Privacy-Preserving Deployment Mechanism for Service Function Chains across Multiple Domains

Jun Cai, Zirui Zhou, Zhongwei Huang, Wenlong Dai, and Fei Richard Yu, *Fellow, IEEE*

***Abstract*—Network function virtualization (NFV) has attracted attention because of its flexible configuration and management of network functions. Based on NFV, the service function chain (SFC) defines a group of virtual network functions (VNFs) connected sequentially, enabling flexible customization and provisioning of network services. In the large-scale and heterogeneous Internet of Things (IoT) environment, e.g., industrial IoT, servers provided by a single infrastructure provider (InP) cannot support the deployment of all VNFs, and SFCs must be deployed across multiple domains. However, SFCs deployed across multiple domains will inevitably bring privacy leakage and resource coordination difficulties, thereby reducing the efficiency of network services. To address these issues, this paper proposes a privacy-preserving deployment mechanism (PPDM) for SFCs that achieves near-optimal SFC deployment across multiple domains while protecting resource and topology privacy. PPDM first performs virtual resource prediction and forms the service intention response matrix (SIRM) based on SFC requests (SFCRs). Second, the multi-domain controller (MDC) discovers a near-optimal SFCs deployment strategy by deep Q-network (DQN) using SIRM as input to protect domains' privacy. Finally, the learned strategies are distributed to intra-domain controllers (IDCs) to implement specific services. Simulation results demonstrate that the proposed method outperforms privacy-preserving and non-privacy-preserving methods.**

***Index Terms*—Service function chain (SFC), industrial Internet of Things (IIoT), privacy protection, resource prediction, binary response, deep Q-network (DQN).**

I. Introduction

THE Internet of Things (IoT) has significantly expedited people's lives and productivity in recent years due to its quick growth and widespread applicability. The interconnection of everything has become an irresistible trend along with the increasing network scale, and more recently, the combination of IoT and industry has created the industrial Internet of Things (IIoT). Due to the numerous and widely dispersed IIoT devices, the single infrastructure providers (InPs) are unable to implement all network functions (NFs), such as security NFs, throughout the entire region [1]. Deploying NFs across several domains is now essential to ensuring effective service provisioning in the large-scale IIoT.

The network function virtualization (NFV) technology decouples NFs from proprietary hardware, enabling low-cost customization and flexible provisioning of NFs. Based on NFV technology, IoT users define a series of virtual network functions (VNFs) of the service function chain (SFC) to deal with massive and heterogeneous IoT services flexibly [2]. To date, SFC deployment in a single domain has been extensively studied [3], [4]. In contrast, deploying SFCs across multiple domains, which have a wide range of potential applications (e.g., Satellite or Ground networks and IIoT), requires further investigation due to some unsolved challenges [5]: 1) Each domain has its own network operator in large-scale multiple domains network and applies different management and operation strategies. 2) Domains are not willing to disclose confidential information (e.g., available resources, deployment strategies and topology information [6], etc) and expect to maintain their autonomy [7]. 3) Due to privacy concerns, information interactions between domains are frequently lacking, which severely impedes the service process of a multi-domain network. Previous studies [8], [9], achieve the delay and cost minimization deployment of SFCs across multiple domains by utilizing a small amount of resource information supplied by intra-domain controllers (IDCs). However, providing even a small amount of resource information compromises the privacy of domains. We argue that the best way to protect privacy is to forbid uploading any private information from each domain.

Therefore, this paper proposes a privacy-preserving deployment mechanism (PPDM) for SFCs across multiple domains, based on intra-domain binary response and resource prediction, which completely hides the real resource and the network topology information in each domain. Specifically, the multi-domain controller (MDC) first sends SFC requests (SFCRs) to IDCs, and nodes in each domain will return the binary response ξ to IDC that indicates the deployment intention of each VNF in SFC (i.e., if accept $\xi = 1$, or reject $\xi = 0$). Second, the IDC will predict the virtual resource capacity based on the node's binary response to match the nodes with sufficient

Manuscript received February 12, 2023; revised June 28, 2023; accepted August 29, 2023. This work was supported by the National Natural Science Foundation of China (61972104, 61902080, 62002072, 61702120), the National Key Research and Development Program of China (2019YFB1804403, 2018YFB1802200), the Special Projects in Key Areas of Guangdong Province (2019B010118001), the Science and Technology Project in Guangzhou (201803010081), Guangzhou Key Laboratory (202102100006) and Science and Technology Program of Guangzhou, China (202002020035), the Research Projects in Guangdong Province (2021ZDJS026). Corresponding author: Zhongwei Huang (jonwong.must@gmail.com).

Jun Cai and Wenlong Dai are with the School of Cyber Security, Guangdong Polytechnic Normal University, Guangzhou 510665, China (email: caijun@gpnu.edu.cn; daiwenlong23@163.com).

Zirui Zhou is with the GuangZhou City Construction College, Guangzhou 510665, China (email: zr1270388293@163.com).

Zhongwei Huang is with the School of Computer Science and Engineering, Macau University of Science and Technology, Macao, China, and also with the Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen University, 518107, P.R. China. (e-mail: jonwong.must@gmail.com)

F. Richard Yu is with the Shenzhen Key Lab of Digital and Intelligent Tech.&Sys. and Guangdong Lab of Artificial Intelligence and Digital Economy (SZ), Shenzhen University, 518107, P.R. China (email: yufei@szu.edu.cn).

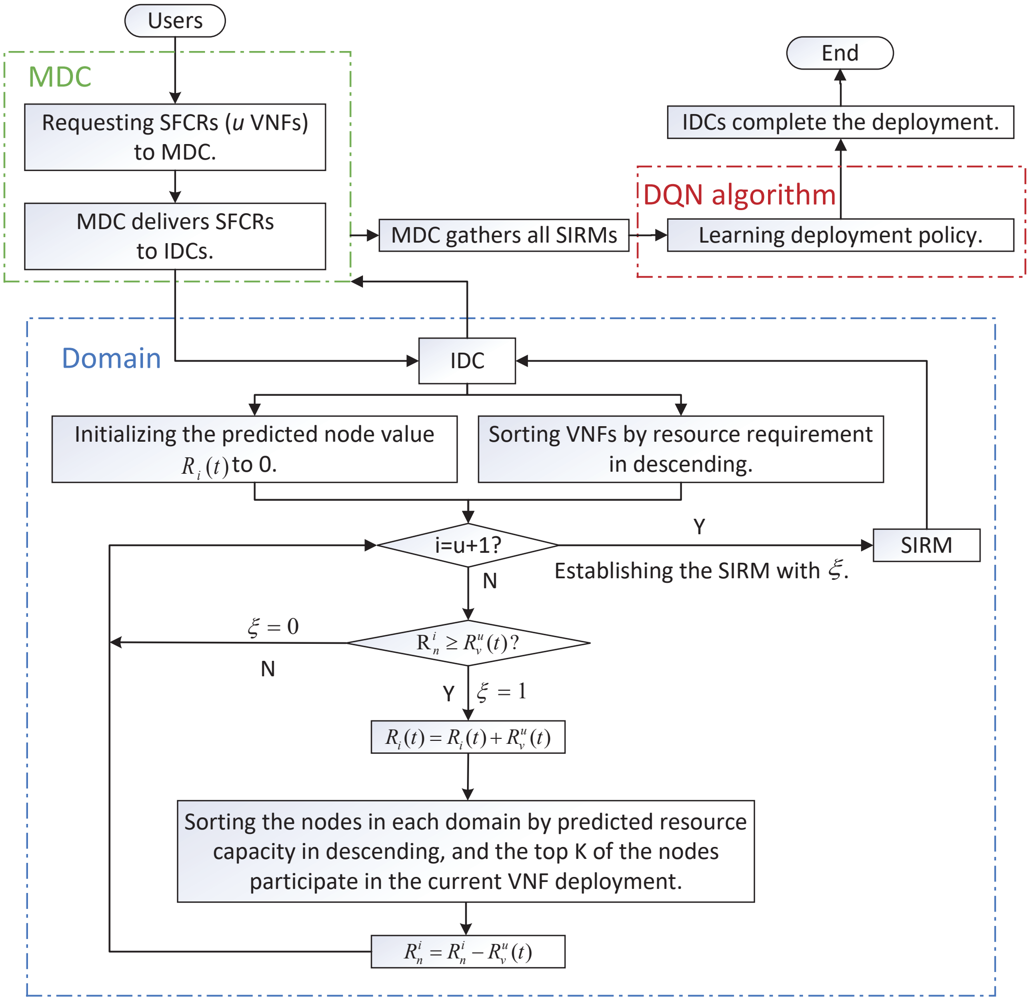

Users
MDC
Requesting SFCRs (u VNFs) to MDC.
MDC delivers SFCRs to IDCs.
MDC gathers all SIRMs
DQN algorithm
Learning deployment policy.
IDCs complete the deployment.
End
Domain
IDC
Initializing the predicted node value $R_i(t)$ to 0.
Sorting VNFs by resource requirement in descending.
i=u+1?
Y
N
SIRM
Establishing the SIRM with ξ.
$\xi = 0$
N
$R_n^i \geq R_v^u(t)$?
Y
$\xi = 1$
$R_i(t) = R_i(t) + R_v^u(t)$
Sorting the nodes in each domain by predicted resource capacity in descending, and the top K of the nodes participate in the current VNF deployment.
$R_n^i = R_n^i - R_v^u(t)$

Fig. 1. Flow chart for the model of the paper.

resources to participate in the current VNF deployment and return the service intention response matrix (SIRM) to MDC. Then, MDC learns the deployment strategies of SFCs across domains based on deep Q-network (DQN) algorithm which takes the SIRM as the input. The flow chart of the proposed method is shown in Fig. 1. In short, the main contributions of this paper are as follows.

- A hierarchical multi-domain SFCs deployment scheme is proposed, with the MDC acting as the central controller to interact with IDCs. Each IDC coordinates the nodes and realizes the specific service deployments in the corresponding domain.
- The SIRM-based privacy-preserving method is proposed to prevent the disclosure of intra-domain information. The upload SIRM strictly maintains privacy within the domain based on binary response and resource prediction.
- The DQN-based cross-domain deployment algorithm (CDDA) is proposed to achieve the efficient deployment of SFCs. DQN receives SIRM as input to learn the SFCs deployment strategy. Moreover, we propose an SFCs acceptance rate optimization algorithm (AROA) to improve the SFCs deployment acceptance rate.
- We implemented and compared PPDM to existing privacy-preserving and non-privacy-preserving algorithms, to demonstrate the effectiveness of PPDM in privacy protection and deployment performance.

The remainder of this article is organized as follows. Section II introduces the related works. Section III presents the system model and problem formulation. Section IV designs and describes the process of the proposed algorithms. Section V implements and evaluates the simulation results, Finally, Section VI concludes this paper.

II. Related Works

To date, researchers have investigated the SFCs deployment problem both in single-domain networks [10], [11] as well as multi-domain networks [12], [13]. The deployment of SFCs in single-domain networks simply takes into account the factors of cost, latency, and resource consumption, because there is no third party involved. In contrast, InPs are unwilling to reveal detailed topology and resource information to a third party in the multi-domain scenario. It is difficult for multiple domains to reach a consensus, which significantly hinders the SFCs deployment process. In this paper, we study the privacy-preserving cross-domain SFC deployment problem based on deep reinforcement learning (DRL) technique.

