Automated Software Engineering (2024) 31:66
https://doi.org/10.1007/s10515-024-00463-8

Energy efficient resource allocation based on virtual network embedding for IoT data generation

Lizhuang Tan[1,2] · Amjad Aldweesh[3] · Ning Chen[4] · Jian Wang[5] · Jianyong Zhang[6] · Yi Zhang[9] · Konstantin Igorevich Kostromitin[8] · Peiying Zhang[1,7]

Received: 10 May 2024 / Accepted: 3 August 2024
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2024

Abstract

The Internet of Things (IoT) has become a core driver leading technological advancements and social transformations. Furthermore, data generation plays multiple roles in IoT, such as driving decision-making, achieving intelligence, promoting innovation, improving user experience, and ensuring security, making it a critical factor in promoting the development and application of IoT. Due to the vast scale of the network and the complexity of device interconnection, effective resource allocation has become crucial. Leveraging the flexibility of Network Virtualization technology in decoupling network functions and resources, this work proposes a Multi-Domain Virtual Network Embedding algorithm based on Deep Reinforcement Learning to provide energy-efficient resource allocation decision-making for IoT data generation. Specifically, we deploy a four-layer structured agent to calculate candidate IoT nodes and links that meet data generation requirements. Moreover, the agent is guided by the reward mechanism and gradient back-propagation algorithm for optimization. Finally, the effectiveness of the proposed method is validated through simulation experiments. Compared with other methods, our method improves the long-term revenue, long-term resource utilization, and allocation success rate by 15.78%, 15.56%, and 6.78%, respectively.

Keywords Internet of Things · Data generation · Network virtualization · Virtual network embedding · Deep reinforcement learning

1 Introduction

In the contemporary epoch characterized by digitalization, the Internet of Things (IoT) emerges as a pivotal catalyst propelling technological progress and societal metamorphosis (Wu et al. 2024a; Chettri and Bera 2020; Wu et al. 2024b). IoT facilitates

Extended author information available on the last page of the article

Published online: 12 August 2024

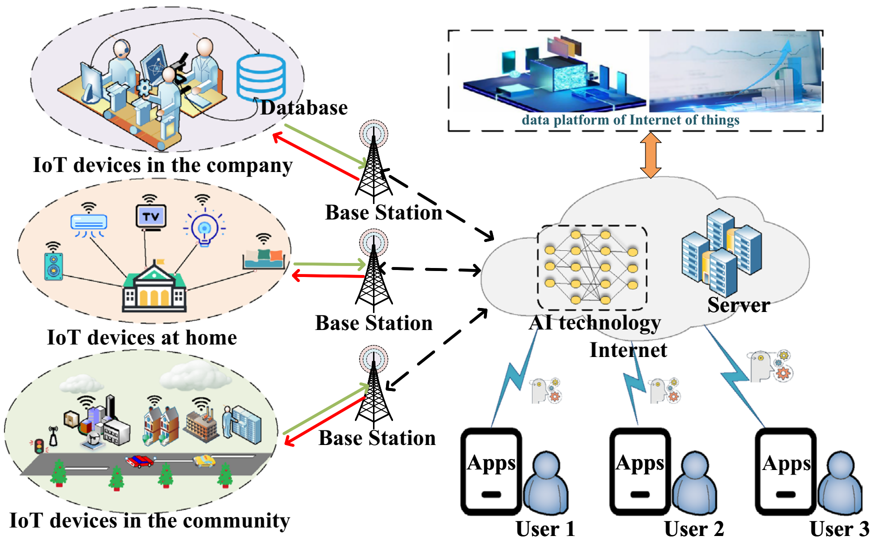

Fig. 1 IoT data generation is used to support various fields

instantaneous communication and data exchange through the interconnection of diverse devices, sensors, and systems. The implementation of IoT engenders a more intelligent, efficient, and sustainable societal lifestyle (Zhang et al. 2023a). Through interaction with intelligent devices, substantial enhancements in home automation, smart city governance, and industrial production optimization can be realized (Du et al. 2023). Nevertheless, the essence of IoT lies in the generation and utilization of data (Ahmed et al. 2017). Data generation relies on a specific data model, wherein raw data undergoes computations to yield the requisite data aligned with the model for the target system (Dash et al. 2020). Continuously, as shown in Fig. 1, connected devices and sensors produce copious amounts of data, constituting a valuable reservoir for insights into business and social operations. Through the analysis of this data, comprehension of consumer behavior, optimization of production processes, enhancement of resource efficiency, and myriad other outcomes can be achieved (Hsu et al. 2021). The significance of data generation lies in furnishing a real-time, factually grounded foundation for decision-making, empowering us to tackle challenges with greater promptness and precision (Duan et al. 2022a).

Within a myriad of IoT scenarios, the tightly woven interconnectivity among devices and sensors renders data generation an increasingly pervasive phenomenon. This expansive and diverse flow of data encompasses various stages, ranging from data collection through processing to subsequent transmission. In this intricate ecosystem, the imperative of efficient resource allocation becomes apparent (Zhang et al. 2024a). Owing to the interconnected nature of sensors and devices, they not only generate copious amounts of data but also present formidable challenges to the efficiency of networks and the utilization of resources (Al-Hadhrami and Hussain 2020). To harness the potential of the Internet of Things, it is imperative to acknowledge the significance of data generation in IoT, with a pivotal challenge being the attainment of efficient resource allocation within the expansive data flow (Zhang et al. 2023b; Duan et al. 2022b). In other scenarios, such as traditional data centers or cloud computing environments, although data generation and processing tasks also exist, the challenge in IoT scenarios is how to efficiently allocate and manage these tasks in resource-constrained environments. Maximizing the reduction of transmitted data not only enhances network efficiency but also mitigates transmission latency by conserving precious resources. Efficient resource allocation not only renders IoT systems more agile and responsive but also establishes a robust foundation for innovation across diverse industries (Feng et al. 2024). Therefore, directing attention to and optimizing resource allocation is pivotal for maximizing the benefits derived from IoT data generation.

Network Virtualization (NV) technology represents an innovative approach to the management of computer networks (Chen et al. 2023). By abstracting and isolating diverse network resources, including bandwidth, routers, and switches, it establishes multiple logically independent and mutually isolated virtual networks (Zhang et al. 2023c). Due to its flexibility and efficiency, this technology finds extensive applications in fields such as cloud computing and data centers (Xiao et al. 2023). NV enables enterprises to more effectively utilize network resources, enhance network performance and flexibility, and reduce operational costs, thereby propelling the process of digital transformation (Chen et al. 2023). In sectors such as cloud computing and data centers, the application of network virtualization endows systems with scalability and elasticity, enabling dynamic allocation and management of resources based on demand (Zhang et al. 2024b). In this context, Virtual Network Embedding (VNE), as an integral component of network virtualization, assumes the role of connecting virtual networks to the underlying physical network (Chen et al. 2022; Wu et al. 2024c). VNE effectively maps virtual networks onto physical networks, ensuring optimal utilization of resources and maximizing performance. In the realm of the Internet of Things, the application of VNE becomes particularly crucial as it offers an innovative approach to addressing resource allocation challenges in IoT data generation. Through VNE, efficient resource allocation can be achieved in the IoT environment, ensuring the stability and reliability of the network during the process of data generation. This technology not only enhances the efficiency of data transmission but also furnishes IoT systems with a more flexible and intelligent network architecture. Therefore, the significance of VNE in the context of NV cannot be overstated, as it provides robust support for the successful implementation of IoT data generation (Wu et al. 2024b).

Based on the above motivations, in this work, we propose a multi-domain virtual network embedding (MD-VNE) algorithm based on deep reinforcement learning (DRL) to provide energy efficient resource allocation decision-making for IoT data generation. Specifically, the contributions are as follows,

- This is one of the first attempts to leverage VNE technology to provide energy-efficient resource allocation decision-making for IoT data generation. In addition, we model IoT as a multi-domain physical network and construct resource constraints with storage, computing, and bandwidth as research objects.

