Online Security-Aware and Reliability-Guaranteed AI Service Chains Provisioning in Edge Intelligence Cloud

Yu Qiu, Junbin Liang, Victor C.M. Leung, *Life Fellow, IEEE*, Min Chen, *Fellow, IEEE*

Abstract—With the rapid development of edge intelligence cloud (EIC), mobile users are not satisfied with a single artificial intelligence inference service, but require multiple inference services with chain dependencies to process data. Each AI service chain (AISC) is provided as a series of interconnected virtual network functions (VNFs) on-demand deployed on edge servers. However, AISCs experience unpredictable failures and potential attacks in EIC, which may violate different inference requirements of mobile users for reliability, security, and accuracy. To provide reliable AISCs, additional backup VNFs (BVNFs) need to be deployed nearby VNFs in case they experience faults, yet inference accuracy may degrade because these new links between the BVNFs and the VNFs would make AISCs vulnerable to more link attacks that affect data quality. How to optimally deploy VNFs and BVNFs on trusted edge servers, and select secure links to form satisfactory AISCs, meanwhile throughput of receiving requests is maximized while the deployment cost of computing resources used to create VNFs and BVNFs with different model sizes is minimized in real-time, is a challenging problem. In this paper, the problem is first formulated as an integer linear programming and proved to be NP-hard. Then, we consider the problem under two online backup scenarios: one is an on-site scenario where AISC requests from the mobile devices arrive one by one, and link securities between VNFs and corresponding BVNFs are ignored because they are always on the same edge server; another is an off-site scenario where a set of AISC requests are given, and VNFs and BVNFs are deployed on different servers. Finally, two online algorithms with provable competitive ratios are proposed to solve the above two problems in polynomial time. Theoretical analyses and experiments based on real network topologies demonstrate that our algorithms are promising compared to baseline algorithms.

Index Terms—Edge intelligence, virtual network function, service function chain, inference, reliability, security.

1 Introduction

1.1 Background

Edge intelligence cloud (EIC) converges artificial intelligence (AI), network function virtualization, and mobile edge computing to push various resources and compute-intensive AI inference services required by mobile user devices from cloud to edge servers [1], [2]. Deploying these AI services to resource-limited edge servers to meet different user requirements, namely **AI (e.g., deep learning) inference in edge** [3], has become one of the major choke points in edge intelligence. In real EIC scenarios (e.g., automatic drive, and intelligence home), multiple AI inference services would be required in the form of AI service chains (AISCs). The EIC not only enables mobile users to obtain tailored AISCs in a low-latency and energy-efficiency way [4], [5], [6], [7], [8], but also facilitates unleashing data potential through various intelligence analyses [9]. The AISCs provisioning[1] [1], [10], [11] refers to interconnecting and deploying sequences of virtual network functions (VNFs, e.g., augmented reality and medical image processing) on the edge servers [12].

1.2 Motivation

Real-world EIC networks suffer from inevitable insecurity and unreliability issues, which incurs AISC provisioning to be vulnerable to attacks and faults. In terms of security, attackers would obtain private information or pass malicious

This work was supported in part by the National Natural Science Foundation of China (Grant No. 62362005), in part by the Natural Science Foundation of Guangxi Province under Grant 2019GXNSFAA185042, in part by the Guangxi Key Research & Development Plan Project (No. Guike AB19259006), in part by the Thousands of Young and Middle aged Backbone Teachers Training Program for Guangxi Higher Education (Education Department of Guangxi (2017) No. 49), and in part by Shenzhen Science and Technology Innovation Commission (Grant R2020A045), in part by Major Projects for Innovation Driven Development in Guangxi (No. GuikeAA20302002), in part by Guangxi Science and Technology Base and Talent project (No. GuikeAD21076002), and part by the Innovation Project of Guangxi Graduate Education (No. YCSW2022037).