TABLE I
SUMMARY OF RELATED WORKS

Literature	**Contribution**	**Technology**	**Advantage**	**Inferiority**
Wang *et al.* [14]	it is impossible for unauthorized users to obtain data information without the decryption method	distortion, encryption, and anonymity	brought powerful encryption capabilities	high complexity and slow encryption speed
Mano *et al.* [13]	employed secure multi-party computation to protecte the privacy of virtual networks	SMPC	provided a new offence for cross-domain privacy protection	the proposed method is complex and time-consuming
Wang *et al.* [15]	employed system to ensure the privacy and security of a large amount of IIoT data by injecting noise interference	DP	ensured the privacy and security of large amounts of data	distorted the data and the information of nodes, which reduced the availability of nodes
N. Toumi *et al.* [9]	uploaded insensitive information by each domain to diminish privacy and security risks	LPP, NLPP, and GPP	guaranteed the privacy of some information within the system	still exposed some private information, which will affect the system's performance
Joshi et al. [16]	used a learning algorithm to efficiently deploy cross-domain SFCs based on the uploaded non-sensitive information	pSMART	protected the privacy to some extent and optimized the response time of MD-SFC orchestration	still exposed some private information
Wang *et al.* [17]	formulated as an ILP problem and employed a distributed architecture to enable cross-domain deployment of SFC	ILP, two heuristic algorithm	realized the cross-domain deployment of SFC	still caused privacy disclosure
Dietrich *et al.* [12]	proposed a distributed architecture to enable cross-domain deployment of SFC	ILP, NF-Graph partitioning and embedding	exchanged critical information which they need to function properly	only considered a two-domains network, which is inapplicable to larger networks
Q. Zhang *et al.* [18]	employed a distributed architecture to enable cross-domain deployment of SFC	heuristic algorithm	utilized the concept of VNFs deployment distribution across multiple domains to effectively deploy an SFC in the correct order	brought higher end-to-end delay and resource consumption
Tusa *et al.* [19]	designed a hierarchical orchestration scheme for SFC, utilizing the global awareness and control capabilities of SDN technology	SDN, HSP	end-to-end slices across the whole infrastructure provide a more effective resource management and also better support the customers mobility requirements	do not suitable for complex networks
Liu *et al.* [7]	proposed a general distributed method (GDM) to deploy SFCs across multiple domains	ILP, SDN, GDM	resolved the issue of SFC cross-domain deployment and enable the system to have better scalability	may reduce the deployment successful rate
Pham *et al.* [20]	proposed a VNF-FG deployment strategy with automated inter-domain load balancing	DDPG	automated inter-domain load balancing and improved SFC acceptance rate	no significant performance advantage in terms of delay, resource utilization, and other aspects
Pei *et al.* [21]	achieved optimal SFCs deployment	DDQN	minimized the VNF placement cost, operation cost, and penalty cost	only considered single-domain networks
Liu *et al.* [22]	innovatively proposed a DRL-based framework for deploying dynamic SFCs that combined MEC and NFV	DDPG	ensured effective and quick implementation of SFC orchestration	only considered single-domain networks
Shah *et al.* [23]	each SFCR was executed by a DRL agent to achieved optimal SFCs deployment	Multi-Agent DQL	achieved more efficient deployment of SFC; centralized controller to make centralized decisions	only considered single-domain networks

Therefore, this paper will summarize the related works from three perspectives: intra-domain privacy protection, SFC cross-domain deployment, and SFC deployment method based on DRL. We further summarize relevant works by comparing the contribution, technology, advantage, and inferiority, as shown in Table I.

A. Intra-Domain Privacy Protection

Researchers have proposed many strategies to solve privacy threats in various scenarios. Qi *et al.* [24] described evaluation criteria and privacy protection algorithms in data mining, including distortion, encryption, privacy, and anonymity. Among them, encryption [14] is one of the most common privacy protection technologies in IIoT and even most scenarios. Although it's difficult for unauthorized people to get real information after encryption, this approach still has certain drawbacks, including high algorithm complexity and sluggish encryption speed, etc. Mano *et al.* [13] employed secure multi-party computation (SMPC) to protect the privacy of virtual networks. However, SMPC is still complex and time-consuming to apply in a large-scale network scenario. To address the issues mentioned above, the differential privacy (DP) method [25] proposed in 2006 provides robust privacy protection by minimizing the likelihood of information identification, i.e., by injecting signal perturbations into original data sets. Wang *et al.* [15] ensured the privacy and security of large amounts of IIoT data by injecting noise interference. However, they distorted the data, and the addition of noise disturbance will distort the information of nodes and even reduce their availability, which will adversely affect the SFC's deployment performance. Toumi *et al.* [9], let each domain upload non-sensitive information through linear physical programming (LPP), non-linear physical programming (NLPP), and global physical programming (GPP) methods, which reduced privacy and security risks to a certain extent. Joshi *et al.* [16] also used a learning algorithm to efficiently deploy cross-domain SFCs based on the uploaded non-sensitive information from each domain. However, these methods still bring information uploading and privacy exposure. Therefore, it cannot be applied to the scenario we have proposed.

In short, privacy protection methods have been extensively studied. However, these methods are not fully applicable to our proposed scenario, particularly in a large-scale network where SFC requires a prompt response. Consequently, this study employs a novel resource prediction and binary response method and generates SIRM through the collaboration of the MDC and IDCs, i.e., each node in the domain only provides the service deploy intention. Our proposed method aims to prevent the uploading of resource information and protects topology privacy in each domain, while realizing the SFCs deployment across multiple domains.

B. SFC Cross-Domain Deployment

In recent years, researchers have also proposed many optimization models and solutions for SFCs deployment across multiple domains. For example, Wang *et al.* [17] formulated the embedding of cross-domain SFCs as an integer linear programming (ILP) problem, and two time-efficient heuristics approaches were proposed to solve the SFC cross-domain deployment. However, they only considered a simple network with two domains, which is inapplicable to larger networks. Dietrich *et al.* [12] assumed that each domain could share key information with a third party and deployed SFC to the physical network from a global perspective to meet users' requirements. A central controller is introduced to monitor the physical network, collecting the resource information of each domain to facilitate the cross-domain deployment of SFCs. With the expansion and upgrade of SFCs, the distributed cross-domain scenario urgently requires additional research. Q. Zhang *et al.* [18] proposed an algorithm to find all feasible mappings of a fixed-order or flexible-order SFC request in multi-domain networks. However, the aforementioned techniques will significantly increase resource consumption and end-to-end delay. Tusa *et al.* [19] designed the hierarchically structured service provider (HSP) for cross-domain SFC, based on the global awareness and control capabilities of software-defined network (SDN) [26] technology. Liu *et al.* [7] also realized cross-domain service deployment via the global awareness of SND controller and proposed a general distributed method (GDM) to deploy SFCs across multiple domains. GDM first divides the SFC and forms multiple sub-SFCs, then each domain is allocated to deploy sub-SFCs according to the deployment strategy. However, the two-stage optimization approach may reduce the success rate, and they didn't propose the corresponding method to optimize it. Nevertheless, the hierarchical architecture helps to avoid information interaction between domains. As a result, this study considers using a hierarchical architecture to achieve cross-domain SFCs deployment, in which the MDC is responsible for interacting with the IDC of each domain while avoiding interaction of domains.

C. SFC Deployment Method Based on DRL

Traditional optimization methods, such as exact [27], heuristic [28], and meta-heuristic [29] algorithms, are capable of optimizing the SFCs deployment process to some extent. However, it is challenging to find an optimal solution with these traditional methods for some practical problems, particularly in dynamic and heterogeneous IIoT scenarios. Therefore, it is necessary to identify an intelligent deployment method capable of rapidly obtaining the solution in complex scenarios.

In recent years, DRL has received significant attention in the areas of workload balancing [30], network resource management [31], routing [32], and other problems. Inspired by this, researchers have attempted to study the intelligent model to achieve the cross-domain deployment of SFC [33]. For example, Pham *et al.* [20] incorporated Deep Deterministic Policy Gradient (DDPG) method into a cross-domain problem, they demonstrate the effectiveness of the suggested VNF-FG deployment strategy with automated inter-domain load balancing. Pei *et al.* [21] described the SFCs deployment problem as a binary integer programming (BIP) issue in a single-domain network. By integrating the double deep Q-network (DDQN) method, the goal was to achieve optimal SFCs deployment by minimizing the VNF placement cost, operation cost, and penalty cost. Liu *et al.* [22] innovatively proposed a DDPG-based framework for deploying dynamic SFCs that combined mobile edge computing (MEC). By utilizing the computing and autonomous learning capability of the cloud and the edge, an intelligent network system is constructed to effectively and quickly implement the SFCR response. Shah *et al.* [23] regard the SFCs deployment as a distributed problem and proposed an approach based on multi-agent DQL method. Each SFCR was executed by a agent, and all individual decisions were aggregated into a general decision made by the central controller to deploy SFCs. Although the preceding methods were utilized in simple networks, they also paved the way for deploying SFCs across domains. Due to the better robustness and convergence performance of the DQN algorithm compared with other DRL algorithms [34]. We utilized the DQN algorithm to realize the cross-domain deployment of SFCs and takes the binary response matrix as the input of the DQN to protect the privacy of each domain.

III. System Model and Problem Description

This section describes the hierarchical system model, privacy protection, and specific problem description of resource prediction and intra-domain response, as well as the mathematical formulations.

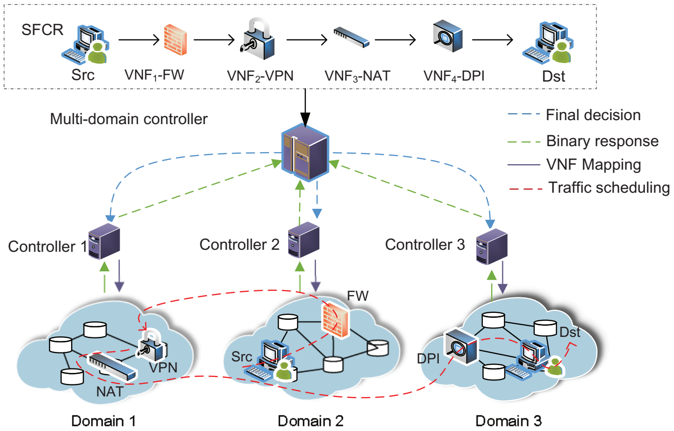


Fig. 2. Cross-domain SFCs deployment architecture.