- We propose a four-layer structured agent for calculating candidate IoT nodes and links that meet data generation demands. Based on the reward mechanism and gradient back-propagation algorithm, the agent is guided to optimize to improve the long-term revenue, long-term resource utilization, and allocation success rate of resource allocation.
- We conducted a large number of simulation experiments to fully verify the effectiveness and superiority of the proposed method in the training and testing stages.

2 Related work

2.1 Data generation

Data generation has evolved through the following main stages:

2.1.1 Sensor data simulation

Sensor data simulation is a common method for IoT data generation. By simulating the operating principles and environmental conditions of sensors, data streams similar to actual sensor data can be generated. For example, by simulating the readings of a temperature sensor, a series of temperature data can be generated for testing and evaluating the performance of an IoT application. Congress and Puppala (2022) applied UAVs to transportation infrastructure health monitoring and feature recognition techniques based on machine learning to identify traffic base information, which helped to optimize the use of UAVs for bridge health monitoring.

2.1.2 Model-based generation

Model-based generation is an approach to generate IoT data by training machine learning models. These models can learn and simulate the distribution and characteristics of IoT data to generate synthetic data that is similar to real data. For example, Generative Adversarial Networks (GANs) can be used to generate realistic IoT data. For the accurate detection and identification of malicious domain names, Woodbridge et al. (2016) for the first time used long and short-term memory networks to construct a DGA (domain generation algoritim) city name detector, and in terms of detection accuracy, recall rate and other aspects compared to the traditional machine learning models have obvious advantages.

2.1.3 Data synthesis

Data synthesis is another commonly used method for IoT data generation. It is based on existing real data and generates new datasets by processing and transforming the data. This method can help in expanding the size of existing datasets and can generate data with different features and attributes. Yao et al.

(2020) proposed Attribute Descent to optimize the attributes of VehicleX for the field of vehicle re-recognition, which is able to generate a synthetic dataset that is better than random attributes, and can effectively improve the training accuracy of real data through joint training.

2.2 Resource allocation

Research and development of IoT applications often necessitate large datasets for training and evaluating models. IoT devices and sensors can collect and transmit data in real-time across various environments and devices. This real-time data collection enables decision-makers to promptly gain insights into resource usage and demand, facilitating timely resource allocation decisions (Duan et al. 2022c). Through the analysis and extraction of these data, we can dynamically allocate and adjust resources based on actual needs and circumstances. This dynamic adaptability ensures that resource allocation aligns with specific requirements, thus enhancing resource utilization efficiency and performance. By establishing a resource allocation model that spans devices and applications, we can better coordinate and optimize resource allocation, thereby improving overall system efficiency and performance.

VNE problems have been proven to be NP difficult problems, and the strategies for solving VNE problems can be roughly divided into two categories, heuristic algorithms and artificial intelligence (AI)-based algorithms.

2.2.1 Heuristic-based VNE algorithm

Heuristics usually divide the solution process into two separate stages, namely node embedding and link embedding (Zhan et al. 2022). In the node embedding phase, the resource properties and topology properties of the node are usually used to embed the virtual node on the physical node. In Ref. (Chen et al. 2021), Chen et al. proposed an algorithm based on the concept of distribution equilibrium for virtual node embedding on flexible mesh optical networks. The simulation results show that there are advantages in terms of acceptance rate and system benefits. Sun et al. (2019) investigated the VNF placement problem for energy-efficient and flow-aware SFC orchestration across multiple clouds. They formulate the problem as an ILP model, and then propose a low-complexity heuristic called Energy-Efficient Online SFC Request Orchestration (EE-SFCO-MD) across multiple domains, which can generate a near-optimal solution to the above problem. However, heuristic algorithms usually adopt a local search strategy, focusing only on the neighborhood of the current optimal solution in the search process. This local search strategy may lead to the algorithm falling into the local optimal solution, unable to jump out of the local optimal solution and find a better global solution. Therefore, the performance of the heuristic algorithm is limited by the choice of search strategy and the limitation of search space.

2.2.2 AI-based VNE algorithm

In recent years, reinforcement learning algorithms have been gradually applied in various fields due to their inherent advantages in solving decision-making problems. Since VNE problems can be naturally modeled as a decision-making process, mapping a VNF at each decision-making step, RL has been applied to the design of VNE schemes in many studies (Zhan et al. 2022; Zhang et al. 2022; Wang et al. 2023). However, many of these RL methods train the manual functions of the physical and virtual networks as input to the RL agent. This doesn't give you a complete picture of the state of your VNE environment, as a separate list of resource characteristics for nodes can lose structural information contained in the network itself.

Deep reinforcement learning (DRL) can be learned directly from the original topology information of virtual and physical networks, without the need to manually design features or rules. Through end-to-end learning, deep reinforcement learning can more comprehensively grasp the state of the VNE environment, including the structure information of the network. Policies and decisions can also be dynamically adjusted based on feedback from the environment, adapting to the changing VNE environment. Liu and Zhang (2024) proposed an advanced service function chain (SFC) embedding algorithm based on an enhanced deep deterministic policy gradient (E-DDPG) model, and achieved remarkable results in highly dynamic and complex cloud networks. Zhao et al. (2022) proposed an advanced cross-domain SFC embedding algorithm based on a collaborative multi-agent reinforcement learning algorithm and achieved two- and four-fold improvements in running time and resource cost. Wu et al. (2024b) proposed a cross-domain VNE algorithm based on DRL for task offloading in Industrial IoT (IIoT) and achieved good results. In order to solve the above problems, this paper proposes a VNE scheme based on deep reinforcement learning for data generation in a general IoT scenario.

3 Problem definition

Based on cluster technology, IoT generally consists of multiple regional networks, each of which is responsible for providing specific functions or services (Wu et al. 2024b). Therefore, this work aims to explore multi-domain VNE (MD-VNE) for data generation in IoT.

3.1 Abstract modeling

First, the abstract modeling of networks is required, where the relevant notations used in this work are shown in Table 1. For clarity, other notations defined in this work are shown in Table 2.

A diagram of this problem is shown in Fig. 2, this work focuses on two types of entities, one is the IoT network and the other is the data generation requests (DGRs)

Table 1 Notations used in this work

Modeling	Attribute	Notation
IoT	Nodes of IoT	N^{I}
	Links of IoT	L^{I}
	Location of IoT nodes	D^{I}
	Storage of IoT nodes	S^{I}
	Computing of IoT nodes	C^{I}
	Bandwidth of IoT links	B^{I}
DGR	Virtual nodes for DGRs	N^{V}
	Virtual links for DGRs	L^{V}
	Virtual storage for DGR nodes	S^{V}
	Virtual computing for DGRs nodes	C^{V}
	Virtual bandwidth for DGRs links	B^{V}

Table 2 Other notations defined in this work

Notation	Definition
$\alpha_{n^{\mathrm{V}}}^{n^{\mathrm{I}}}$	Whether the IoT node resources are successfully allocated
$\beta_{l^{\mathrm{V}}}^{l^{\mathrm{I}}}$	Whether the IoT link resources are successfully allocated
$R\_S^{\mathrm{I}}(n_i^{\mathrm{I}})$	The remaining storage resources of the IoT node
$R\_S^{\mathrm{I}}(n_i^{\mathrm{I}})$	The remaining computing resources of the IoT node
$R\_B^{\mathrm{I}}(l_i^{\mathrm{I}})$	The remaining bandwidth resources of the IoT link
$\mathcal{C}(G_i^{\mathrm{V}})$	The embedded cost of i-th DGR
$\mathcal{R}(G_i^{\mathrm{V}})$	The embedded revenue of i-th DGR
$\mathcal{LR}$	The long-term revenue
$\mathcal{LRU}$	The long-term resource utilization
$\mathcal{ASR}$	The allocation success rate
$TR\_B^{\mathrm{I}}(n_i^{\mathrm{I}})$	The total remaining bandwidth of the links connected to the current IoT node
$Dis^{\mathrm{I}}(n_i^{\mathrm{I}})$	The distance to other IoT nodes

in the IoT network. Moreover, they are all abstractly modeled as weighted undirected graphs. Specifically, the IoT is modeled as,

$$G^{\mathrm{I}} = \left\{N^{\mathrm{I}}, L^{\mathrm{I}}, D^{\mathrm{I}}(N^{\mathrm{I}}), S^{\mathrm{I}}(N^{\mathrm{I}}), C^{\mathrm{I}}(N^{\mathrm{I}}), B^{\mathrm{I}}(L^{\mathrm{I}})\right\}. \tag{1}$$