- *Corresponding author: Junbin Liang (email: liangjb@gxu.edu.cn).*
- *Yu Qiu and Junbin Liang are with the Guangxi Key Laboratory of Multimedia Communications and Network Technology, School of Computer, Electronics and Information, Guangxi University, Nanning, China (email: qyu@st.gxu.edu.cn).*
- *V. C. M. Leung is with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China, and also with the Department of Electrical and Computer Engineering, the University of British Columbia, Vancouver, BC V6T 1Z4, Canada (email: vleung@ieee.org).*
- *Min Chen is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, China (email: minchen@ieee.org).*

1. The differences between AISCs and traditional service chains (SCs): 1) Unique performance metric: Model accuracy needs to be additionally considered during AISCs provisioning, while SC does not. Accuracy can be affected by the size of model parameters and the quality of input data. 2) Deployment scheme: AISC provisioning is a variety of full offload schemes that refer to deploying all services requested by users to edge servers. Compared with compute-intensive AISC, part offload schemes (cooperation between user equipment and servers) are mainly used in SCs provisioning because traditional services can be partially and locally executed due to fewer resources required by SCs.

data to degrade well-trained AI models through link attacks, such as replay attacks, man-in-the-middle attacks, traffic analysis attacks, or VNF location shift attacks [13], [14]. Links are divided into secure and insecure based on security levels required by users and the availability of protection mechanisms (e.g., data encryption and digital signature) [15]. Physical isolation of VNFs from insecure links, namely topology validation [16], is an essential but effective method to ensure the confidentiality and integrity of user information. On this basis, compared with traditional networks, we have added some AI-based defense mechanisms [17] in the EIC to extend the available and existing ones, such as the DeepEC [18] etc. In terms of reliability, any hardware malfunctions (e.g., CPU and memory), software faults (e.g., VNFs and operating systems), and wrong operations will cause VNFs and even AISCs temporarily unreliable [19]. Deploying redundant backup VNFs (BVNFs) near VNFs is a practical way to prevent these faults. Specifically, one or multiple BVNFs are deployed near a single VNF, and these BVNFs are activated when the VNF experiences faults [4], [20].

However, the vulnerable EIC has limited resources compared with cloud data centers, which leads to bottlenecks in model size, and numbers of VNFs, BVNFs, and security mechanisms. The bottlenecks become obstacles to meet all user demands on inference reliability, security, accuracy, and sequence in real-time. Specifically, 1) arbitrary deployment of VNFs and BVNFs with inappropriate model sizes to form the AISCs rapidly consumes capacities of resource-constrained edge servers. 2) This also disturbs correct orders of data transmission among VNFs (e.g., firewall first, then face recognition) [21]. 3) Users are also subject to link attacks due to misuse of insecure links to transmit data, which reduces data quality and model accuracy. Note that the more BVNFs are requested, the more likely users are to be attacked due to the increase in number of communication links between BVNFs and VNFs. Therefore, how to implement secure and reliable AISC provisioning to maximize the throughput of receiving requests while minimizing the deployment cost, poses several significant challenges: 1) how to optimally select the VNFs to add BVNFs in each AISC, and how many BVNFs are required to meet reliability requirements, 2) how to optimally select deployment locations and model sizes of VNFs and corresponding BVNFs to meet inference sequence and accuracy demands and capacity constraint, 3) how to select secure links to connect each user with trusted edge servers hosting the VNFs and BVNFs to meet security requirements.

1.3 Contribution

In this paper, the problem first is formulated as an integer linear programming, and proved to be NP-hard by reducing it to the Lowest cost Generalized Assignment Problem (LGAP) [22], [23]. Then, the problem is analyzed under two online backup scenarios without considering any future information, namely **on-site** and **off-site**. The scenarios represent two different deployment location constraints. In the former (on-site) scenario, AISC requests from users arrive one after another, and VNFs and corresponding BVNFs must be on the same edge server. This not only eliminates concerns of link security between them but also reduces switching delay and bandwidth consumption between them. However, all VNFs of the same type in a single edge server experience faults when the server fails due to malfunctions or attacks. In the latter scenario, a set of AISC requests arrive at the same time, VNFs and BVNFs are deployed on different servers to mitigate the impact of single edge server faults. Finally, we propose two online algorithms with provable competitive ratios to solve the above two problems in polynomial time.