TABLE II
NOTATIONS

Symbol	Description
S	A set of SFCs
N	A set of nodes
L	A set of links, $l \in L$
E	A set of routing paths
N_n	n-th domain in N
N_n^i	i-th node in domain N_n
$l_n^{ii'}$	Link between nodes N_n^i and $N_n^{i'}$
$l_{nn'}^{ii'}$	Link between nodes N_n^i and $N_{n'}^{i'}$
d_u^i	u-th VNF processing delay deployed in node N_n^i
$d_n^{ii'}$	Link delay of $l_n^{ii'}$
$d_{nn'}^{ii'}$	Link delay of $l_{nn'}^{ii'}$
ξ	VNF deploy intention, ξ = {0,1}
S	A set of SFCs
V	A set of VNFs
$s_s(t)$	s-th SFC entering the system at time t
$v_u(t)$	u-th VNF entering the system at time t
$R_v^u(t)$	Resources requirement of VNF $v_u(t)$ at time t
R_n^i	Real resource capacity of node N_n^i
$R_i(t)$	Predicted resources capacity of node N_n^i
X	Whether R_n^i is sufficient to deploy SFC $s_s(t)$
$X_{v_u}^{N_n^i}$	Indicate whether $v_u(t)$ is deployed on node N_n^i
$Y_l^{i,j}$	Whether the virtual link l is mapped between nodes N_n^i and N_n^j
$P_i(t)$	Resource utilization of node N_n^i
$P_N(t)$	Resource utilization of all nodes N
d_{prop}	Propagation delay
d_{tran}	Transmission delay
t_{proc}	Processing time of each data packet
d_{proc}	Processing delay
$\vartheta_t(N_n^i)$	Load utilization of node N_n^i at time t
$O_i(t)$	Available load of node N_n^i
$O_u(t)$	Load of VNF $v_u(t)$
L_{packet}	Data packet length
dis	Distance between two nodes in the network
c	Signal propagation speed in the physical link
f	Packet transmission rate
σ_o	Data packet rate
σ_u^t	Packet rate of VNF $v_u(t)$
τ_u^t	Packet processing rate in VNF $v_u(t)$
W^n	Service intention response matrix(SIRM) of domain N_n
W	Overall SIRM
$r(s,a)$	Obtained immediate reward
$D_{end}^s(t)$	End-to-end delay of SFC $s_s(t)$
$r(\delta\|s,a)$	Immediate reward obtained by ξ
$r(\rho\|s,a)$	Immediate reward obtained by estimating whether the nodes selected by MDC are the same
$o_{P_N(t)}$	Normalization of $P_N(t)$
$o_{D_{end}^s(t)}$	Normalization of $D_{end}^s(t)$
K	The proportion of response nodes

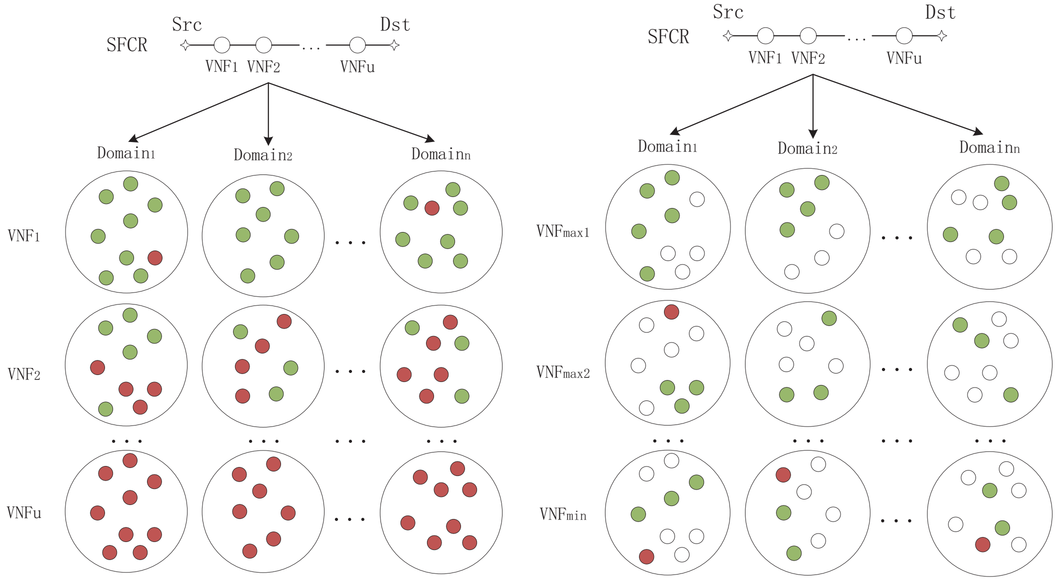
A. Hierarchical Model

We employ a hierarchical framework for cross domain SFC deployment, as shown in Fig. 2, where the MDC acts as the central controller, which connects all IDCs to achieve the connection between users and InPs. Assuming that SFCRs enter the system according to a Poisson distribution, MDC will immediately send requirements to each domain controller. When receiving the SFCR, each IDC will compare intra-domain nodes with the SFCR and return the service deployment intention ξ. Accept (ξ = 1) indicates that the node has sufficient resources for VNF deployment, whereas reject (ξ = 0) indicates that there are insufficient resources. Notably, if the node responds with accept, it assumes that resources are occupied and will not release before the SFC departure. This situation is referred to as “virtual occupancy”. The MDC will combines the binary response returned by each domain to form a global perspective, i.e., SIRM. The DQN takes the SIRM as input and learns the deployment strategy. Finally, MDC allocates the deployment strategies to each IDC for SFCs implementation in each domain.

B. Node Response Process Optimization

During the response process, the nodes will continuously update its state whenever they receive a new SFCR at time t. Nonetheless, if all available nodes provide the accepted response and node resources are occupied, the resource will be insufficient to provide all VNFs due to virtual occupation, as depicted in Fig. 3(a). Assume that MDC obtains an SFCR at time t and transmits it to each domain. The IDC compares the node's resources to the requirements of the VNFs in each domain and returns ξ. Among them, green nodes indicate that the node has enough resources to instantiate the VNF, and the red one indicates that it can't instantiate the VNF. All green nodes will deduct the resources capacity of the corresponding VNF, as a result, when subsequent VNFs enter the system, most of the nodes may give a rejection response, and the acceptance response will be greatly reduced, as shown in Fig. 3(a), which will have a significant impact on the acceptance rate.

Therefore, to avoid this issue, this paper optimizes the response process. When an IDC receives an SFCR, the IDC will first predict the available node resources based on the node's response, for example, if two VNFs have received the accept response ξ = 1, and the resource requirement of these two VNF are a and b respectively, the resource capacity of the node must be large than $a + b$. Then match the nodes with sufficient resources to participate in the current VNF deployment by resource sorting, that is only the matched nodes will be taken into account for the current VNF deployment. According to the simulation results presented in Section V, in the response process, the SFC acceptance rate and end-to-end delay will be improved if the VNF with the largest resource requirement match with the nodes that have the top K resource capacity in each domain, where K is determined by



(a) Nodes response without reasonable matching.

(b) Nodes response with reasonable matching.

Fig. 3. Optimizing the response process by resource matching. After reasonable matching between VNF and response nodes, the virtual resource occupation can be reduced as much as possible, thereby improving the service success rate.

the simulation setup. As shown in Fig. 3(b), green nodes and red nodes represent acceptance and reject nodes, respectively, while blank nodes indicate that they are not compared with incoming VNFs, in this example, $K = 30\%$. Through the optimization mentioned above, the system can prevent the previous VNFs from consuming an excessive amount of node resources, and improve the acceptance rate.

C. Privacy Protection

As we discussed in Section I, the underlying network topology and resource capacity information in each domain will be considered private information and can not be exposed to third parties, therefore, node binary response and resource prediction technologies are used to protect privacy while achieving the cross-domain SFC deployment, as we discussed in previous sections III-A and III-B.

In the whole response process, we can say that only SIRM is uploaded to the upper MDC for DQN policy learning, the IDC predicts the node resource for reasonable matching between VNF and nodes rather than directly perceiving the real resource information, the network topology and node resource information in each domain is strictly stored locally. It is worth mentioning that although the MDC is directly connected to each IDC, and the IDCs upload the binary matrix to the MDC for decision-making, there is no direct information exchange between domains, and security technologies such as blockchain can be applied to prevent MDC from information leakage, it is also difficult for attackers to restore the real network resource situation just based on the binary matrix.

D. Problem Description

Assume that the network topology is represented by D = (N, L), where N represents a set of nodes, and L is a set of links between two nodes. The n-th domain in N is denoted by N_n, and a node in the network is denoted by $N_n^i \in N_n$. The link between the node N_n^i and $N_n^{i'}$ in the same domain is represented by $l_n^{ii'} \in L$, and the link between the node N_n^i and $N_{n'}^{i'}$ in the different domains is denoted as $l_{nn'}^{ii'} \in L$. The VNF processing delay in node N_n^i is represented as d_n^i, and the link delay of $l_n^{ii'}$ and $l_{nn'}^{ii'}$ is represented by $d_n^{ii'}$ and $d_{nn'}^{ii'}$, respectively. Each VNF requires various resources (e.g., memory and CPU cores). In this paper, the computing resources of nodes and delay of links are taken into account for SFC deployment, while we only consider the general computing resources (i.e., CPU cores) [35] which can be extended easily to multiple resource types, the node resource measured by units and represented by R_n^i, the link delay are measured by ms. When SFC is deployed across multiple domains, the VNFs are deployed to various nodes in different domains. Suppose, at time t, the SFC consists of u VNFs entering the system, which is denoted as $v_u(t)$, and needs to deploy by searching for suitable nodes in the physical network. The predicted resources of N_n^i and required resources of $v_u(t)$ are denoted as $R_i(t)$ and $R_v^u(t)$, respectively.

1) Node Binary Response: We assume that the resource information is strictly maintained within the node and is unavailable to both IDC and MDC. After SFCs enter the system, MDC receives the service requests and sends the SFCRs to IDC. Then, IDC directs the nodes in each domain to give the binary response and IDC will predict the node's virtual resources capacity of the nodes to match the node for VNF deployment. The resource $R_v^u(t)$ required by each VNF on the SFCR will be compared with the resource capacity R_n^i of each node N_n^i, and each node only gives the binary response indicates whether it is enough for the u-th VNF deploying:

$$X = \begin{cases} 1, \text{ if } R_n^i \geq R_v^u(t). \\ 0, \text{ otherwise.} \end{cases} \tag{1}$$

2) Node Resource Prediction: Assume that there are total m VNFs satisfy $X = 1$, i.e., can be deployed in the corresponding node, and these m VNFs have the corresponding resources requirement of $R_v^{u(X=1)} = \{R_v^{1(X=1)}, R_v^{2(X=1)}, ..., R_v^{m(X=1)}\}$. Therefore, the resource capacity of the node must be greater than the total resources of all VNFs that can be instantiated, and the predicted resources of node N_n^i can be expressed as:

$$R_i(t) \geq \sum_{u=1}^{m} R_v^{u(X=1)}(t). \tag{2}$$

Since we only consider node resource values that are compatible with VNFs deployment, the final predicted value can be taken directly as the minimum predicted value of $R_i(t)$. This ensures that the predicted value is equal to or less than the actual value of node N_n^i, thus ensuring that all nodes which receive an accepted reply can be deployed to node N_n^i successfully. Therefore, the predicted value of the node resource capacity is:

$$R_i(t) = \sum_{u=1}^{m} R_v^{u(X=1)}(t). \tag{3}$$

3) Reasonably matching VNF with Nodes: To increase the deployment acceptance rate, VNF and nodes need to be sorted reasonably to achieve the best match between VNF and nodes, and those nodes that do not match with the VNF don't need to give a reply, avoiding the virtual occupation of network nodes as much as possible. Specifically, the VNFs in an SFC are sorted by resource requirement before comparison, with the VNFs requiring the most resources being compared first. Similarly, all nodes are sorted according to the predicted resource capacity, and the top K of nodes are chosen to provide the VNF accept or reject response to avoid the excessive resource virtual occupy, as discussed in Section III-B. This ensures the matching of VNFs that require great resource requirements with the resource-sufficient node, hence maximizing node resource usage. Finally, when a node gives an acceptable signal, it will deduct the corresponding VNF resources, and other nodes will be release. Therefore, the preceding process prevents all nodes from participating in resource reduction and reduces the risk of SFC deployment failure.