In addition, the DGR is modeled as,

$$G^{\mathrm{V}} = \left\{N^{\mathrm{V}}, L^{\mathrm{V}}, S^{\mathrm{V}}(N^{\mathrm{V}}), C^{\mathrm{V}}(N^{\mathrm{V}}), B^{\mathrm{V}}(L^{\mathrm{V}})\right\}, \tag{2}$$

Furthermore, we use lowercase letters to describe specific elements. For example, n^{I} represents a specific IoT node, l_{ab}^{I} represents the IoT link between n_a^{I} and n_b^{I}, etc.

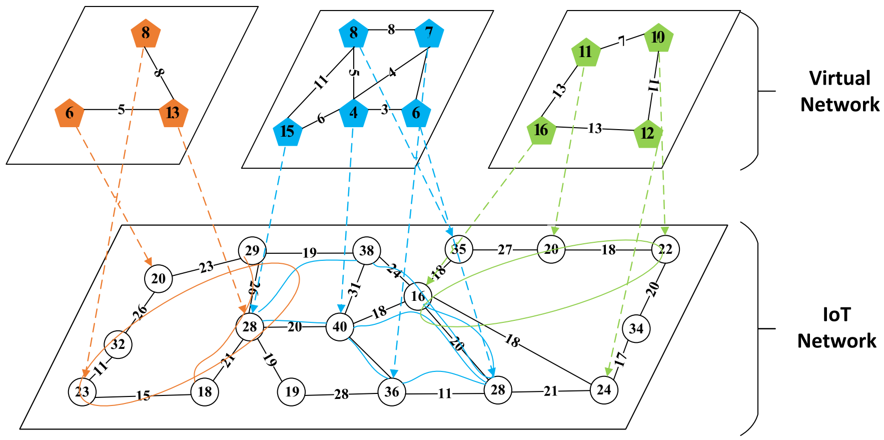

Fig. 2 A diagram of abstract modeling in this problem

3.2 Problem modeling

Based on the above analysis, the problem of this work can be summarized as: according to the resource requests of DGRs, allocate corresponding IoT resources to them for data generation tasks. Therefore, it can be characterized as a resource allocation process, as follows,

$$G^{\mathrm{I}} \rightarrow G_i^{\mathrm{V}},\ \text{for}\ \forall\ 1 \le i \le |\mathrm{DGRs}|, t_{s,i} \le t \le t_{e,i}, \tag{3}$$

where |DGRs| represents the number of DGRs; $t_{s,i}$ and $t_{e,i}$ represent the start time and end time of the i-th DGR, respectively. It can be found that this problem is a process of dynamically allocating resources for multiple groups of DGRs. Moreover, it is an obviously NP-hard problem. Therefore, the optimization goal of this work is to satisfy as many DGRs as possible with the least resource cost, i.e., energy-efficient data generation for IoT.

3.3 Problem constraint

$\alpha_{n^{\mathrm{V}}}^{n^{\mathrm{I}}}$ and $\beta_{l^{\mathrm{V}}}^{l^{\mathrm{I}}}$ respectively indicate whether the IoT node and link resources are successfully allocated. If the value is 1, it means success, otherwise it means failure, as follows,

$$\alpha_{n^{\mathrm{V}}}^{n^{\mathrm{I}}} = \begin{cases} 1, & \text{node allocated successfully}, \\ 0, & \text{others}, \end{cases} \tag{4}$$

$$\beta_{l^{\mathrm{V}}}^{l^{\mathrm{I}}} = \begin{cases} 1, & \text{link allocated successfully}, \\ 0, & \text{others}. \end{cases} \tag{5}$$

Accordingly, the relevant constraints that need to be met during the resource allocation process of IoT data generation are as follows,

- One virtual node can only be hosted on one IoT node, as follows,

$$\sum_{i=1}^{|N^{\mathrm{V}}|} \alpha_{n_i^{\mathrm{V}}}^{n_j^{\mathrm{I}}} = 1, \text{ for } n_j^{\mathrm{I}} \in N^{\mathrm{I}}, \tag{6}$$

  where $|N^{\mathrm{V}}|$ represents the number of virtual nodes for a DGR.
- One virtual links can be hosted on multiple IoT links, i.e., deployed across links, as follows,

$$\sum_{i=1}^{|L^{\mathrm{V}}|} \beta_{l_i^{\mathrm{V}}}^{l_k^{\mathrm{I}}} \geq 1, \text{ for } l_k^{\mathrm{I}} \in L^{\mathrm{I}}, \tag{7}$$

  where $|L^{\mathrm{V}}|$ represents the number of virtual links for a DGR.
- The remaining storage resources $R\_S^{\mathrm{I}}(n_i^{\mathrm{I}})$ of the IoT node cannot be negative, as follows,

$$R\_S^{\mathrm{I}}(n_i^{\mathrm{I}}) = S^{\mathrm{I}}(n_i^{\mathrm{I}}) - \sum_{j=1}^{|N^{\mathrm{V}}|} \alpha_{n_j^{\mathrm{V}}}^{n_i^{\mathrm{I}}} \cdot S^{\mathrm{V}}(n_j^{\mathrm{V}}), \tag{8}$$

$$R\_S^{\mathrm{I}}(n_i^{\mathrm{I}}) \geq 0, \text{ for } n_i^{\mathrm{I}} \in N^{\mathrm{I}}. \tag{9}$$

- The remaining computing resources $R\_S^{\mathrm{I}}(n_i^{\mathrm{I}})$ of the IoT node cannot be negative, as follows,

$$R\_C^{\mathrm{I}}(n_i^{\mathrm{I}}) = C^{\mathrm{I}}(n_i^{\mathrm{I}}) - \sum_{j=1}^{|N^{\mathrm{V}}|} \alpha_{n_j^{\mathrm{V}}}^{n_i^{\mathrm{I}}} \cdot C^{\mathrm{V}}(n_j^{\mathrm{V}}), \tag{10}$$

$$R\_C^{\mathrm{I}}(n_i^{\mathrm{I}}) \geq 0, \text{ for } n_i^{\mathrm{I}} \in N^{\mathrm{I}}. \tag{11}$$

- The remaining bandwidth resources $R\_B^{\mathrm{I}}(l_i^{\mathrm{I}})$ of the IoT link cannot be negative, as follows,

$$R\_B^{\mathrm{I}}(l_i^{\mathrm{I}}) = B^{\mathrm{I}}(l_i^{\mathrm{I}}) - \sum_{j=1}^{|L^{\mathrm{V}}|} \beta_{l_j^{\mathrm{V}}}^{l_i^{\mathrm{I}}} \cdot B^{\mathrm{V}}(l_j^{\mathrm{V}}), \tag{12}$$

$$R\_B^{\mathrm{I}}(l_i^{\mathrm{I}}) \geq 0, \text{ for } l_i^{\mathrm{I}} \in L^{\mathrm{I}}. \tag{13}$$

- If the resources of an IoT node are to be allocated to a virtual node, its available resources should be sufficient for allocation, as follows,

$$\text{if } \alpha_{n_j^{\mathrm{V}}}^{n_i^{\mathrm{I}}} = 1, \text{ then } R\_S^{\mathrm{I}}(n_i^{\mathrm{I}}) \geq S^{\mathrm{V}}(n_j^{\mathrm{V}}), R\_C^{\mathrm{I}}(n_i^{\mathrm{I}}) \geq C^{\mathrm{V}}(n_j^{\mathrm{V}}). \tag{14}$$