NEW INSIGHT: To the best of our knowledge, we are the first to formulate AISCs provisioning in EIC from new insight into antagonistic effects among reliability, security, and accuracy. Enhancing AISC reliability would sacrifice its accuracy, since attacks from these additional links among VNFs and BVNFs lead to deterioration of input data; but, improving AISC accuracy would sacrifice its reliability, since additional model parameters require more resources resulting in fewer BVNFs that can be deployed. Both qualities of input data and sizes of model parameters are affected by link security in our model. In addition, we also consider chain dependencies between multiple AI inference services with different models requested by multiple users in terms of model granularity, rather than the simple scenario of a single user or single model.

The main contributions of this paper are as follows:

- To the best of our knowledge, we are the first to formulate AISCs provisioning in the EIC as an integer linear programming from a new perspective of antagonistic effects among reliability, security, and accuracy, and prove it to be NP-hard.
- We propose two online algorithms with provable competitive ratios to solve the NP-hard problem under two online backup scenarios. The scenarios have different request arrival patterns, VNFs sequence per user, deployment location, reliability, security, and capacity constraints.
- Theoretical analyses and extensive experiments based on the real network topologies show that our algorithms effectively adapt to insecure and unreliable EIC networks compared to baseline algorithms.

The rest of this article is organized as follows. Section II reviews related works. Section III introduces mathematical models. In Section IV, algorithm and mathematical analyses are introduced under on-site backup scenario. Section V introduces related algorithm of the off-site backup scenario. Section VI evaluates the performances of our algorithms. Section VII summarizes the full paper.

2 Related Work

As the next generation of mobile edge computing, edge intelligence cloud (EIC) has attracted the attention of many scholars. Reliability and security in EIC are extremely significant but easily overlooked. In overviews [9], [13], [24], [25], [26], problems and solutions related to failures and attacks (such as type, detection, and recovery) are initially explored and discussed. In this paper, we also divide related research into two categories: security and reliability.

2.1 Security

This kind of research effectively ensures security by physically isolating services from unsafe infrastructures, but lacks reliable measures to provide continuous and stable services in the face of unpredictable faults.

Bays et al. [27] was one of the first to study security-aware network function virtualization. He considered multiple link encryptions (e.g., end-to-end or point-to-point) and proposed a metaheuristic algorithm combining security requirements and resource allocation. On this basis, Liu et al. [15] took node security into extra consideration. He simplified security requirements into security levels, and proposed two heuristic algorithms to solve services provisioning problems. Dwiardhika et al. [28] proposed a genetic algorithm combining node security and the chain ordering of VNFs, but they ignored dependencies and link securities between VNFs. Zhang et al. [14] used slicing technology to construct multiple virtual networks in edge computing, where VNFs were deployed by a joint security and link bandwidth architecture. In addition, Bagga et al. [29] designed an orchestration system with extra consideration of link security, delay, and various resources (e.g., CPU, RAM) to manage the life-cycle of chains of VNFs across multiple cloud-edges. Zhang et al. [30] [31] used four and five (addition: node degree) features as the input of reinforcement learning-based algorithms to solve VNFs deployment problems in polynomial time respectively.

2.2 Reliability

This kind of paper guarantees reliability by deploying additional BVNFs near VNFs, but lacks security measures to provide confidential and complete services in the face of potential attacks.

Fan et al. [32] first studied the shared backup scheme, namely online joint protection (JP), which simultaneously selected and protected the two least reliable VNFs to improve the reliability of chain of VNFs. Kanizo et al. [33] adopted offline backup schemes to improve the full survival probability of failed nodes and resist small-scale faults in the scenario with resources-limited and ignoring VNF dependencies. Qu et al. [34] proposed a centralized algorithm to combine VNFs chain order, delay, and reliability to ensure different QoS requirements of users. Huang et al. [4] deployed VNFs and BVNFs required by different users through approximation algorithms to maximize network throughput. Li et al. [20] considered an online model and extended the problem [4] to on-site and off-site backup scenarios. Shang et al. [12] came up with an algorithm that used static and dynamic backup methods in edge-cloud networks with unknown fault probability to provide the chain of VNFs. Wang et al. [35] considered the influence of VNFs connection construction stage on reliability in a scenario where both hardware and software could fail, aiming to minimize bandwidth consumption while meet reliability of users.

