificial intelligence for networking.

YONG LIU is currently a Technical Expert with Beijing Smartchip Microelectronics Technology Company Ltd. He has obtained the title of Senior Engineer and is mainly engaged in research on the application of new technologies such as TSN and AI in power systems. He has participated in the compilation of five national standards and three industry standards.

HAILONG ZHU received the B.S. degree in measurement and control technology and instrumentation from Northwestern Polytechnical University, in 2009, and the Ph.D. degree in control science and engineering from Tsinghua University, in 2019. He is currently a Lecture with the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications. His research interests include the network and communication technologies in automated driving, TSN, C-V2X, and industrial control networks and systems.

ZHE ZHENG received the Ph.D. degree from the University of Southern California, USA. He is currently the Deputy General Manager with the Energy Efficiency Monitoring Division, Beijing Smartchip Microelectronics Technology Company Ltd., and he is also a Senior Engineer. He has published more than 30 papers and obtained more than 20 authorized patents. He has been deeply involved in electric power industry for more than ten years, and is good at industrial chip technology research and product application. He has led the development of a number of chip products, including the TSN switch chip and the low-power artificial intelligence chip. He has published more than 30 papers and obtained more than 20 authorized patents. He has served as a member for the Artificial Intelligence Standardization Technical Committee of the China Electricity Council, the National Information Technology Standardization Technical Committee, and the IEEE Electric Power and Energy Association.

RUI LIU is currently the Assistant General Manager with the Energy Efficiency Monitoring Division, Beijing Smartchip Microelectronics Technology Company Ltd. He has obtained the title of Senior Engineer and is mainly engaged in artificial intelligence technology, battery sampling technology, and related system research. He has participated in one national key research and development program project and five state grid science and technology projects. He is mainly engaged in artificial intelligence technology, battery sampling technology, and related system research. He has participated in one national key research and development program project and five state grid science and technology projects.

PEIYING ZHANG received the Ph.D. degree from the School of Information and Communication Engineering, University of Beijing University of Posts and Telecommunications, in 2019. He is currently an Associate Professor with the College of Computer Science and Technology, China University of Petroleum (East China). He has published multiple IEEE/ACM TRANSACTIONS/ journal/magazine articles since 2016, such as the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING, *IEEE Network*, IEEE INTERNET OF THINGS JOURNAL, *ACM TALLIP*, *Computer Communications*, and *IEEE Communications Magazine*. His research interests include semantic computing, future internet architecture, network virtualization, and artificial intelligence for networking. He served as the Technical Program Committee Member for IEEE ICC'23, IEEE ICC'22, DPPR 2021, ISCIT 2016, ISCIT 2017, ISCIT 2018, ISCIT 2019, GLOBECOM 2022, GLOBECOM 2021, GLOBECOM 2019, COMNETSAT 2020, ICICoS 2022, SoftIoT 2021, IWCMC-Satellite 2019, IWCMC-Satellite 2020, and IWCMC-Satellite 2022. He is the Leading Guest Editor of *Electronics* and *Frontiers in Psychiatry*, and an Editorial Board Member of *Artificial Intelligence and Applications* (AIA).