- If the resources of an IoT node are to be allocated to a virtual node, its available resources should be sufficient for allocation, as follows,

$$\text{if } \beta_{l_j^{\mathrm{V}}}^{l_i^{\mathrm{I}}} = 1, \text{ then } R\_B^{\mathrm{I}}(l_i^{\mathrm{I}}) \geq B^{\mathrm{V}}(l_j^{\mathrm{V}}). \tag{15}$$

3.4 Optimization goal

The goal of this work is to provide energy-efficient resource allocation strategies for IoT data generation. In other words, obtain the maximum allocation revenue with the lowest energy cost. When resources are successfully allocated to DGRs, energy revenue and cost will generate accordingly. Specifically, for i-th DGR, the corresponding revenue and cost are defined as follows,

$$\mathcal{C}\left(G_i^{\mathrm{V}}\right) = \sum_{j=1}^{|N^{\mathrm{V}_i}|} \left(S^{\mathrm{V}}\left(n_j^{\mathrm{V}_i}\right) + C^{\mathrm{V}}\left(n_j^{\mathrm{V}_i}\right)\right) + \sum_{k=1}^{|L^{\mathrm{V}_i}|} h\left(l_k^{\mathrm{V}_i}\right) \cdot B^{\mathrm{V}}\left(l_k^{\mathrm{V}_i}\right), \tag{16}$$

$$\mathcal{R}\left(G_i^{\mathrm{V}}\right) = \sum_{j=1}^{|N^{\mathrm{V}_i}|} \left(S^{\mathrm{V}}\left(n_j^{\mathrm{V}_i}\right) + C^{\mathrm{V}}\left(n_j^{\mathrm{V}_i}\right)\right) + \sum_{k=1}^{|L^{\mathrm{V}_i}|} B^{\mathrm{V}}\left(l_k^{\mathrm{V}_i}\right), \tag{17}$$

where $h\left(l_k^{\mathrm{V}_i}\right)$ represents the hop count of the IoT links hosting $l_k^{\mathrm{V}_i}$. According to the one-to-many relationship between the virtual link and the IoT link shown in Eq. 7, it can be observed that the larger the number of hops, the higher the energy cost.

Correspondingly, the optimization goals of this work are long-term revenue, long-term resource utilization, and allocation success rate. Their calculation methods are as follows,

$$\mathcal{LR} = \lim_{T \to \infty} \frac{\sum_{i=1}^{|\mathrm{DGRs}|} \sum_{t=0}^{T} \mathcal{R}(G_i^{\mathrm{V}})}{T}, \tag{18}$$

$$\mathcal{LRU} = \lim_{T \to \infty} \frac{\sum_{i=1}^{|\mathrm{DGRs}|} \sum_{t=0}^{T} \mathcal{R}(G_i^{\mathrm{V}})}{\sum_{i=1}^{|\mathrm{DGRs}|} \sum_{t=0}^{T} \mathcal{C}(G_i^{\mathrm{V}})}, \tag{19}$$

$$\mathcal{ASR} = \frac{\sum_{i=1}^{|\mathrm{DGRs}|} \left(\prod_{j=1}^{|N^{\mathrm{V}_i}|} \alpha_{n_j^{\mathrm{V}}}^{n^{\mathrm{I}}} \cdot \prod_{k=1}^{|L^{\mathrm{V}_i}|} \beta_{l_j^{\mathrm{V}}}^{l^{\mathrm{I}}}\right)}{|\mathrm{DGRs}|}, \tag{20}$$

where as Eq. 18, the long-term revenue is expressed as the cumulative sum of revenue over the average time. As Eq. 19, the long-term resource utilization is expressed as the ratio of long-term cumulative revenue to long-term cumulative cost. As Eq. 20, the allocation success rate is expressed as the ratio of the number of successfully allocated DGRs to the total number.

4 The proposed energy efficient resource allocation method

DGRs arrive at IoT sequentially in chronological order, and IoT allocates corresponding resources based on the demands of each DGR. Upon the completion of the DGR lifecycle, the resources it occupied are released. Thus, the resource allocation process for IoT data generation is a sequential decision-making process. Motivated by this, we deploy a DRL model to learn and optimize the decision-making process.

4.1 Model design

First, the DRL model needs to model the Markov Decision Process (MDP), which includes the following elements,

4.1.1 State

It refers to the specific situation or configuration that an agent finds itself during interaction with the environment. In DRL, an agent learns how to perform a task or achieve a certain goal by interacting with the environment. The state of the environment describes the current environmental conditions in which the agent operates, including various variables, features, or attributes present in the environment. The agent perceives its current environmental information by observing the state of the environment, and makes actions based on this information. In this work, the following IoT environment information is extracted as the state,

- The remaining storage resources of IoT nodes: $R\_S^{\mathrm{I}}(n_i^{\mathrm{I}})$. The larger the value, the more indicative it is that the IoT node is more likely to be used for resource provisioning.
- The remaining computing resources of IoT nodes: $R\_C^{\mathrm{I}}(n_i^{\mathrm{I}})$. The larger the value, the more indicative it is that the IoT node is more likely to be used for resource provisioning.
- The total remaining bandwidth of the links connected to the current IoT node: $TR\_B^{\mathrm{I}}(n_i^{\mathrm{I}})$. The larger the value, the more indicative it is that the IoT node is more likely to be used for resource provisioning.

$$TR\_B^{\mathrm{I}}(n_i^{\mathrm{I}}) = \sum_{\forall l_{ij}^{\mathrm{I}}} R\_B^{\mathrm{I}}(l_{ij}^{\mathrm{I}}), \tag{21}$$

where l_{ij}^{I} represents the IoT links connected to n_i^{I}.

- The distance to other IoT nodes: $Dis^{\mathrm{I}}(n_i^{\mathrm{I}})$. The smaller its value, the less link resource loss the node may cause, and accordingly the more likely to be used for resource provisioning.

$$Dis^{\mathrm{I}}(n_i^{\mathrm{I}}) = \frac{\sum_{\forall l_{ij}^{\mathrm{I}}} \|D^{\mathrm{I}}(n_i^{\mathrm{I}}) - D^{\mathrm{I}}(n_j^{\mathrm{I}})\|^2}{1 + h(l_{ij}^{\mathrm{I}})}. \tag{22}$$

Combined with the above environmental information, the following matrix can be constructed as the state input of the agent,

$$S = \begin{bmatrix} R\_S^{\mathrm{I}}(n_1^{\mathrm{I}}) & R\_C^{\mathrm{I}}(n_1^{\mathrm{I}}) & TR\_B^{\mathrm{I}}(n_1^{\mathrm{I}}) & Dis^{\mathrm{I}}(n_1^{\mathrm{I}}) \\ R\_S^{\mathrm{I}}(n_2^{\mathrm{I}}) & R\_C^{\mathrm{I}}(n_2^{\mathrm{I}}) & TR\_B^{\mathrm{I}}(n_2^{\mathrm{I}}) & Dis^{\mathrm{I}}(n_2^{\mathrm{I}}) \\ \vdots & \vdots & \vdots & \vdots \\ R\_S^{\mathrm{I}}(n_{|N^{\mathrm{I}}|}^{\mathrm{I}}) & R\_C^{\mathrm{I}}(n_{|N^{\mathrm{I}}|}^{\mathrm{I}}) & TR\_B^{\mathrm{I}}(n_{|N^{\mathrm{I}}|}^{\mathrm{I}}) & Dis^{\mathrm{I}}(n_{|N^{\mathrm{I}}|}^{\mathrm{I}}) \end{bmatrix}, \tag{23}$$

where the values in each column are derived from Eqs. 8, 10, 21, and 22 respectively.