2.3 Security and reliability

This type of paper preliminarily realizes reliability and security to a certain extent in edge computing cloud networks.

A review article [17] comprehensively analyzed the reliability and safety of VNF, and preliminarily discussed the trade-off between safety and reliability. Park et al. [36] investigated how to route secure VNFs with the lowest cost, and design a fail-recovery route mechanism in data centers. Based on the model in [36], Feng et al. [37] considered the uncertainty of attack and network capacity to design secure routes of VNFs and fail-recovery routes. Thiruvasagam et al. [38] considered deploying both the chain of VNFs and BVNFs to enhance reliability; in addition, virtual monitoring functions were also deployed to identify and mitigate performance degradation after malfunctions. Wang et al. [39] designed a deep reinforcement learning algorithm to develop a resilient recovery scheme for improving service success rate after attacks.

However, there are obvious differences between existing papers and our study: 1) We consider AISCs provisioning in EIC from new insight into antagonistic effects among reliability, security, and accuracy, and we nontrivial formulate these performance models at the same time. 2) We also consider chain dependencies between multiple AI inference services with different models requested by multiple users in terms of model granularity in EIC networks, rather than the simple scenario of a single user or single model.

3 Problem Description

In this section, we begin with introducing related system models. Then, the AISCs provisioning problem is proved to be NP-hard. Finally, the problem is defined precisely and formulated as an integer linear programming (ILP) under on-site and off-site backup scenarios.

3.1 Network model

We model a resource-constrained EIC network to be an undirected weighted graph $G_s = (V^s, E^s)$. $V^s = V \cup C$ represents all access point (AP) nodes, and $E^s = \{e_{u,v} | u, v \in V^s\}$ signifies a group of physical links connecting any two APs. In addition, V^s and E^s have the attributes of computing capacity and security level, respectively. Specifically, C is a set of edge servers with computing capacity $cap_c > 0$ co-located with AP, and V with $cap_v = 0$ is only responsible for transmitting data. Each physical link e has different security levels S_e due to different security mechanisms (e.g., data encryption and digital signature). In the rest of this article, servers and edge servers are interchangeable unless otherwise specified. In addition, we assume that each e is a high-capacity optical cable, which refers to any e has sufficient bandwidth capacity. Notice the assumption has been adopted in literature about VNFs deployment [5], [40], and this has little effect on the performances of deployment algorithm [41].

3.2 User request model

$F = \{f_j | 1 \leq j \leq |F|\}$ represents all VNF types. The traditional VNF instance has only one model, while the AI-based VNF inference instance has multiple models. We define VNF as $f_j \triangleq [acc_f, cap_f]$, where acc_f denotes the size of model parameters, and cap_f is the corresponding

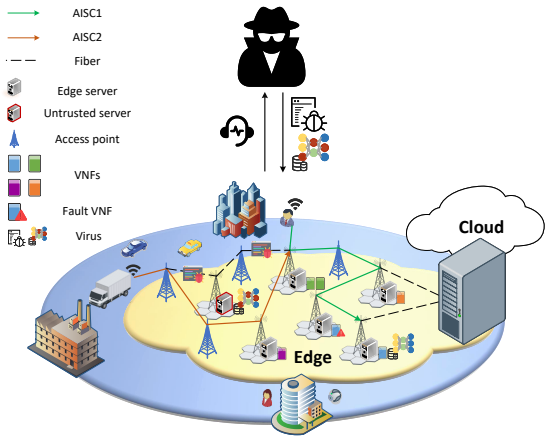


Fig. 1: The EIC network.