As shown in Fig. 4, multi-domain networks consist of three domains, with four switching nodes and three servers in domain 1, while domains 2 and 3 consist of a single node. Assume that the s-th SFCR enters the system at time t and the required resources for VNFs are assigned in decreasing order as follows:

$$R_v^2(t) \geq R_v^3(t) \geq R_v^1(t) \geq R_v^4(t). \tag{4}$$

Assuming that IDC correctly predicted the virtual resource situation and three nodes in domain 1 are sorted in descending according to their predicted resource capacity as follows:

$$R_2(t) \geq R_1(t) \geq R_3(t). \tag{5}$$

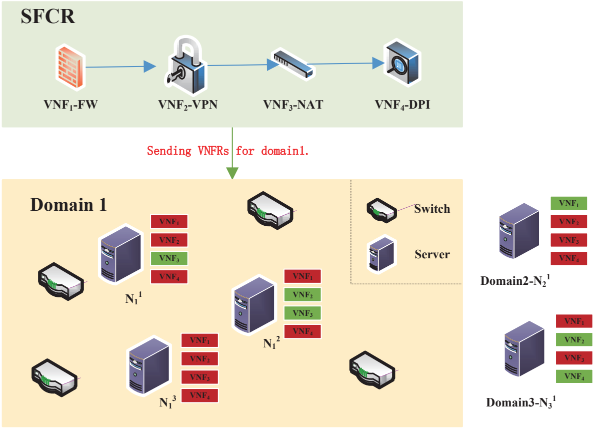


Fig. 4. VNFRs response status of nodes in domains.

TABLE III
VNFRs Response on Each Domain Nodes

	Domain1			Domain2	Domain3
Node	N_1^1	N_1^2	N_1^3	N_2^1	N_3^1
VNF1	0	0	0	1	0
VNF2	0	1	0	0	1
VNF3	1	1	0	0	0
VNF4	0	0	0	0	1

Then, top K of nodes are first compared with VNF_2 which requires the largest resource requirement. Suppose that in the previous prediction step, N_1^3 has insufficient resource capacity to instantiate any VNFs. Therefore, we exclude N_1^3, selecting N_1^2 to compare with VNF_2. If $R_2(t) \geq R_v^2(t)$, the resource capacity of N_1^2 can ensure the deployment of VNF_2. In this case, the node resources are deducted:

$$R_2(t) = R_2(t) - R_v^2(t). \tag{6}$$

An acceptance response is given after deduction. Similarly, node N_3^1 with the second largest predicted resource capacity will do the same operation. The nodes are reordered after deducting resources, and the top K of nodes are compared with VNF_3 again, then the corresponding resource value is deducted. After comparison and resource deduction, each VNF of s-th SFCR can obtain two different responses (i.e., ξ = 1, or ξ = 0), as shown in Table III. Finally, the SIRM will feed into DQN neural network for SFC deployment action learning.

4) SFCs Deployment: Each IDC returns the binary response of nodes to MDC to form the SIRM, and these local matrices comprise the global view. MDC utilizes this global view to determine the deployment strategy, which identifies specific nodes in each domain to provide resources for VNFs deploying and connecting links between nodes. When the deployment is completed, MDC receives a signal indicating whether it was successful or failed. Using the success signal, MDC can evaluate the performance of the deployment. On the contrary, MDC can improve the deployment process by failed signal. At the same time, when new SFCs arrive, each node will execute a new round of resource prediction and binary response. Then, each domain uploads a new SIRM to MDC for the next round of deployment. During the deployment of SFCs, the SFCRs received by MDC are dynamically entered into the system. SIRM will be continuously updated, and the updated SIRM will be used to deploy the corresponding SFCRs. Then, the IDC utilizes the deployment strategy learned by MDC to deploy the VNFs and link.

E. Joint Resource Utilization and Delay Optimization

Considering the single optimization goal (e.g. end-to-end delay) may cause resource and energy waste, which is not appropriate for resource and energy-limited IIoT. Therefore, this paper jointly optimizes service delay and resource utilization. Reducing overall service delay to the maximum extent by making full use of the existing network resource.

1) Resource utilization: We introduced two variables $P_i(t)$ and $P_N(t)$ to describe the resource utilization of each node and the entire network, respectively:

$$P_i(t) = \frac{\sum R_v^u(t)}{R_i}, \tag{7}$$

$$P_N(t) = \frac{\sum P_i(t)}{N}, \tag{8}$$

where $R_v^u(t)$ denotes the resource requirement of VNF deployed in node N_n^i. The resource utilization of the node $P_i(t)$ represents the resource usage of this node at time t. While the resource utilization of the whole network $P_N(t)$ refers to the average resource usage of network nodes and is calculated by dividing the total resource utilization of all nodes by the node number.

Let $X_{v_u}^{N_n^i}$ be a binary variable, indicating whether the u-th VNF in the SFC has been deployed to N_n^i:

$$X_{v_u}^{N_n^i} = \begin{cases} 1, \text{ if the VNF of SFC is placed on } N_n^i \ . \\ \qquad\qquad 0, \text{ otherwise.} \end{cases} \tag{9}$$

Similarly, $Y_l^{i,j}$ is a binary variable, indicating whether the virtual link is mapped to the link between N_n^i and N_n^j:

$$Y_l^{i,j} = \begin{cases} 1, \text{ if virtual link } l \text{ is mapped between } N_n^i \text{ and } N_n^j \ . \\ \qquad\qquad 0, \text{ otherwise.} \end{cases} \tag{10}$$

If both binary variables $X_{v_u}^{N_n^i}$ and $Y_l^{i,j}$ are 1, the SFC is considered deployed successfully.

2) End-to-end delay: The SFC end-to-end delay depends on various factors, including computing, networking, and storage resources, as well as the traffic mode of the links. Generally, the end-to-end delay consists of processing, transmission, and propagation delays [36]:

$$D_{end}^s(t) = d_{proc}(s_t) + d_{tran}(s_t) + d_{prop}(s_t), \forall s \in S. \tag{11}$$

Propagation delay (d_{prop}): We use real-world topology in this paper. d_{prop} predominantly depends on the physical distance and the transmission speed of a signal in a transmission medium:

$$d_{prop}(s_t) = \frac{dis}{c}, \tag{12}$$

where dis represents the distance between two nodes, and c represents the speed of signal propagation in the link (c mainly depends on the transmission medium of the link).

Transmission delay (d_{tran}): d_{tran} depends on the period of sending the data packet from the sender to the receiver and is related to the packet length and data transmission rate:

$$d_{tran}(s_t) = \frac{L_{packet}}{f}, \tag{13}$$

where L_{packet} represents the length of the processed packet, f indicates the packet transmission rate.

Processing delay (d_{proc}): d_{proc} typically occurs when a host or a system needs to process a received packet. The packet processing includes, but is not limited to, header analysis, data extraction, error checking, and finding the appropriate route. As the data rate increases, the processor's load becomes heavier, causing significant processing delays. The processing delay of the u-th VNF in the SFC at time t can be expressed as:

$$d_u^i = \frac{X_{v_u}^{N_n^i}(1 + \vartheta_t(N_n^i))}{1 - \vartheta_t(N_n^i)} t_{proc}, \tag{14}$$

where $\vartheta_t(N_n^i) = \frac{O_u(t)}{O_i(t)}$ represents load rate of node N_n^i at time t, $O_i(t)$ indicates the available load of each node, $O_u(t)$ denotes the load requirement of VNFs in node N_n^i, t_{proc} represents the processing time of each packet. Therefore, the processing delay of the SFC s at time t is expressed as:

$$d_{proc}(s_t) = \sum_{u=1}^{|V|} d_u^i, \tag{15}$$

where $|V|$ represents the total number of VNFs in SFC $s_s(t)$.

3) Optimization objective: The optimization objective of our SFCs deployment problem is defined as:

$$\mathcal{O} = \min_{\{X_{v_u}^{N_n^i}, Y_l^{i,j}\}} \frac{\sum_{s=1}^{|S|} D_{end}^s(t)}{|S| P_N(t)}, \forall s \in S, \tag{16}$$

where $X_{v_u}^{N_n^i}$ and $Y_l^{i,j}$ are two optimization variables learned by the DRL-based algorithm, as described in the next section, and $|S|$ represents the total number of SFCs. The objective can be achieved by minimizing service end-to-end delay and improving the utilization of network resources.

In addition, the following network conditions must be restricted to ensure a successful SFCs deployment. The resource demand of all VNFs instantiated in the node should not exceed the total resource capacity, and the computing load of the node should not exceed the maximum load rate, as defined in (17) and (18), where $R_v^u(t)$ denotes the resource requirement of VNFs deployed in node N_n^i, while $\vartheta_t(N_n^i)$ represents the load rate of node N_n^i at time t. The constraint of (19) ensures each VNF can be assigned to only one node in the network, while (20) ensures the routing path between VNFs can be assigned to a single physical link.

$$R_i(t) \geq \sum R_v^u(t) \cdot X_{v_u}^{N_n^i}, \tag{17}$$

$$\vartheta_t(N_n^i) \cdot X_{v_u}^{N_n^i} \leq 1, \tag{18}$$

$$\sum X_{v_u}^{N_n^i} \leq 1, \forall u \in V, \tag{19}$$

$$\sum Y_l^{i,j} \leq 1, \forall l \in L. \tag{20}$$

IV. SFCs Deployment Strategy Based on DQN

Traditional methods cannot learn the deployment strategy well just relying on the SIRM, while the DQN algorithm has obvious advantages in matrix input learning. Therefore, this section presents the DQN to assist MDC in determining the SFCs deployment strategy based on the SIRM.

A. Acceptance rate optimization algorithm

As stated in Section III-B, binary responses without resource prediction and reasonably matching will result in a lower SFCs acceptance rate due to the virtual occupation of node resources. The VNFs that request resources later will receive a rejection response because the node resources are "insufficient". Therefore, by predicting each node's resources and carrying out a suitable sorting and node matching during the response, the SFC acceptance rate can be greatly enhanced, as shown in Fig. 3.

We propose an acceptance rate optimization algorithm (AROA), as shown in Algorithm 1, AROA takes a group of VNFs connected in sequence as input. Let the VNFs set $V = (V_1, V_2, ..., V_u)$, MDC sends the set to each IDCs, and IDC distributes V to all intra-domain nodes (line 1). Assume that the node set in the current system is N, and the initial predicted value $R_i(t)$ is initialized to 0 (line 2). Then, the resource R_n^i of each node N_n^i in the node-set N is compared with the requested resource $R_v^u(t)$ of VNF one by one, select the VNFs set $V' = (V_1', V_2', ..., V_{u'}')$ that can be deployed in the node and add all the required resources of VNFs in set V' to get node prediction resources (lines 3-11). Then, the set V'' is obtained by sorting VNFs resources in descending order (line 12). The nodes are also sorted according to the predicted value $R_i(t)$ in descending order, and the top K of the nodes are selected to form the new set N' (line 13), where K is determined by the corresponding simulation setup. Comparing each VNF with the new node set N' in order. If the resource R_n^i of node N_n^i is larger than the requested resource $R_v^u(t)$ in set V'', return an acceptance answer, i.e., ξ = 1; otherwise, ξ = 0 (lines 14-23).