4.1.2 Agent

To ensure comprehensive data exposure, complete state observability, flexible action reconfigurability, and robust long-term computability, the customized IoT system

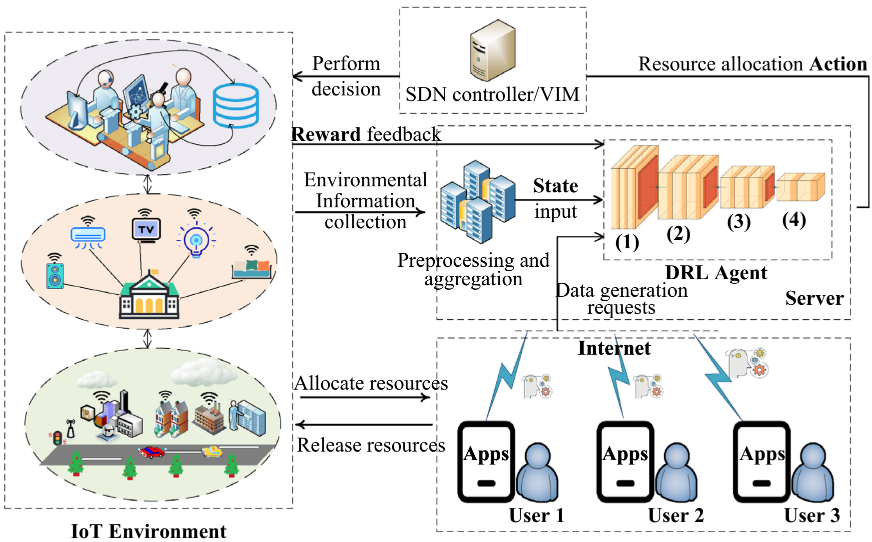

Fig. 3 The IoT framework integrating the proposed DRL model for resource allocation of data generation. The structure of the agent of the DRL model: **1** Extraction layer. **2** Convolution layer. **3** Probability layer. **4** Filtering layer

architecture integrating the DRL model of the proposed algorithm is shown in Fig. 3,. Specifically, IoT devices collect environmental information through sensors and other facilities and transmit it to the central server or the edge node for data preprocessing and aggregation. Then, the processed data serves as the state input of the agent. Among them, the agent is a 4-layer neural network structure, as follows,

- Extraction layer: According to the environmental information, the state matrix is extracted as the input of the agent, as Eq. 23.
- Convolution layer: Use the convolution factor to operate on the state to obtain the available resource vector.
- Probability layer: Based on the available vectors, the softmax function is used to calculate the probability that each IoT node is used to provide resources for data generation.
- Filtering layer: Filter out IoT nodes and links that do not meet relevant conditions according to the constraints in Sect. 3.3, and finally obtain candidate nodes and candidate links.

The agent predicts resource allocation actions based on the current status and DGRs, which are converted into specific instructions and execute resource allocation decisions through the SDN controller or VIM. At the same time, the corresponding IoT resources will be allocated to users, and users will release the resources after they are finished using them. In addition, while executing the decision, environmental reward feedback will be collected for real-time learning and optimization of the agent.

It should be noted that to encourage exploration, we introduce randomness to the policy through the softmax function of the probability layer, which converts the policy output into a probability distribution. At the same time, the temperature parameter is used to control the trade-off between exploration and exploitation. Higher parameter values encourage broader exploration, increasing randomness. Lower parameter values favor strategies that exploit the known maximum probability value, i.e., choosing actions that are more likely to result in high rewards, encouraging exploitation.

In the feedforward process, the agent receives the current state as input and calculates the probability distribution of each action based on the neural network, which will be used to select the next action and obtain the new state and reward based on the selected action's interaction with the environment. It should be noted that due to the existence of randomness in exploration, the policy with the largest probability value is not always selected. During the backward propagation process, the gradient is calculated based on the obtained reward and state transfer information, and the parameters are updated (details in Sect. 4.2). The parameter update process is implemented through backpropagation of the gradient, thereby adjusting the agent's actions to maximize the expected reward.

4.1.3 Action

It is the response of the agent to the environment. The action in this work is a discrete action, which is the decision of node resource allocation, that is, which physical nodes are selected to allocate to DGR. Its specific representation is as follows,

$$a^{\mathsf{V}_i} = \left\{ (a_1^{\mathsf{V}_i}, a_2^{\mathsf{V}_i}, \cdots, a_j^{\mathsf{V}_i}, \cdots, a_{|N^{\mathrm{I}}|}^{\mathsf{V}_i}) | a_j^{\mathsf{V}_i} = \{0, 1\}, \sum_{j=1}^{|N^{\mathrm{I}}|} a_j^{\mathsf{V}_i} = |N^{\mathsf{V}_i}| \right\}. \tag{24}$$

Moreover, the selection of physical links is based on the breadth-first search (BFS) of the candidate physical links after selecting the physical nodes.

4.1.4 Reward

It is a guide for the learning direction of the agent. By continuously interacting with the environment, the agent can learn what actions to take in different states to maximize its expected reward. This work aims to provide energy-efficient resource allocation decisions for IoT data generation, so we define the resource utilization as the reward function, as Eq. 19.

4.2 Model learning

Combined with the gradient backpropagation method and the reward mechanism, the optimization process of the agent is guided. Then the loss function iterates in the following form,

$$\mathcal{L} := \mu \times r_t \times \nabla \mathcal{L}, \tag{25}$$

where $\mathcal{L}$ represents the loss function. It should be noted that this work adopts the cross-entropy loss. $\nabla \mathcal{L}$ represents the gradient. μ represents the learning rate. If its value is too small, the iteration step of the agent will be small, and the convergence will be too slow; if its value is too large, the iteration step of the agent will be large, and it will easily not converge. Therefore, in this work, μ is set to 0.01. In addition, it should be noted that the network environment and status in the IoT scenario are constantly changing. Therefore, we re-extract the instantaneous network environment and status in each training stage to more accurately simulate and predict network behavior, thereby better optimizing resource allocation strategies. The algorithm flow is shown in Algorithm 1.

Algorithm 1 The Learning Process of the proposed MD-VNE algorithm.

Require: G^{I}, G^{V} (training set), Randomly initialize the agent network parameters.
Ensure: Current iteration $iter = 1$, batch size $m = 64$, sum iteration sum_iter.

while $iter \neq sum\_iter$ **do**
  for $G^{\mathsf{V}_i} \in$ batch data **do**
    Extract state by Eq. 23;
    Calculate candidate IoT nodes through forward propagation;
    if $n_j^{\mathsf{V}_i}$ is successfully embedded **then**
      Calculate candidate IoT links through the shortest path algorithm;
      if $l_k^{\mathsf{V}_i}$ is successfully embedded **then**
        Calculate candidate IoT links through the shortest path algorithm;
        Calculate reward by Eq. 19;
        Calculate loss by Eq. 25;
        Update parameters of the agent network;
      else
        VNR embedding failed;
      end if
    else
      VNR embedding failed;
    end if
  end for
  $iter = iter + 1$;
end while

4.3 Time complexity

Time complexity analysis is necessary. For the method proposed in this paper, the complexity of the online training stage is mainly analyzed. For simplicity, let the number of IoT nodes be n. Due to the dynamic characteristics of IoT, each time a new VNR arrives, the state matrix needs to be constructed, and the time complexity of this process is $O(n)$. In addition, the agent is based on a neural network structure, so the time complexity of obtaining actions after processing is $O(n^2)$. After obtaining the node resource allocation action, when performing node embedding, each time a virtual node is successfully embedded, the state matrix needs to be updated. Assuming that q virtual nodes are successfully embedded, the time complexity of this process is $O(q \times n^2)$. In summary, the time complexity of this algorithm is $O(n + n^2 + q \times n^2)$.