computing capacity of f_j with acc_f (e.g., VGG-16 and VGG-19 [42] for image processing [43], [44]). The larger the size of AI model parameters, the higher the accuracy of the calculation results; therefore, acc_f also represents the initial model accuracy of f_j [45]. Γ is denoted by a collection of single AISC request $\gamma = (L^\gamma, F^\gamma, R^\gamma, S^\gamma, A^\gamma)$ at any time. Specifically, let L^γ be a location where a user issues AISC request. Note that we use the macroscopic AI model as deployment granularity in this paper [1], [2], that is, the deployment unit is a single AI service with whole partition dependency, and various AI service required by a user constitute an AISC. $F^\gamma = \{f_i^\gamma | 1 \le i \le |F^\gamma|, f_i^\gamma (\neq f_i) \in F\}$ indicates personalized AI inference chain requested by a user, where index i indicate the required order of each VNF f_i^γ. R^γ, S^γ, A^γ are user requirements on reliability, security, and accuracy level, respectively. In addition, when a user is in an overlapping area covered by multiple APs, we assume that the user device will connect to the closest AP or the one with the strongest signal strength [5], as is shown in Fig. 1.

3.3 Reliability, security, and accuracy model

Reliability and security are extremely important but easily overlooked because they are not always considered top priorities. To this end, we explicitly consider reliability and security of AISC provisioning in the EIC.

3.3.1 AISC reliability model

In the reliability model, VNFs may be in either normal or faulty states. Failures of edge servers or virtual machines would cause VNF faults. The reliability of a VNF f can be calculated by the VNF's normal working time (uptime) probability [46], which can be obtained from the operating data or log files on the server. Therefore, the reliability r_f of VNF is:

$$r_f = \frac{T_{uptime}}{T_{all}} = \frac{MTTN}{MTTN + MTTR} \tag{1}$$

$MTTN$ and $MTTR$ represent mean time to normal and mean time to repair respectively. A complex AISC consists of multiple interconnected VNFs. Without loss of generality, we assume that each VNF fault occurs independently and does not affect or extend to other VNFs. Therefore, we define the reliability of AISC as the product of the reliabilities of corresponding VNFs:

$$\mathbb{R}^\gamma = \prod_{f \in F^\gamma} r_f \tag{2}$$

Unfortunately, although various VNFs as a whole make the AISC more powerful, any VNF fault would interrupt the inference service and incur data loss. To achieve reliable AISC, a set of additional BVNFs $\cup BF_i = \{f_{ib}^m | i \in F, 1 \le m \le |f_{ib}^M|\}$ need to be deployed nearby VNFs. The VNF experiences fault only when all BVNFs and itself fail simultaneously. Therefore, the reliability of protected VNF and AISC are as follows:

$$\begin{aligned}\hat{r}_f =& Pr\left[\text{at least one VNF of each type is available}\right] \\ =& 1 - (1 - r_{f_i}) \prod_{f_{ib}^m \in BF_i, 1 \le m \le |f_{ib}^M|} \left(1 - r_{f_{ib}^m}\right)\end{aligned} \tag{3}$$

$$\hat{\mathbb{R}}^\gamma = \prod_{f \in F^\gamma} \hat{r}_f \tag{4}$$

For the sake of convenience, we assume that BVNFs have the same reliability as the corresponding VNF, and this assumption has been widely adopted in literature [32], [33], [34], [4].

3.3.2 AISC security and accuracy models

In the security model, link security[2] and edge server security are considered:

1) Link security: We formulate link security from data integrity, confidentiality, and authorization. Firstly, data integrity S_{di}^e of a link e can be guaranteed by hash functions and calibration techniques (e.g., cyclic redundancy check) during transmission. S_{di}^e in this paper is regarded as a basic attribute (data quality) and its value can be calculated based on historical exception information of links (e.g., packet loss rate or hash error rate) [3]. $N_{exception}$ indicates exception times of link e during total monitoring time N_{all}.

$$S_{di}^e = 1 - \frac{N_{exception}}{N_{all}} \tag{5}$$

Next, confidentiality can be ensured by various protection mechanisms, such as encryption algorithms and protocols (TLS and SSH). The higher number S_c^e of available protection mechanisms of a link, the more effective it is against attack impacts, such as packet loss or packet retransmission. S_{max} is settled to the highest availability of security mechanisms in advance. Finally, random key pre-distribution schemes [47] (e.g., Eschenauer and Gligor scheme) can be used for authorization among servers. Each of the $|C|$ (number of servers) servers is assigned K_C distinct keys that are selected uniformly at random from

2. The basic difference between the factors of security and congestion is that 1) reliability demands of users affect security, but it has little impact on network congestion because the communications between VNFs and BVNFs are at regular intervals, rather than using bandwidth all the time. 2)Low-security links can continue to be used in certain cases where the security demands of users are low than security levels of the links, while congested links cannot be used.