Algorithm 1 SFCs Acceptance Rate Optimization Algorithm

```
Input: VNFs set $V = (V_1, V_2, ..., V_u)$
Output: SIRM
 1: MDC distributes VNFs set $V = (V_1, V_2, ..., V_u)$ to all IDCs;
 2: Initialize the predicted node value $R_i(t) = 0$;
 3: for $N_n^i \in N$ do
 4:     for $V_i \in V$ do
 5:         if $R_n^i \geq R_v^u(t)$ then
 6:             Node predicted value is $R_i(t) = R_i(t) + R_v^u(t)$;
 7:         else
 8:             The predicted value of node $R_i(t)$ remains unchanged;
 9:         end if
10:     end for
11: end for
12: Sort the VNFs resources in descending to obtain the new set $V''$;
13: Sort nodes in descending according to the predicted value $R_i(t)$, and the top $K$ of nodes is selected to form the set $N'$;
14: for $N_n^i \in N$ do
15:     for $V_i \in V''$ do
16:         Compare each VNF with the selected nodes of $N'$ in order;
17:         if $R_n^i \geq R_v^u(t)$ then
18:             return $\xi$ = 1;
19:         else
20:             return $\xi$ = 0.
21:         end if
22:     end for
23: end for
```

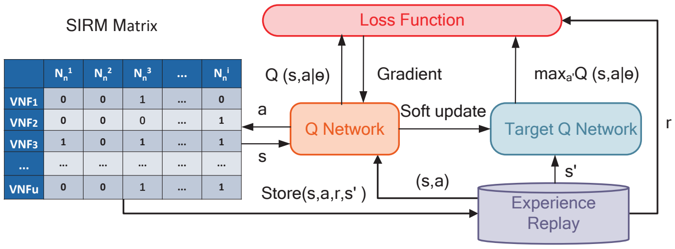
B. Cross-domain SFCs Deployment based on DQN

This section will describe the learning process of the DQN-based SFCs deployment strategy. The entire system comprises two fundamental components: the observed SIRM and MDC. These components combine with DQN to determine the deployment decision for cross-domain SFCs.

1) Markov Decision Process Modeling: During the deployment process, the deployment of SFCs in the current state is dependent on the previous state, or the response at the previous instant, indicating that the deployment of SFCs has the Markov property. Due to the dynamic feature of SFCRs and the complexity of the network, traditional approaches hard to simultaneously obtain promising performance while protecting the privacy, which is an NP-hard problem [37]. This problem can be transformed into a Markov decision process (MDP), and solved by DRL-based algorithm. MDC constructs the global view provided by the SIRM return from each domain, which is considered the system state ($\mathcal{S}(t) = s$). DQN takes SIRM as input and learns action ($\mathcal{A}(t) = a$) which determines the VNF placement and traffic scheduling across multiple domains. Considering a standard DQN configuration, MDC continuously learns by interacting with the environment and improves its performance via the rewards function. Then, the system goes to the subsequent state $\mathcal{S}(t+1)$ and obtained the long-term expected reward $\mathcal{R}(t)$. which are defined as follows:

State space $\mathcal{S}(t)$: Let $\mathcal{S}(t)$ denotes the network state at time t. $\mathcal{S}(t)$ = $\{W\}$, $W = \{W^1, W^2, ..., W^n\}, n \in N$, is a binary matrix, as shown in Algorithm 1, each IDC obtains SIRM (W^n) from its domain by comparing the resources of nodes and VNFs. MDC observes W^n of each domain and combines it to form a global view (W).

Action space $\mathcal{A}(t)$: Let $\mathcal{A}(t)$ denotes the action at time t, $\mathcal{A}(t)$ = $\left\{X_{v_u}^{N_n^i}, Y_l^{i,j}\right\}$, where, $X_{v_u}^{N_n^i}$ and $Y_l^{i,j}$ are both binary variables, and represent the SFC embedding and traffic scheduling strategy, respectively, where $X_{v_u}^{N_n^i} = 1$ if VNF of SFC is placed on N_N^i, otherwise it's equals to 0, in the same way, $Y_l^{i,j} = 1$ if the virtual link is mapped between N_n^i and N_n^j, otherwise it's equals to 0. The service is considered as successful when traffic passes through all of the VNFs


SIRM Matrix

	N_n^1	N_n^2	N_n^3	...	N_n^i
VNF1	0	0	1	...	0
VNF2	0	0	0	...	1
VNF3	1	0	1	...	1
...	...	...	...	...	...
VNFu	0	0	1	...	1

Fig. 5. SFCs deployment process based on DQN.

in an SFC. MDC continually learns by interacting with the environment and improves its performance through a reward function.

Reward function $\mathcal{R}(t)$: This paper aims jointly optimize the resource utilization and the service end-to-end delay of the tasks. However, these two optimization objectives have different units and cannot be accumulated directly, therefore, we first need to normalize these two objectives into [35]:

$$o_{P_N(t)} = max\left\{-1, \frac{P_N(t)}{P_{max}} + 1\right\} \in [-1, 1], \quad (21)$$

$$o_{D^s_{end}(t)} = max\left\{-1, \frac{-D^s_{end}(t)}{D_{max}} + 1\right\} \in [-1, 1]. \quad (22)$$

In our model, a higher system reward can be obtained by high resource utilization and low service end-to-end delay. To guarantee that VNFs can be successfully deployed on nodes, this study added parameters and reward discounts for selecting various nodes. It took the inverse function of resource occupancy rate and end-to-end delay as the reward after taking action. The immediate reward $r(t)$ is defined as:

$$r(t) = r(\delta|s, a) + \alpha \cdot [o_{P_N(t)}] + \beta \cdot [o_{D^s_{end}(t)}] \cdot r(\rho|s, a), \quad (23)$$

where $0 \leq \alpha, \beta \leq 1$ represents the reward discount and $\alpha + \beta = 1$. $r(\delta|s, a)$ and $r(\rho|s, a)$ represent the immediate reward obtained after action $\mathcal{A}(t)$ in state $\mathcal{S}(t)$. $P_N(t)$ and $D^s_{end}(t)$ are the resource utilization and SFC end-to-end delay after deployment. $r(\delta|s, a)$ indicates the response status of the node selected by MDC. If $\xi = 1$, give a larger reward; $\xi = 0$, give a smaller reward. $r(\rho|s, a)$ indicates whether the nodes selected by MDC are the same. If a node is selected repeatedly, a small reward will be given; otherwise, a large reward will be given. $r(\rho|s, a)$ avoids that the system continuously selects the same node to reduce the end-to-end delay, resulting in insufficient node resources and eventual deployment failure. The DRL method aims to find an optimal policy to maximize the cumulative reward $\mathcal{R}(t) = \sum_{t=1}^{T} r(t)$ while following the policy.

2) SFC Cross-domain Deployment Algorithm based on DQN: This section will specifically introduce the CDDA based on DQN for SFC, as shown in Fig. 5.

Both Q-learning and DQN [38] are typical value-based RL methods. The approaches use the value function to learn the optimal strategy through interaction with the environment. The action value function of Q-learning can be presented as:

$$Q^\pi(s, a) = E_s[r(s, a) + \gamma E_{a'\pi}[Q^*(s', a')]], \quad (24)$$

where $r(s, a)$ represents the obtained immediate reward after the state s takes action a. γ is the discount reward that is used to calculate the cumulative reward from the state to the end, and $Q^*(s', a')$ is the optimal value action function.

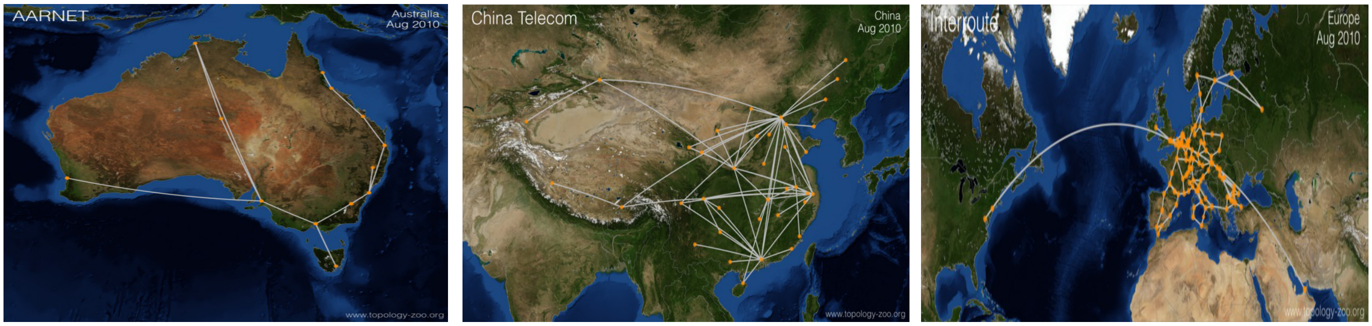
DQN integrates the Q-learning algorithm with a deep neural network, which introduces the training target network and experience replay. The learning process is shown in Algorithm 2. First, MDC observes SIRM (W^n) of each domain and combines it as input to form a global view (W) (line 1). Initialize the predicted value $R_i(t)$ of node, and initialize the Q-network parameters (line 2), then, MDC selects nodes for VNFs deployment with the input SIRM based on current policy and outputs the action $\mathcal{A}(t)$ (line 4). After the action is executed, MDC will receive an immediate reward $r(t)$. Then, the current state s will be transited to the next state s' (lines 5-9); and the experience $E(s, a, r, s')$ will be stored in the replay buffer (line 10). When the replay buffer has enough transition samples, a mini-batch of data can be selected from the experience pool for network training (line 12). DQN adopts a dual deep neural network structure (i.e., target network and evaluate network), and updates the network parameters by minimizing the loss function (lines 13-14), as shown in:

$$\begin{cases} y = r(t) + \gamma \cdot max_{a'}\bar{Q}^\pi(s', a'), \\ L(\theta) = E_{(s,a,r,s')}[(Q^*(s, a|\theta) - y)^2], \end{cases} \quad (25)$$

where $\bar{Q}$ represents the target network, and θ is the parameter of the evaluation network. The target network $\bar{Q}$ copies the parameters from the evaluated network in a fixed-steps. This parameter updating scheme can break the learning experience's correlation in order to stabilize the training process.

Algorithm 2 DQN-based CDDA of SFCs

```
Input: SIRM $(W^n)$
Output: Deployment strategies
1: MDC observes global SIRM($W$).
2: Initialize the predicted value of node $R_i(t)$, and initialize
   the network parameters.
3: for $N_n^i \in N$ do
4:     Select nodes and links for SFCs deployment;
5:     Return immediate reward according to the response of
   selected nodes: $r(\delta|s, a)$;
6:     Return immediate reward according to the coincidence
   of selection nodes: $r(\rho|s, a)$;
7:     Get rewards:
8: $r(t) = r(\delta|s, a) + \alpha \cdot [o_{P_N(t)}] + \beta \cdot [o_{D^s_{end}(t)}] \cdot r(\rho|s, a)$;
9:     Update status: $E(s, a, r, s')$;
10:    Store $E(s, a, r, s')$ into experience pool;
11:    if MDC has enough transition samples then
12:        Select a small batch of data from the experience
   pool for training;
13:        Minimize loss function:
14:        $\begin{cases} y = r(t) + \gamma \cdot max_{a'}\bar{Q}^\pi(s', a'), \\ L(\theta) = E_{(s,a,r,s')}[(Q^*(s, a|\theta) - y)^2]. \end{cases}$;
15:    else
16:        Return to line 2;
17:    end if
18: end for
```



(a) topology scale = 19 nodes (b) topology scale = 38 nodes (c) topology scale = 110 nodes

Fig. 6. Real-world topology selected from the Internet Topology Zoo.