5 Experimental result and analysis

5.1 Environment simulation configuration

This work uses the GI-ITM tool to generate simulated IoT environments and DGRs. The parameter configuration is shown in Table 3. Specifically, the IoT network

Table 3 Simulation environment configuration in this work

Parameter	Configuration
Number of IoT domains	4
Number of IoT nodes	100
Number of IoT links	600
Storage of IoT nodes	U[50, 100]
Computing of IoT nodes	U[50, 100]
Bandwidth of IoT links	U[50, 100]
Number of DGRs	2000
Number of DGRs in the training set	1000
Number of DGRs in the testing set	1000
Number of virtual nodes for DGR	2 – 10
Generation probability of virtual link	50%
Virtual storage for DGR nodes	U[1, 20]
Virtual computing for DGRs nodes	U[1, 20]
Virtual bandwidth for DGRs links	U[1, 20]
The learning rate	0.001

includes 4 physical domains, 100 IoT nodes, and 600 IoT links. In addition, a total of 2000 DGRs are generated, half of which are used in the training process and the other half in the testing process. Moreover, to simulate a continuous process, DGRs arrive at the IoT following a Poisson process.

5.2 Training performance

The performance of long-term revenue, long-term resource utilization, and allocation success rate during the training phase are shown in Figs. 4, 5, and 6, respectively. It can be found that in the early stage of iteration, the agent performed poorly because it was in an unfamiliar IoT environment. With the iteration process, the agent will actively adjust its decision-making based on positive feedback from the environment. After about 100 times, all indicators reach a stable state, which shows that the agent can provide reasonable and energy efficient resource allocation decisions for IoT data generation.

5.3 Testing performance

We chose the following classic VNE algorithms as baselines,

- NodeRank (Cheng et al. 2011): A classic heuristic VNE algorithm that allocates resources based on the available resources of nodes.
- CDRL (Yao et al. 2020): A classic RL-based VNE algorithm that utilizes RL to model VNE as a continuous decision-making process.

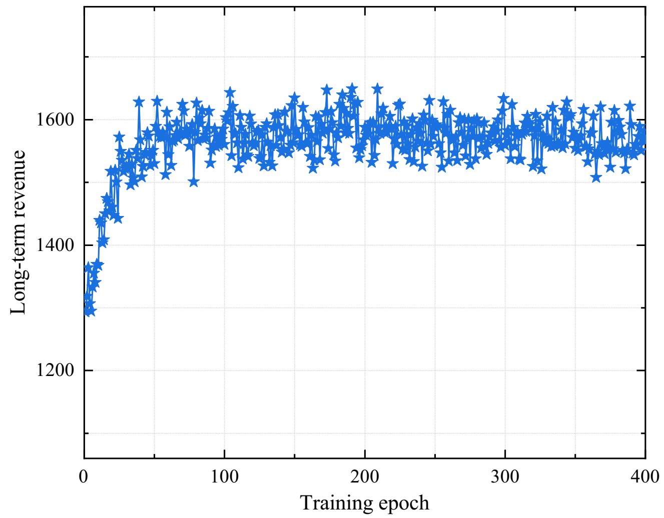

Fig. 4 Long-term revenue in the training

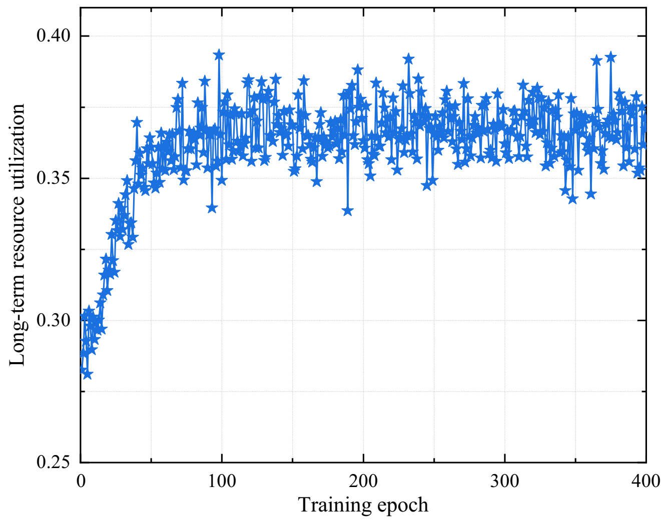

Fig. 5 Long-term resource utilization in the training

- GCNRL (Zhang et al. 2021): A classic VNE algorithm that combines RL and graph convolutional neural network (GCN), which uses GCN to extract available resource attributes and perform dynamic resource allocation based on the fitness matrix.

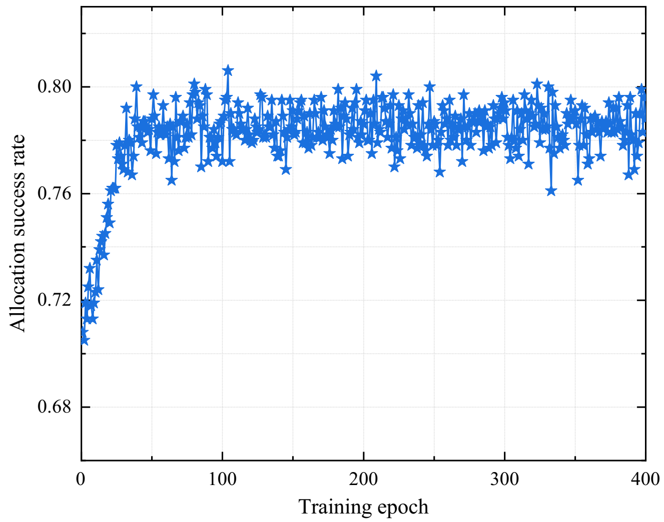

Fig. 6 Allocation success rate in the training

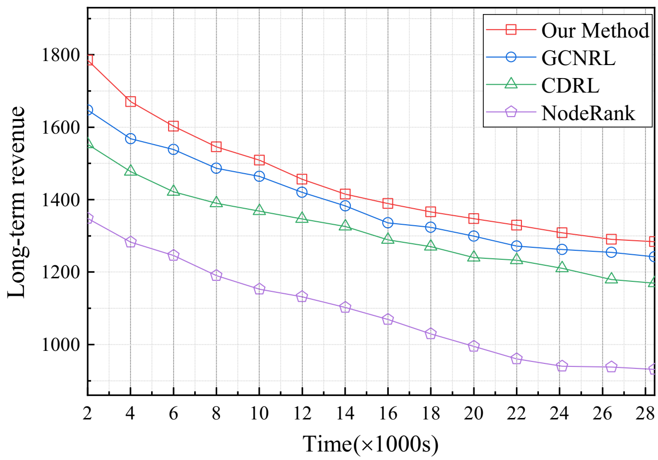

Fig. 7 Long-term revenue in the testing

The comparative performance of each indicator is shown in Figs. 7, 8, and 9, respectively. It can be verified from the comparison results that the RL-based algorithm is better than the heuristic algorithm. In addition, because NodeRank is based on priority allocation of node resources, its early effect is better. However, its performance is less effective in the long run. Based on RL, the CDRL algorithm performs one step further, which comes from effective interaction with the environment. On this basis, GCNRL