3. www.crawdad.org/due/packet-delivery. The dataset contains some link information, such as deliver success/fail, overflow, link quality indicator, noise floor, etc.

a key pool of size P_C. These K_C keys make up a key ring S_u of a server u, and any two servers (u and v) can securely communicate over a link if and only if they have at least one key in common, namely $[S_u \cap S_v \neq \varnothing]$ is true. Therefore, link security level S_e is defined as:

$$S_e = \lfloor S_{di}^e \cdot S_c^e \rfloor \cdot [S_u \cap S_v \neq \varnothing] \tag{6}$$

Note that our security model improves and refines the concept of security levels that has been applied to both articles [14], [15], [27], [28], [30], [31] and actual projects (e.g., H2020 Anastacia EU project [29] and Alibaba). When considering an AISC that contains multiple links, its security level is defined as follows:

Definition 1. *Define $\mathbb{S}^\gamma$ as the link security level of AISC, which is the minimum link security level in the link group interconnected to corresponding VNFs in AISC:*

$$\mathbb{S}^\gamma = \min_{e_{f_i,f_{i+1}} \in F^\gamma, 1 \leq i \leq |F^\gamma|-1} S_e \tag{7}$$

2) Edge server security: The trust evaluation mechanisms are designed to solve and avoid attacks from hardware and links [48], [49]. The trust values are calculated based on direct and indirect evidence. In terms of the direct trust value T_c of edge server c, we adopt the below calculation model that has been applied to articles [50], [51], [52].

$$T_c = \frac{2b+u}{2},\ where\ b = \frac{s}{s+f+1},\ u = \frac{1}{s+f+1} \tag{8}$$

Where, $b, d, u > 0$ indicates belief, disbelief, and uncertainty, and $b+d+u = 1$. s represents numbers of generating desired results, which means AI models on service c are normal. f is unsuccessful numbers. Note that paper [53] has proposed a practical method to calculate and obtain s and f.

In terms of the indirect trust values, we consider the average fault probability $\bar{\rho}_{f,c}$ of all VNFs running on the server. The average difference of security level $\bar{S}_{\gamma,c}$ of the shortest path from a user to the server.

$$\bar{\rho}_{f,c} = 1 - \bar{r}_{f,c} = 1 - \frac{1}{|N_{f,c}|} \sum_{n=1}^{N_{f,c}} r_{f,c} \tag{9}$$

$$\bar{S}_{\gamma,c} = \frac{1}{|P_{\gamma,c}|} \sum_{n=1}^{P_{\gamma,c}} (S_e - S^\gamma) \tag{10}$$

Where, $N_{f,c}$ is a set of VNFs on c. $r_{f,c}$ represents the reliability of VNF f on c. $P_{\gamma,c}$ indicates a group of links consisting the shortest path from γ to c. Thus, the total trust value $T_{\gamma,c} \in [-2, 2]$ of edge server c for AISC request γ is as follows. In addition, $T_{\gamma,c} \geq T_{thr}$ indicates that edge service c is secure and trusted for the γ.

$$T_{\gamma,c} = T_c + \frac{\bar{S}_{\gamma,c}}{S_{max}} - \bar{\rho}_{f,c} \tag{11}$$

3) Accuracy: In the accuracy model, data quality and the size of AI model parameter are considered.