C. Algorithm Complexity

Assume that the topology scale of the network is N and the number of VNFs in SFCs is M. In Algorithm 1, the initialization process (lines 1-2) and the resource sorting process (lines 12-13) are executed only once. The first for loop (lines 3-11) executes $2MN + N$ operations, while the second for loop (lines 14-23) requires $3MN + N$ operations. Therefore, Algorithm 1 has a total complexity of $5MN + 2N + 4$. In our actual simulation, the network topology scale is set to 19, 38, and 110 nodes, and is divided into three domains. The total number of VNFs considered in our simulation ranges from 3-6 VNFs. The total algorithm complexity can be represented as $O(zMN)$, where z is an integer, implying that the first proposed algorithm has an acceptable complexity. In the same way, Algorithm 2 requires $8N + 2$ operations. The main algorithm complexity of Algorithm 2 is the training of the DQN neural network, which takes between 26-45 seconds to converge on 2000 episodes, the training time is also acceptable.

V. Simulation Results and Discussions

This section describes the parameter settings and analyzes the simulation results in order to evaluate the advantages and disadvantages of the proposed algorithm.

A. Simulation Setup

1) Node and Link Setting: Due to the lack of a topology dataset designed for the cross-domain deployment of IIoT, we still use existing distributed topologies for custom design, the topology scale is configured based on the conventional number of equipment in IIoT simulation [39]. We select three real-world topologies from the Internet Topology Zoo [40], namely Aarnet, China Telecom, and Interroute, which are comprised of 19, 38, and 110 nodes, respectively, as shown in Fig. 6. In each domain, each topology is divided into three network domains based on the actual topology distribution, as shown in Table IV. The node resources capacities are set randomly between 7-19 units. The link delay is configured as shown in Table V, according to our previous work [41], where the diagonal value is configured as the link delay in each domain, and the rest are link delays between two domains. It can be seen that the link delay of the inter-domain link is larger than that of the intra-domain, which is consistent with the actual situation. The node resource and link delay is randomly set according to the defined range and fixed during the comparison process to ensure fairness. But when starting a new round of experiments, these settings will reset correspondingly.

TABLE IV
Domain Classification of Topologies

Internet Topology	Domain 1	Domain 2	Domain 3
Aarnet	Node 0-6	Node 7-13	Node 14-18
China Telecom	Node 0-15	Node 16-29	Node 30-37
Interroute	Node 0-25	Node 26-79	Node 80-109

TABLE V
Link Delay Setting

Domain	Domain 1	Domain 2	Domain 3
Domain 1	5-10ms	11-20ms	21-30ms
Domain 2	11-20ms	5-10ms	11-20ms
Domain 3	21-30ms	11-20ms	5-10ms

2) SFC and VNF Setting: The SFC requests follow the realistic traffic trace pattern that arrives following a Markov-modulated Poisson process (MMPP), with two states mean inter-arrival time 12 and 8 (50% higher rate) every 100-time steps with 5% probability, which has been widely used for modeling SFC requests in recent works [35]. The simulation runs in 100,000 time steps, and 3,000 SFCs enter the system according to the defined traffic pattern. Each SFC consists of 3-6 VNFs, the processing delay of each VNF is set between 5-10 ms. In this paper, we employ eight commonly used VNFs, including DPI (Deep Packet Inspection), NAT (Network Address Translation), FW (Firewalls), TM (Traffic Management), WOC (WLAN over CATV), IPS (Intrusion Prevention System), and IDS (Intrusion Detection System). A set of VNFs are connected sequentially to comprise the SFC based on user requirements, and the resource requirement of each VNF is randomly assigned a value between 2-8 units.

The simulation runs on server with 12GB memory, Intel(R) Core (TM) I5-9500, CPU @ 3 GHz. We use python 3.7 development environment for programming and implementation.

B. Baseline Algorithms

As benchmarks, we employ three comparing algorithms, including privacy protection and non-privacy protection methods, which are introduced as follows:

1) Exposing privacy shortest path (EP-SP) algorithm: EP-SP algorithm directly finds the shortest path with the shortest delay and high resource utilization in a completely transparent network when SFC enters the system. The EP-SP algorithm can directly identify the shortest path with the theoretically minimum delay based on the exposed resource information. In an ideal situation, the delay performance of our proposed method should be comparable to that of the EP-SP algorithm while protecting privacy.

2) Exposing privacy deep Q-network (EP-DQN) algorithm: Similar to EP-SP algorithm, EP-DQN algorithm also learns the SFCs deployment strategy in a fully transparent network. Our proposed method and EP-DQN algorithm use the same DQN algorithm to determine the deployment strategy. However, using resource information as input to discover the deployment strategy will compromise privacy. Therefore, EP-DQN algorithm verifies whether the proposed scheme has advantages in the dynamic cross-domain deployment of SFC by using SIRM as the input of the DQN network.

3) Column generation algorithm based SFCs deployment (CG-BSFC) algorithm [42]: CG-BSFC algorithm is a privacy-preserving algorithm that hides the topology and resource information of each domain based on the column generation method. In addition, the CG-BSFC algorithm employs a conventional heuristic algorithm to determine the deployment decision based on the pre-allocated scheme. After receiving SFCR, the system directly provides available schemes according to the SFC length and SFC source node to avoid exposing the resource information. CG-BSFC algorithm is used to compare the privacy protection of SFCs deployment across domains with the proposed scheme.

C. Simulation Result

The simulation results presented in this study include the learning curve of various DQN algorithm parameter settings, the performance improvement by AROA, and a comparison of the deployment performance of four algorithms.

1) Learning Curve: We first verify the learning curve of different learning rates (0.01, 0.005, 0.001, 0.0005, and 0.0001) under an increasing topology scale. To obtain the best convergence performance of the DQN-based cross-domain deployment algorithm, we keep the original discount factor of 0.95, by default, $r(\delta|s,a)$ and $r(\rho|s,a)$ in reward function were set with the range of [-10, 10] and [-1, 1], respectively. As shown by the cumulative reward in Fig. 7, when the topology is constantly increasing, the learning rate of 0.001 can obtain the maximum cumulative reward in different network topologies. Indicates that it can find a better solution in minimizing end-to-end delay and improving resource utilization. In contrast, the learning rate of 0.0001 is less efficient, and 0.005, 0.001, and 0.0005 are also not superior.

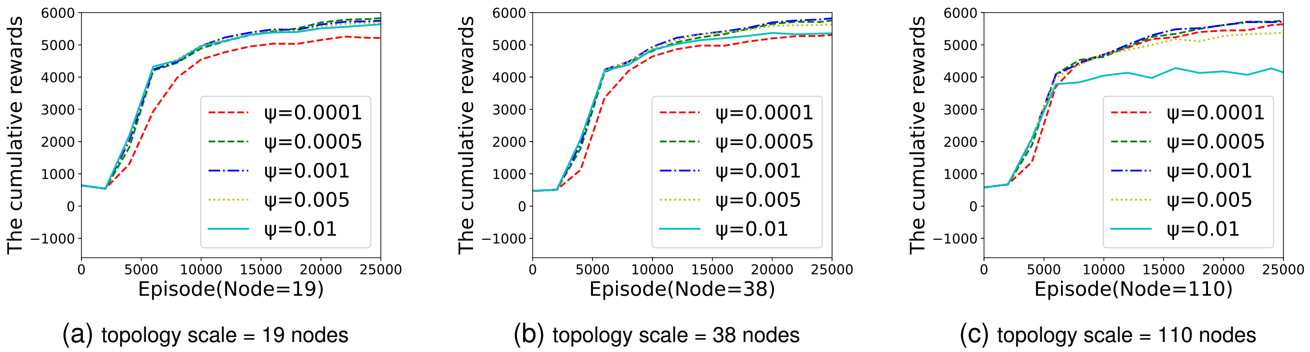
As shown in Fig. 7, as the topology scale increases, the cumulative reward of learning rates = 0.01, 0.005, and 0.0005 decreases significantly, whereas the learning rate of 0.001 remains relatively stable. In the case of a small number of samples, the learning rate = 0.01, and 0.005 may achieve better results. However, as the topology scales increase, the reward decreases, and it becomes more difficult to achieve the global optimal compared to learning rate = 0.001. While the learning rate = 0.0005 is too small, which leads to slow convergence, and more learning steps are needed. In short, the 0.001 was selected as the learning rate of the DQN algorithm because it provided the best convergence performance.

This paper aims to jointly optimize network resource utilization and end-to-end delay, and control these two objectives by α and β. We verify the best trad-off of these two objectives as shown in Fig. 8. Three cases were verified, i.e., $\alpha = 0.2$, $\beta = 0.8$, $\alpha = 0.5$, $\beta = 0.5$, and $\alpha = 0.8$, $\beta = 0.2$. While the parameters of $\alpha = 0.2$ and $\beta = 0.8$ obtain the highest cumulative rewards and obtain the best convergence performance in three increasing network topologies. Therefore, we choose $\alpha = 0.2$ and $\beta = 0.8$ as the coefficients for subsequent simulation.

2) Performance Improvement by AROA: As described in Section III, the proposed AROA optimizes the SIRM to improve the SFCs deployment acceptance rate. The proportion (i.e., K) of nodes responsible for binary response will significantly affect the acceptance rate of services, as shown in Fig. 9. To determine the best proportion, K was set between 10% to 100%. Fig. 9(a) demonstrates that a smaller proportion of response nodes leads to improved performance. When the responding nodes are more than 60% of all available nodes, the SFC acceptance rate decline sharply and only reaches about 36-60% acceptance rate in different topologies, due to the virtual resource occupation. In contrast, when the proportion of response nodes is between 10-30%, the SFC acceptance rate can exceed 90% in various topologies.

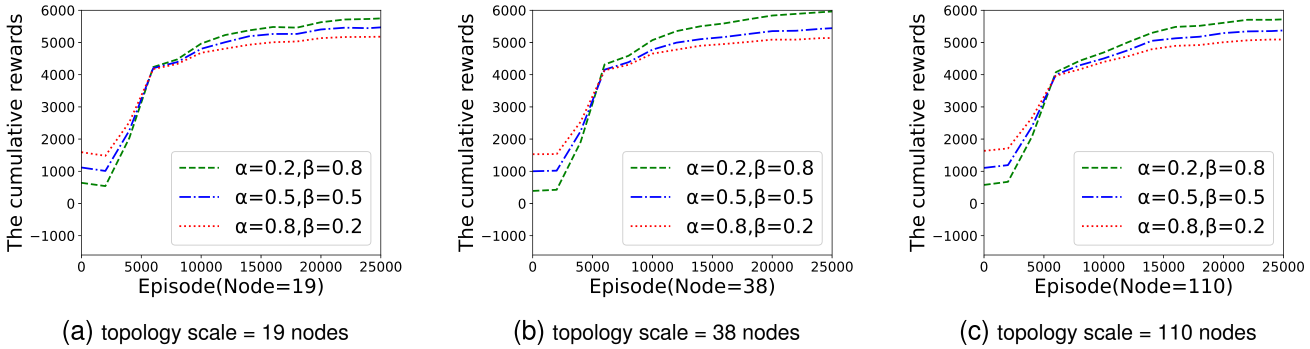
As shown in Fig. 9(b), further examination of the end-to-end delay of SFCs with K ranging from 10% to 30% reveals that K = 30% yields the best delay performance among the different topologies. In addition, we found that the end-to-end delay of SFCs decreases slightly as the topology scale increases. This is because the SFCs will be preferentially deployed in the same domain. That is why we introduce the instants reward $r(\rho|s,a)$ in the reward function (23), $r(\rho|s,a)$ avoids the system continuously selects the same node to reduce the delay, resulting in insufficient node resources and eventual deployment failure.