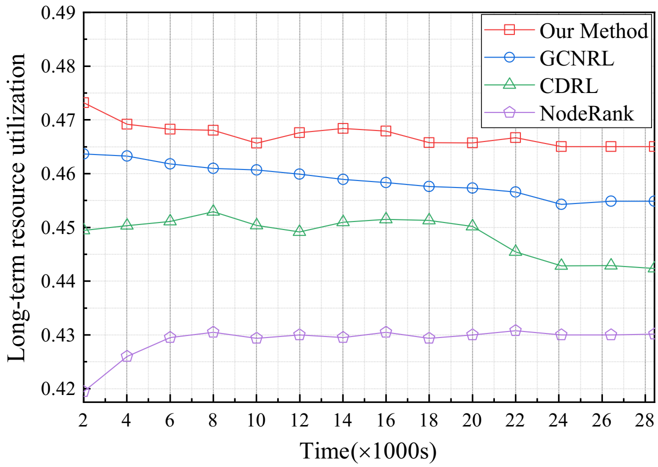

Fig. 8 Long-term resource utilization in the testing

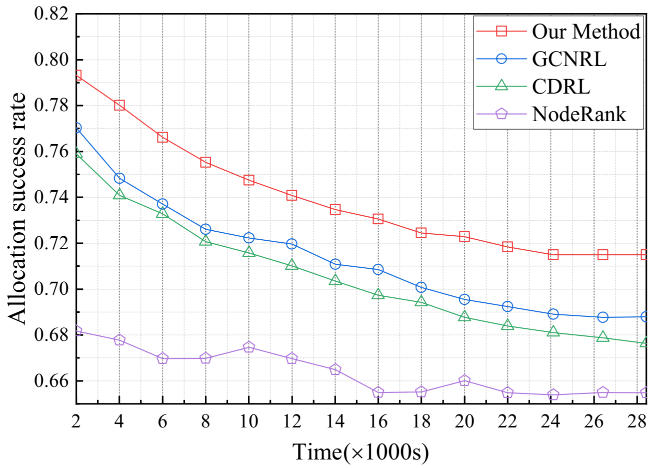

Fig. 9 Allocation success rate in the testing

introduces a deep neural network to improve the agent's perception ability, and its performance is better than the previous two. The method proposed in this work is based on DRL, which not only combines the interactive decision-making ability of RL and the environment, but also combines the ability of deep learning to perceive and extract environmental information. Therefore, it outperforms baselines in all three metrics. This demonstrates that the proposed method can obtain more resource revenue with less resource cost, achieve higher resource utilization as well as successfully implement resource allocation for more data generation tasks.

6 Conclusion and future planning

Data generation is one of the key factors driving IoT development and application. Aiming at its key resource allocation problem, this work proposes a DRL-based VNE algorithm to provide energy efficient resource allocation decision-making. We model IoT as a multi-domain physical network and construct resource constraints with storage, computing, and bandwidth as research objects. In addition, we propose an agent based on a four-layer structure, guiding its optimization through the reward mechanism and the gradient back-propagation mechanism. Finally, we have conducted extensive experiments to verify the effectiveness of the proposed method.

However, in this work, we did not include transmission delay, which is an important QoS metric. In future work, we will explore more deeply the impact of latency on IoT data generation tasks, and seek how to effectively reduce or optimize latency during the resource allocation process. In addition, we plan to study how to employ more advanced machine learning and prediction algorithms to more accurately capture network dynamics and adjust resource allocation strategies in real-time to adapt to these changes.

Acknowledgements This work is partially supported by the Natural Science Foundation of Shandong Province under Grant ZR20231ZH017 (for administration), ZR2022LZH015 (for experimental data collection), and 2023QF025 (for experimental data collection), partially supported by the Open Foundation of Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Qilu University of Technology (Shandong Academy of Sciences) under Grant 2023ZD010 (for comparative experiment), partially supported by the National Natural Science Foundation of China under Grant 62173345 (for checking metrics), partially supported by the open project of the Key Laboratory of All Optical Network and Advanced Telecommunication Network of EMC, Ministry of Education, Beijing Jiaotong University under Grant AON2023K01 (for problem investigation), partially supported by the Integrated Innovation of Science, Education and Industry of Qilu University of Technology (Shandong Academy of Sciences) under Grant 2023PX057 (for comparative experiment), partially supported by the Talent Project of Qilu University of Technology (Shandong Academy of Sciences) under Grant 2023RCKY141 (for model development), partially supported by the RSF project under Grant 22-71-10095 (for checking metrics), partially supported by the Fundamental Research Funds for the Central Universities under Grant 2023JBZY014 (for problem investigation).

Author Contributions Lizhuang Tan: Conceptualization, Methodology, Writing - review & editing, Funding acquisition. Amjad Aldweesh: Conceptualization, Investigation, Methodology. Ning Chen: Conceptualization, Investigation, Methodology, Validation, Writing - original draft, Writing - review & editing. Jian Wang: Conceptualization, Writing - review & editing, Funding acquisition. Jianyong Zhang: Conceptualization, Writing - review & editing. Yi Zhang: Conceptualization, Writing - review & editing. Konstantin Igorevich Kostromitin: Investigation, Methodology. Peiying Zhang: Conceptualization, Funding acquisition. All authors reviewed the manuscript.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

Ahmed, E., Yaqoob, I., Hashem, I.A.T., Khan, I., Ahmed, A.I.A., Imran, M., Vasilakos, A.V.: The role of big data analytics in internet of things. Comput. Netw. **129**, 459–471 (2017)

Al-Hadhrami, Y., Hussain, F.K.: Real time dataset generation framework for intrusion detection systems in IoT. Futur. Gener. Comput. Syst. **108**, 414–423 (2020)

Chen, X., Wang, L., Li, C., Yang, J., Lu, Z., Lu, G., Gu, Y., Jiang, Y.: Resource distribution equilibrium for virtual network embedding over flexi-grid optical networks. J. Lightwave Technol. **39**(15), 4894–4908 (2021)

Chen, N., Zhang, P., Kumar, N., Hsu, C.-H., Abualigah, L., Zhu, H.: Spectral graph theory-based virtual network embedding for vehicular fog computing: a deep reinforcement learning architecture. Knowl.-Based Syst. **257**, 109931 (2022)

Chen, N., Shen, S., Duan, Y., Huang, S., Zhang, W., Tan, L.: Non-euclidean graph-convolution virtual network embedding for space-air-ground integrated networks. Drones **7**(3), 165 (2023)

Chen, N., Shen, S., Duan, Y., Huang, S., Zhang, W., Tan, L.: Non-euclidean graph-convolution virtual network embedding for space-air-ground integrated networks. Drones **7**(3), 165 (2023). https://doi.org/10.3390/drones7030165

Cheng, X., Su, S., Zhang, Z., Wang, H., Yang, F., Luo, Y., Wang, J.: Virtual network embedding through topology-aware node ranking. ACM SIGCOMM Comput. Commun. Rev. **41**(2), 38–47 (2011)

Chettri, L., Bera, R.: A comprehensive survey on internet of things (IoT) toward 5G wireless systems. IEEE Internet Things J. **7**(1), 16–32 (2020). https://doi.org/10.1109/JIOT.2019.2948888

Congress, S., Puppala, A.: Eye in the sky: condition monitoring of transportation infrastructure using drones. In: Proceedings of the Institution of Civil Engineers-Civil Engineering, vol. 176, pp. 40–48 (2022). Thomas Telford Ltd

Dash, S., Yale, A., Guyon, I., Bennett, K.P.: Medical time-series data generation using generative adversarial networks. In: Artificial Intelligence in Medicine: 18th International Conference on Artificial Intelligence in Medicine, AIME 2020, Minneapolis, MN, USA, August 25–28, 2020, Proceedings 18, pp. 382–391 (2020). Springer

Du, H., Wang, J., Niyato, D., Kang, J., Xiong, Z., Guizani, M., Kim, D.I.: Rethinking wireless communication security in semantic internet of things. IEEE Wirel. Commun. **30**(3), 36–43 (2023). https://doi.org/10.1109/MWC.011.2200547

Duan, Y., Chen, N., Shen, S., Zhang, P., Qu, Y., Yu, S.: FDSA-STG: Fully dynamic self-attention spatio-temporal graph networks for intelligent traffic flow prediction. IEEE Trans. Veh. Technol. **71**(9), 9250–9260 (2022). https://doi.org/10.1109/TVT.2022.3178094

Duan, Y., Chen, N., Zhang, P., Kumar, N., Chang, L., Wen, W.: MS2GAH: multi-label semantic supervised graph attention hashing for robust cross-modal retrieval. Pattern Recogn. **128**, 108676 (2022)