3) SFC Acceptance Rate: PPDM protects privacy during the cross-domain deployment of SFCs by using the SIRM as input for DQN rather than resource information. We further improve the service acceptance rate through the AROA algorithm. The SFC acceptance rate comparison of four algorithms is shown in Fig. 10. The SFC acceptance rate of the EP-SP algorithm exceeds 90% in various topologies and can approach 100% when sufficient resources are available. This is because the EP-SP algorithm can immediately retrieve the resource situation and select nodes in the topology for each VNF to deploy. Therefore, as long as all nodes have enough resources, SFC can be effectively deployed and maintained a high SFC acceptance rate. In contrast, the SFC acceptance rate of PPDM can approach or even surpass that of EP-SP algorithm in the same simulation environment, reaching over 90%. EP-DQN algorithm and EP-SP algorithm, learn SFC deployment strategies in a completely transparent network. However, EP-DQN



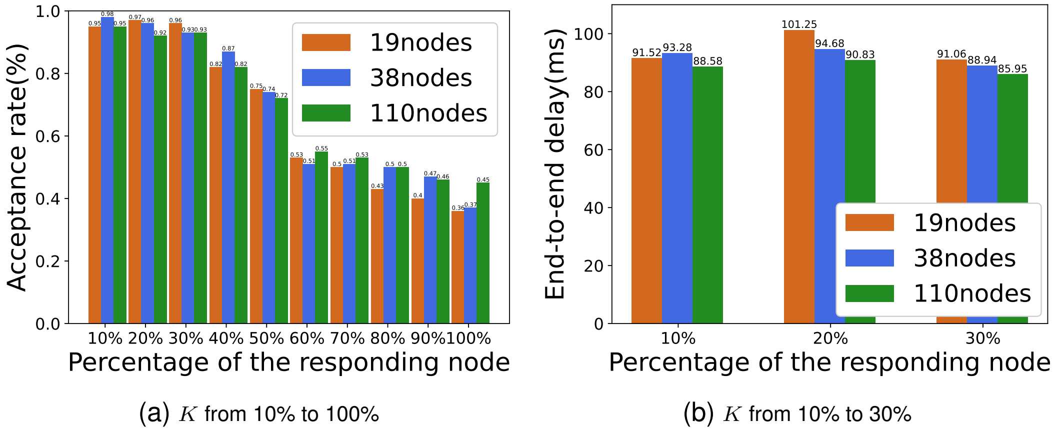
(a) topology scale = 19 nodes (b) topology scale = 38 nodes (c) topology scale = 110 nodes

Fig. 7. Learning curves of different learning rates.



(a) topology scale = 19 nodes (b) topology scale = 38 nodes (c) topology scale = 110 nodes

Fig. 8. Resources utilization and delay optimization trad-off.



(a) K from 10% to 100% (b) K from 10% to 30%

Fig. 9. Comparison of SFC acceptance rate and end-to-end delay with different K.

still cannot achieve a higher SFC acceptance rate even if it can find the best strategy in a fully exposed topology, due to the lack of acceptance rate optimization process (i.e., Algorithm 1). Consequently, its acceptance rate and can only reach about 53-78%, which is significantly lower than EP-SP algorithm and PPDM. As described in Section V-B, the CG-BSFC algorithm seeks the best strategy for ultimate deployment in order to maintain privacy. The SFC acceptance rate cannot achieve a high result since this algorithm selects the best strategy among permutations and combinations, making it impossible to know if the chosen node will have sufficient resources for VNF at the next instant. The simulation results demonstrate that the SFC acceptance rate of the CG-BSFC algorithm can exceed 70%. In addition, as the topology scale increases, the CG-BSFC algorithm must generate more schemes in advance, which is inapplicable when facing a large number of continuous tasks. While the DQN-based method solves this issue essentially, it is more suited for large-scale state and action space tasks.

4) SFC End-to-end Delay: Then, we compare the SFC end-to-end delay of four algorithms, as shown in Fig. 11. EP-SP algorithm can guarantee the minimum end-to-end delay since it is an idealized algorithm. The PPDM has a 15-19ms greater end-to-end delay than the EP-SP algorithm, which is considered the near-optimal result. However, compared with the CG-BSFC algorithm, the end-to-end delay of PPDM is reduced by 7-15ms, and the delay decreases with the increase of the topology scale. Moreover, the EP-DQN algorithm, which also utilizes the DQN method, achieves the same delay level as PPDM. In other words, PPDM utilizes resource prediction and binary response to conceal network resource information

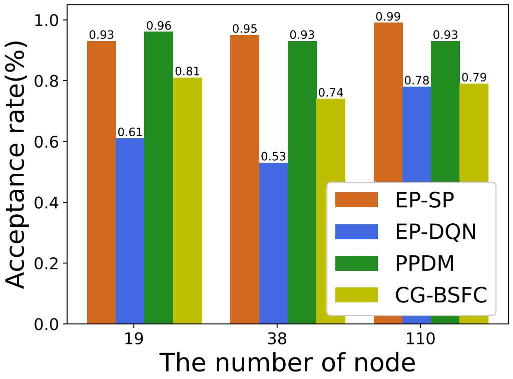


Fig. 10. SFCs acceptance rate in different topologies.

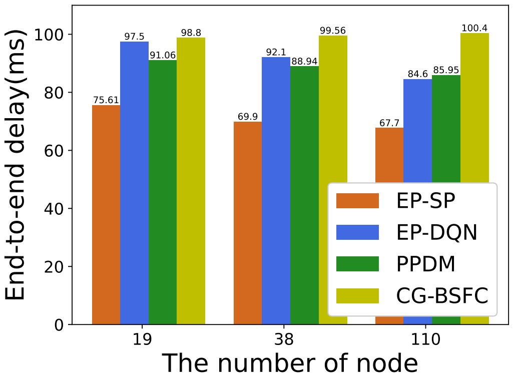


Fig. 11. SFCs End-to-end delay with different topologies.

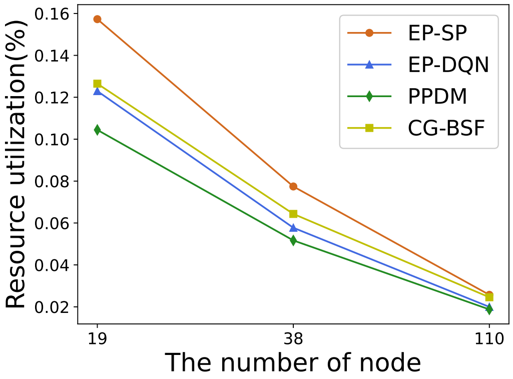


Fig. 12. SFCs resource utilization with different topologies.

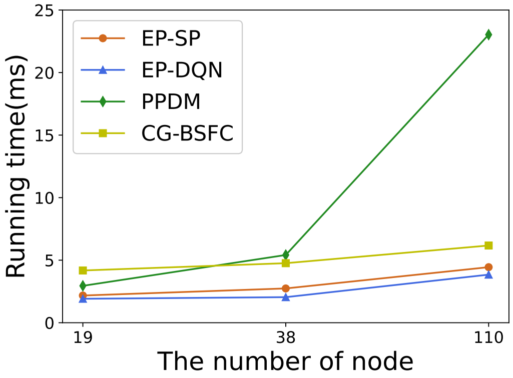


Fig. 13. Running time comparison.

without significantly impacting the end-to-end delay of the SFC. The end-to-end delay of PPDM is the same as that of the EP-DQN algorithm, meanwhile, protecting the resource privacy information. Although the global view observed by MDC of EP-DQN algorithm and DQN-based PPDM is not the same (i.e., transparent resources information and global SIRM). However, in the learning process, the objective of both approaches is to find a solution with lower delay and higher resource utilization, i.e., higher cumulative rewards. Therefore, even if the state of the input neural network is different, the delay of the two algorithms is similar.

5) Resource utilization: In this paper, the average resource utilization of the four deployed algorithms is also compared, as shown in Fig. 12. The EP-SP algorithm tends to select the nodes with minimum link delay and higher resource utilization for the deployment of SFC in a complete view of the resource and topology that exposes privacy. This makes it an idealized algorithm that exhibits excellent advantages in resource utilization. Similarly, compared to PPDM, the EP-DQN algorithm can also accurately find the node to match VNF's required resources by observing the complete view of the network, hence guaranteeing more efficient resource utilization than PPDM. For the CG-BSFC algorithm, a pre-allocation plan is made before VNF selects nodes. The overall controller also tends to select nodes with higher resource utilization when generating the pre-allocation plan. While PPDM tends to find nodes with larger resources to ensure the acceptance rate of deployment. Therefore, it offers no significant advantage in resource utilization. However, as the topology scale increases, the resource utilization of the four algorithms gradually approaches. In other words, for a large-scale network environment like IIoT, the resource utilization of the four algorithms will be close. Therefore, this utilization of resources is also acceptable for PPDM.

6) Running time: The average running time for SFCs across all algorithms is shown in Fig. 13. It can be seen that the deployment time of the EP-SP, EP-DQN, and CG-BSFC algorithms is relatively balanced across different topologies based on the average execution time of more than 3000+ deployed SFCs. However, PPDM must predict node resources and respond before deployment, so when the topology scale increases, the algorithm's complexity will be relatively high, resulting in no running time advantage for SFCs deployment. In our simulation, PPDM requires between 26 and 45 seconds to converge at 2000 episodes of training, and the converged model will be directly applied to the online SFCs deployment without additional training, which will drastically reduce the deployment time. Since the single SFC deployment time is still at the millisecond level, it still considers acceptable in the IIoT environment.

D. Discussion

To verify the privacy-preserving performance, we compare both the privacy protection and non-privacy protection methods. Simulation results verify that the proposed PPDM approach achieves competitive performance compared with non-privacy protection methods, i.e., EP-SP and EP-DQN algorithms, which use a completely transparent network for SFCs deployment. Furthermore, compared to the privacy-preserving algorithms, i.e., the CG-BSFC approach which hides resource information of each domain based on the column generation method, our approach demonstrates more promising performance while retaining the ability to protect privacy.

However, the current method still has some disadvantages that need further research, for example, we have tried to consider more realistic IIoT use cases by considering the real-world topology and traffic of IIoT, but due to the lack of a specialized dataset designed for cross-domain deployment of IIoT, we still use existing distributed topologies for custom design. When applied to real IIoT networks in the future, the delay-constrained SFC deployment optimization is more convincing than pure latency reduction, and SFC deployment with both hardware-based NF and VNF needs to be considered as well since not all NFs in IIoT can be virtualized.

VI. Conclusion

This paper investigates a privacy-preserving deployment mechanism (PPDM) for SFCs across multiple domains, based on strict network topology and network resource information protection. Specifically, the deployment of SFCs across multiple domains has been considered a distributed problem managed by the MDC and presented as a hierarchical structure. First, MDC sends the received service requests to each domain controlled by IDC, each node returns a binary response to IDC indicating whether the nodes can deploy the current VNFs. Second, IDC predicts the virtual node's resources based on the response and matches the nodes with sufficient resource capacity to participate in the current VNF deployment and construct the SIRM. Then, the DQN-based CDDA algorithm uses the SIRM as input to learn the near-optimal SFCs deployment strategy. Finally, MDC distributes the strategy to each domain, and IDCs implement the specific cross-domain deployment and traffic scheduling. Simulation results demonstrate that the proposed PPDM achieves considerable performance both in privacy protection and SFCs deployment. The proposed method has promising application prospects in multi-domain IIoT and privacy-preserving assurance scenarios. Further work is in progress to explore the collective reinforcement learning (CRL)-based deployment method in multiple domains scenario.

References

[1] G. Sun, Y. Li, H. Yu, A. V. Vasilakos, X. Du, and M. Guizani, "Energy-efficient and traffic-aware service function chaining orchestration in multi-domain networks," *Future Generation Computer Systems*, vol. 91, pp. 347–360, 2019.

[2] X. Fu, F. R. Yu, J. Wang, Q. Qi, and J. Liao, "Dynamic service function chain embedding for NFV-enabled IoT: A deep reinforcement learning approach," *IEEE Transactions on Wireless Communications*, vol. 19, no. 1, pp. 507–519, 2019.

[3] N. H. Thanh, N. T. Kien, N. Van Hoa, T. T. Huong, F. Wamser, and T. Hossfeld, "Energy-aware service function chain embedding in edge-cloud environments for IoT applications," *IEEE Internet of Things Journal*, vol. 8, no. 17, pp. 13 465–13 486, 2021.

[4] G. Sun, Z. Xu, H. Yu, and V. Chang, "Dynamic network function provisioning to enable network in box for industrial applications," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 10, pp. 7155–7164, 2020.

[5] R. A. Ferrús Ferré, J. O. Sallent Roig, T. Rasheed, A. Morelli, H. Koumaras, G. Agapiou, C. Boustie, P. Gélard, R. Mestari, H. Makis *et al.*, "Enhancing satellite & terrestrial networks integration through NFV/SDN technologies," *Multimedia Communications Technical Commitee. IEEE Communications Society e-letter*, vol. 10, no. 4, pp. 17–21, 2015.

[6] V. Katewa, A. Chakrabortty, and V. Gupta, "Protecting privacy of topology in consensus networks," in *2015 American Control Conference (ACC)*. IEEE, 2015, pp. 2476–2481.

[7] Y. Liu, H. Zhang, D. Chang, and H. Hu, "GDM: A general distributed method for cross-domain service function chain embedding," *IEEE Transactions on Network and Service Management*, vol. 17, no. 3, pp. 1446–1459, 2020.

[8] G. Kibalya, J. Serrat, J.-L. Gorricho, D. Okello, and P. Zhang, "A deep reinforcement learning-based algorithm for reliability-aware multi-domain service deployment in smart ecosystems," *Neural Computing and Applications*, vol. 32, no. 1, pp. 1–23, 2020.

[9] N. Toumi, O. Bernier, D.-E. Meddour, and A. Ksentini, "On using physical programming for multi-domain SFC placement with limited visibility," *IEEE Transactions on Cloud Computing (Early Access)*, 2020.

[10] S. Ayoubi, S. Sebbah, and C. Assi, "A logic-based benders decomposition approach for the VNF assignment problem," *IEEE Transactions on Cloud Computing*, vol. 7, no. 4, pp. 894–906, 2017.

[11] S. Khebbache, M. Hadji, and D. Zeghlache, "Virtualized network functions chaining and routing algorithms," *Computer Networks*, vol. 114, pp. 95–110, 2017.

[12] D. Dietrich, A. Abujoda, A. Rizk, and P. Papadimitriou, "Multi-provider service chain embedding with nestor," *IEEE Transactions on Network and Service Management*, vol. 14, no. 1, pp. 91–105, 2017.

[13] T. Mano, T. Inoue, D. Ikarashi, K. Hamada, K. Mizutani, and O. Akashi, "Efficient virtual network optimization across multiple domains without revealing private information," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 477–488, 2016.

[14] T. Wang, H. Luo, X. Zheng, and M. Xie, "Crowdsourcing mechanism for trust evaluation in CPCS based on intelligent mobile edge computing," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 6, pp. 1–19, 2019.

[15] X. Wang, S. Garg, H. Lin, G. Kaddoum, J. Hu, and M. S. Hossain, "PPCS: An intelligent privacy-preserving mobile-edge crowdsensing strategy for Industrial IoT," *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10 288–10 298, 2020.

[16] K. D. Joshi and K. Kataoka, "pSMART: A lightweight, privacy-aware service function chain orchestration in multi-domain NFV/SDN," vol. 178. Elsevier, 2020, p. 107295.

[17] Y. Wang, P. Lu, W. Lu, and Z. Zhu, "Cost-efficient virtual network function graph (vNFG) provisioning in multidomain elastic optical networks," *Journal of Lightwave Technology*, vol. 35, no. 13, pp. 2712–2723, 2017.

[18] Q. Zhang, X. Wang, I. Kim, P. Palacharla, and T. Ikeuchi, "Vertex-centric computation of service function chains in multi-domain networks," in *Proceedings of the 2016 IEEE netsoft conference and workshops (NetSoft)*. IEEE, 2016, pp. 211–218.

[19] F. Tusa, S. Clayman, D. Valocchi, and A. Galis, "Multi-domain orchestration for the deployment and management of services on a slice enabled NFVI," in *Proceedings of the 2018 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*. IEEE, 2018, pp. 1–5.

[20] T. Pham, Y. Hadjadj-aoul, and A. Outtagarts, "VNF-FG embedding: a deep reinforcement learning approach," *IEEE Trans Netw Serv Manag*, vol. 16, no. 4, pp. 1–10, 2019.

[21] J. Pei, P. Hong, M. Pan, J. Liu, and J. Zhou, "Optimal VNF placement via deep reinforcement learning in SDN/NFV-enabled networks," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 2, pp. 263–278, 2019.

[22] Y. Liu, H. Lu, X. Li, Y. Zhang, L. Xi, and D. Zhao, "Dynamic service function chain orchestration for NFV/MEC-enabled IoT networks: A deep reinforcement learning approach," *IEEE Internet of Things Journal*, vol. 8, no. 9, pp. 7450–7465, 2020.

[23] H. A. Shah and L. Zhao, "Multiagent deep-reinforcement-learning-based virtual resource allocation through network function virtualization in Internet of Things," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3410–3421, 2020.

[24] X. Qi and M. Zong, "An overview of privacy preserving data mining," *Procedia Environmental Sciences*, vol. 12, pp. 1341–1347, 2012.

[25] C. Dwork, "Differential privacy: A survey of results," in *Proceedings of the International conference on theory and applications of models of computation*. Springer, 2008, pp. 1–19.

[26] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2014.

[27] S. Yang, F. Li, S. Trajanovski, X. Chen, Y. Wang, and X. Fu, "Delay-aware virtual network function placement and routing in edge clouds," *IEEE Transactions on Mobile Computing*, vol. 20, no. 2, pp. 445–459, 2019.

[28] D. T. Nguyen, C. Pham, K. K. Nguyen, and M. Cheriet, "Placement and chaining for run-time IoT service deployment in edge-cloud," *IEEE Transactions on Network and Service Management*, vol. 17, no. 1, pp. 459–472, 2019.

[29] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, and S. Davy, "Design and evaluation of algorithms for mapping and scheduling of virtual network functions," pp. 1–9, 2015.

[30] Z. Xu, J. Tang, J. Meng, W. Zhang, Y. Wang, C. H. Liu, and D. Yang, "Experience-driven networking: A deep reinforcement learning based approach," in *Proceedings of the IEEE INFOCOM 2018-IEEE conference on computer communications*. IEEE, 2018, pp. 1871–1879.

[31] H. Mao, M. Alizadeh, I. Menache, and S. Kandula, "Resource management with deep reinforcement learning," in *Proceedings of the 15th ACM workshop on hot topics in networks*, 2016, pp. 50–56.

[32] G. Stampa, M. Arias, D. Sánchez-Charles, V. Muntés-Mulero, and A. Cabellos, "A deep-reinforcement learning approach for software-defined networking routing optimization," *arXiv preprint arXiv:1709.07080*, 2017.

[33] C. Zhang, X. Wang, A. Dong, Y. Zhao, F. Li, and M. Huang, "The intelligent multi-domain service function chain deployment: Architecture, challenges and solutions," *International Journal of Communication Systems*, vol. 34, no. 1, p. e4665, 2021.

[34] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[35] S. Schneider, R. Khalili, A. Manzoor, H. Qarawlus, R. Schellenberg, H. Karl, and A. Hecker, "Self-learning multi-objective service coordination using deep reinforcement learning," *IEEE Transactions on Network and Service Management*, vol. 18, no. 3, pp. 3829–3842, 2021.

[36] A. Dwaraki and T. Wolf, "Adaptive service-chain routing for virtual network functions in software-defined networks," in *Proceedings of the 2016 workshop on Hot topics in Middleboxes and Network Function Virtualization*, 2016, pp. 32–37.

[37] I. Benkacem, T. Taleb, M. Bagaa, and H. Flinck, "Optimal VNFs placement in CDN slicing over multi-cloud environment," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 3, pp. 616–627, 2018.

[38] S. Singh, A. Okun, and A. Jackson, "Learning to play Go from scratch," *Nature*, vol. 550, no. 7676, pp. 336–337, 2017.

[39] P. Zhang, C. Wang, C. Jiang, and Z. Han, "Deep reinforcement learning assisted federated learning algorithm for data management of IIoT," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 12, pp. 8475–8484, 2021.

[40] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The internet topology zoo," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 9, pp. 1765–1775, 2011.

[41] J. Cai, Z. Huang, L. Liao, J. Luo, and W.-X. Liu, "APPM: adaptive parallel processing mechanism for service function chains," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1540–1555, 2021.

[42] R. Lin, S. Yu, S. Luo, X. Zhang, J. Wang, and M. Zukerman, "Column generation based service function chaining embedding in multi-domain networks," *IEEE Transactions on Cloud Computing*, vol. 11, no. 1, pp. 185–199, 2023.

Jun Cai is currently a professor and dean of the School of Cyber Security, Guangdong Polytechnic Normal University, Guangzhou, China. He received the B.S. degree from Hunan Normal University, Changsha, China, the M.S. degree from Jinan University, Guangzhou, China, and the Ph.D. degree from Sun Yat-Sen University, China in 2003, 2006, and 2012, respectively. He is interested in the research of network function virtualization (NFV), software-defined networks (SDN), and complex networks.

Zirui Zhou received the B.S. degree and the M.S. degree form Guangdong Polytechnic Normal University, Guangzhou, China in 2020 and 2023, respectively. She is currently working as a full-time faculty member at Guangzhou City Construction College, Guangzhou, China. Her current research interests include privacy protection, service function chain (SFC), and machine learning (ML).

Zhongwei Huang is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Macau University of Science and Technology, Macau, China. He is also a visiting student at Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ) in 2022. His current research interests include Artificial Intelligence (AI), Multi-access Edge Computing (MEC), and Privacy-preserving industrial Internet of things (IIoT).

Wenlong Dai received the B.S. degree in Software Engineering from East China Jiaotong University, in 2021. He is currently pursuing the master's degree with Electronic Information, Guangdong Polytechnic Normal University, Guangzhou, China. His current research interests include Artificial Intelligence (AI), service function chain (SFC) and machine learning (ML).

F. Richard Yu (Fellow, IEEE) received the PhD degree in electrical engineering from the University of British Columbia (UBC) in 2003. His research interests include connected/autonomous vehicles, artificial intelligence, blockchain, and wireless systems. He has been named in the Clarivate Analytics list of "Highly Cited Researchers" since 2019 with 40,000+ citatations (Google Scholar) and H-index 101+, and received several Best Paper Awards from some first-tier conferences. He was a Board Member the IEEE VTS and is the Editor-in-Chief for IEEE VTS Mobile World newsletter. He is a Fellow of the IEEE, Canadian Academy of Engineering (CAE), Engineering Institute of Canada (EIC), and IET. He is a Distinguished Lecturer of IEEE in both VTS and ComSoc.