Duan, Y., Chen, N., Bashir, A.K., Alshehri, M.D., Liu, L., Zhang, P., Yu, K.: A web knowledge-driven multimodal retrieval method in computational social systems: unsupervised and robust graph convolutional hashing. IEEE Trans. Comput. Soc. Syst. (2022). https://doi.org/10.1109/TCSS.2022.3216621

Feng, Q., Li, D., Wu, Z.: A data envelopment analysis approach for resource allocation and reallocation. IEEE Trans. Eng. Manage. **71**, 3295–3307 (2024). https://doi.org/10.1109/TEM.2023.3340302

Hsu, I., Huang, K.-H., Boschee, E., Miller, S., Natarajan, P., Chang, K.-W., Peng, N., et al.: Degree: a data-efficient generation-based event extraction model. arXiv preprint arXiv:2108.12724 (2021)

Liu, Y., Zhang, J.: Service function chain embedding meets machine learning: deep reinforcement learning approach. IEEE Trans. Netw. Serv. Manage. (2024). https://doi.org/10.1109/TNSM.2024.3353808

Sun, G., Li, Y., Yu, H., Vasilakos, A.V., Du, X., Guizani, M.: Energy-efficient and traffic-aware service function chaining orchestration in multi-domain networks. Futur. Gener. Comput. Syst. **91**, 347–360 (2019)

Wang, L., Zheng, Z., Chen, N., Chi, Y., Liu, Y., Zhu, H., Zhang, P., Kumar, N.: Multi-target-aware energy orchestration modeling for grid 2.0: a network virtualization approach. IEEE Access **11**, 21699–21711 (2023). https://doi.org/10.1109/ACCESS.2023.3251698

Woodbridge, J., Anderson, H.S., Ahuja, A., Grant, D.: Predicting domain generation algorithms with long short-term memory networks. arXiv preprint arXiv:1611.00791 (2016)

Wu, S., Chen, N., Xiao, A., Jia, H., Jiang, C., Zhang, P.: AI-enabled deployment automation for 6G space-air-ground integrated networks: challenges, design, and outlook. IEEE Netw. (2024). https://doi.org/10.1109/MNET.2024.3368753

Wu, S., Chen, N., Xiao, A., Zhang, P., Jiang, C., Zhang, W.: AI-empowered virtual network embedding: a comprehensive survey. IEEE Commun. Surv. Tutor. (2024). https://doi.org/10.1109/COMST.2024.3424533

Wu, S., Chen, N., Wen, G., Xu, L., Zhang, P., Zhu, H.: Virtual network embedding for task offloading in IIoT: a DRL-assisted federated learning scheme. IEEE Trans. Industr. Inf. (2024). https://doi.org/10.1109/TII.2024.3353848

Xiao, A., Chen, N., Wu, S., Zhang, P., Cao, S., Jiang, C.: DNFS-VNE: deep neuro fuzzy system driven virtual network embedding. arXiv e-prints, 2310 (2023)

Yao, Y., Zheng, L., Yang, X., Naphade, M., Gedeon, T.: Simulating content consistent vehicle datasets with attribute descent. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16, pp. 775–791 (2020). Springer

Yao, H., Ma, S., Wang, J., Zhang, P., Jiang, C., Guo, S.: A continuous-decision virtual network embedding scheme relying on reinforcement learning. IEEE Trans. Netw. Serv. Manage. **17**(2), 864–875 (2020)

Zhan, K., Chen, N., Santhosh Kumar, S.V.N., Kibalya, G., Zhang, P., Zhang, H.: Edge computing network resource allocation based on virtual network embedding. Int. J. Commun. Syst. (2022). https://doi.org/10.1002/dac.5344

Zhang, Y., Li, C., Chen, N., Zhang, P.: Intelligent requests orchestration for microservice management based on blockchain in software defined networking: a security guarantee. In: 2022 IEEE International Conference on Communications Workshops (ICC Workshops), pp. 254–259 (2022). IEEE

Zhang, P., Wang, C., Kumar, N., Zhang, W., Liu, L.: Dynamic virtual network embedding algorithm based on graph convolution neural network and reinforcement learning. IEEE Internet Things J. **9**(12), 9389–9398 (2021)

Zhang, P., Li, Y., Kumar, N., Chen, N., Hsu, C.-H., Barnawi, A.: Distributed deep reinforcement learning assisted resource allocation algorithm for space-air-ground integrated networks. IEEE Trans. Netw. Serv. Manage. **20**(3), 3348–3358 (2023). https://doi.org/10.1109/TNSM.2022.3232414

Zhang, P., Chen, N., Li, S., Choo, K.-K.R., Jiang, C., Wu, S.: Multi-domain virtual network embedding algorithm based on horizontal federated learning. IEEE Trans. Inf. Forensics Secur. **18**, 3363–3375 (2023). https://doi.org/10.1109/TIFS.2023.3279587

Zhang, P., Chen, N., Xu, G., Kumar, N., Barnawi, A., Guizani, M., Duan, Y., Yu, K.: Multi-target-aware dynamic resource scheduling for cloud-fog-edge multi-tier computing network. IEEE Trans. Intell. Transp. Syst. (2023). https://doi.org/10.1109/TITS.2023.3330419

Zhang, P., Chen, N., Shen, S., Yu, S., Kumar, N., Hsu, C.-H.: AI-enabled space-air-ground integrated networks: management and optimization. IEEE Netw. **38**(2), 186–192 (2024). https://doi.org/10.1109/MNET.131.2200477

Zhang, P., Chen, N., Kumar, N., Abualigah, L., Guizani, M., Duan, Y., Wang, J., Wu, S.: Energy allocation for vehicle-to-grid settings: a low-cost proposal combining DRL and VNE. IEEE Trans. Sustain. Comput. **9**(1), 75–87 (2024). https://doi.org/10.1109/TSUSC.2023.3307551

Zhao, D., Lu, Y., Li, X., Li, Z., Liu, Y.: Cross-domain service function chain routing: multiagent reinforcement learning approaches. IEEE Trans. Circuits Syst. II Express Briefs **69**(12), 4754–4758 (2022). https://doi.org/10.1109/TCSII.2022.3183345

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

Authors and Affiliations

Lizhuang Tan[1,2] · Amjad Aldweesh[3] · Ning Chen[4] · Jian Wang[5] · Jianyong Zhang[6] · Yi Zhang[9] · Konstantin Igorevich Kostromitin[8] · Peiying Zhang[1,7]

✉ Ning Chen
nchen@bupt.edu.cn

✉ Jianyong Zhang
jyzhang@bjtu.edu.cn

Lizhuang Tan
tanlzh@sdas.org

Amjad Aldweesh
A.aldweesh@su.edu.sa

Jian Wang
wangjiannl@upc.edu.cn

Yi Zhang
zhangyi.upc@qq.com

Konstantin Igorevich Kostromitin
kostromitinki@susu.ru

Peiying Zhang
zhangpeiying@upc.edu.cn

1 Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), 3501 Daxue Road, Jinan 250013, Shandong, China

2 Shandong Provincial Key Laboratory of Computer Networks, Shandong Fundamental Research Center for Computer Science, 28789 Jingshi East Road, Jinan 250013, Shandong, China

3 College of Computing and Information Technology, Shaqra University, Al Duwadimi Road, Shaqra 15273, Riyadh, Saudi Arabia

4 School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, 10 Xitucheng Road, Beijing 100876, China

5 College of Science, China University of Petroleum (East China), 66 Changjiang West Road, Qingdao 266580, Shandong, China

6 Institute of Lightwave Technology, Beijing Jiaotong University, 3 Shangyuan Village, Beijing 100044, China

7 Qingdao Institute of Software, College of Computer Science and Technology, China University of Petroleum (East China), 66 Changjiang West Road, Qingdao 266580, Shandong, China

8 Department of Physics of Nanoscale Systems, South Ural State University, 76 Lenin Avenue, Chelyabinsk 454080, Russian Federation

9 School of Information and Communication Engineering, University of Electronic Science and Technology of China, 2006 West Avenue Road, Chengdu 611731, Sichuan, China