Data quality can affect the convergence speed and accuracy of AI model during the training and inference process [54]. Data quality can be improved by providing additional link protection measures. The size of AI model parameter can affect the neural network structure, such as fully connected, recurrent, and convolution structures [45]. Even when other layers are constant, having more hidden layers usually reduces the approximation error, and thus inference accuracy is higher. Once the size of AI model parameter acc_f and the corrected data quality through protection measures $\lfloor S_{di}^e \cdot S_c^e \rfloor$ are determined, the real inference accuracy level acc_f^r of VNF f is calculated as:

$$acc_f^r = \lfloor acc_f \cdot \lfloor S_{di}^e \cdot S_c^e \rfloor \rfloor \tag{12}$$

$$\mathbb{A}^\gamma = \min_{i \in \gamma^k} acc_{f_i}^r \tag{13}$$

In addition, the size of AI model parameters is also calculated and selected based on the inference accuracy requirement A^γ and the corrected data quality through protection measures $\lfloor S_{di}^e \cdot S_c^e \rfloor$. Based on the intuition that high-accuracy models are often used in important scenarios that are more attractive to attackers, we assume that the security requirement of the user is always higher than the accuracy requirement. The selected model size acc_f^s of VNF f is calculated as:

$$acc_f^s = \frac{A^\gamma}{\lfloor S_{di}^e \cdot S_c^e \rfloor},\ when\ A^\gamma \leq S^\gamma \tag{14}$$

3.4 Cost model

In resource-limited EIC, deployment cost is also a major factor affecting AISC provisioning. In the cost model, the deployment cost is related not only to the security level requested by a user, but also to the workloads of a server. The underlying reasons are that 1) the number of protection mechanisms increases with the security levels required by users, and 2) the fault probability of server increases with its workloads. These require additional resources for protection mechanisms to ensure stable AISCs. For VNFs and BVNFs deployed on server v by request γ_k, the deployment cost ψ_v^k is as follows [4]:

$$\psi_v^k = S^{\gamma_k} \cdot C_v \cdot \left(\alpha^{1 - \frac{C_v(k)}{C_v}} - 1 \right) \tag{15}$$

$$C_{vu}(k) = \sum\nolimits_{i=1}^{|F^\gamma|} \sum\nolimits_{m=1}^{|f_{ib}^M|} cap_{f_i} \cdot \left(x_{i,v}^k + y_{ib_m,v}^k \right) \tag{16}$$

$$C_v(k) = C_v(k-1) - C_{vu}(k) \tag{17}$$

Where, $\alpha > 1$ is a constant that reflects the workload sensitivity on each server. C_v, $C_{vu}(k)$, and $C_v(k)$ ($C_v(0) = C_v$) represent the initial computing capacity of server v, the computing resources used to accept request γ_k, and the remaining computing capacity after receiving γ_k, respectively. Let $x_{i,v}^k$ be a binary variable indicating whether the ith VNF in γ_k is deployed on v. Set $y_{ib_m,v}^k$ as a binary variable to indicate whether the mth backup of the ith VNF in γ_k is

4. The reasons for choosing a more conservative exponential cost model instead of a linear one are as follows: 1) the fault probability of a server is exponential with temperature generated by high workloads [55]. The more important factor is 2) when a server fails due to excessive workloads, the workloads will be redistributed to normal servers, which causes cascading failures and exponential impacts [56]. The exponential model can effectively avoid deploying VNFs on servers with high workloads.

deployed on v. The total deployment cost C^{γ_k} for request γ_k is expressed as:

$$C^{\gamma_k} = \sum_{v \in C} \psi_v^k \tag{18}$$

In addition, the normalized cost of reflecting server workload after deploying VNFs and BVNFs in γ_k is defined as follows:

$$\phi_v^k = \frac{\psi_v^k}{C_v} = S^{\gamma_k} \cdot \left(\alpha^{1-\frac{C_v(k)}{C_v}} - 1 \right) \tag{19}$$

3.5 Problem definition

The AISCs provisioning problem in EIC is how to optimally deploy VNFs and BVNFs on trusted servers, and select secure links and model size to form expected AISCs that have higher reliability $\hat{\mathbb{R}}^{\gamma_k}$, security $\mathbb{S}^{\gamma_k}$, and accuracy $\mathbb{A}^{\gamma_k}$ than user requirement levels (R^{γ_k}, S^{γ_k}, A^{γ_k}, and T_{thr}), meanwhile throughput of receiving requests is maximized while deployment cost of instantiating VNFs and BVNFs is minimized in real-time. Proving the problem to be NP-hard is as follows:

Theorem 1. *The problem of online security-aware and reliability-guaranteed AISCs provisioning (SRAP) in resource-limited EIC is NP-hard.*

Proof. We prove the NP-hardness of the SRAP problem by a reduction from the Lowest cost Generalized Assignment Problem (LGAP) as follows. Given a group of items $Item = (i_1, i_2, ..., i_n)$ with different cost $cost_i$ and size $size_i$, and a group of bins $Bin = (b_1, b_2, ..., b_m)$ with different capacities cap_{b_i}. The LGAP is a variant of the classical NP-hard Generalized Assignment Problem (GAP), which aims to pack as many items into bins as possible at minimal cost, while the total size of items in each bin does not exceed its capacity cap_b.

We consider a special instance I of the offline SRAP problem, where I consists of $G_s = (C \cup V, E^s)$. In addition, all AISC requests arrive at the same time, and each request contains only one VNF. Each server $c \in C$ and $\gamma \in \Gamma$ have cap_c and cap_f computing capacity, respectively. The deployment cost C^{γ} of VNFs and BVNFs per request not only is related to security level demand but also varies with server workloads. Different requests occupy different computing resources cap_γ due to differences in VNF computing capacity and backup quantity. In this article, we investigated how to deploy VNFs and BVNFs on a set of edge servers, aiming to maximize the throughput of receiving requests while minimizing deployment cost. I is NP-hard because the reduction process can be completed in polynomial time. So, the SRAP problem is NP-hard too. That's the proof. □

In this paper, we consider the SRAP optimization problem under two online backup scenarios without any future information. Precise definitions of SRAP problem under two scenarios are given below.

Definition 2. *Given a resource-limited EIC network $G_s = (V^s, E^s)$ with a set C of edge servers with computer capacity $cap_c > 0$, a set of links e with different security levels, and request $\gamma = (L^\gamma, F^\gamma, R^\gamma, S^\gamma, A^\gamma)$ arrive one by one without any future information, the SRAP under on-site online backup scenario (SRAPON) is to receive as many AISC requests as possible with minimal cost of instantiating VNFs and BVNFs, that is maximizing the throughput of receiving requests and minimizing deployment costs. Meanwhile, the reliability demand R^{γ_k}, security demand S^{γ_k}, accuracy demand A^{γ_k}, service demand F^{γ_k}, and computing demand cap_γ of each receiving request γ_k are met, subject to resource capacity and deployment location constraint (VNFs and their BVNFs must be deployed on the same edge server) in G_s.*

Definition 3. *Given an EIC network $G_s = (V^s, E^s)$ including a collection C of edge servers with computer capacity $cap_c > 0$, a set V of APs with $cap_v = 0$, a group of links e with distinguishing security levels, and a set of new arrival requests $\gamma = (L^\gamma, F^\gamma, R^\gamma, S^\gamma, A^\gamma)$ without any future information, the SRAP under off-site online backup scenario (SRAPOFF) is to accept as many AISC requests as possible with minimal cost of instantiating VNFs and BVNFs. Meanwhile, the reliability demand R^{γ_k}, security demand S^{γ_k}, accuracy demand A^{γ_k}, service demand F^{γ_k}, and computing demand cap_γ of each received request γ_k are met, subjected to resource capacity and deployment location constraints (the VNFs in an AISC must be deployed on different servers, and VNFs and corresponding BVNFs must also be deployed on different servers) in G_s.*

3.6 ILP model

In this section, we formulate SRAPON and SRAPOFF problem as integer linear programming (ILP), respectively. Let X^{γ_k} be a boolean variable indicating whether request γ_k is accepted. Set BN_i^γ as the backup number of VNF f_i in γ_k. A_{rr} and A_{cc} are denoted by the set of new arrival AISC requests and the group of receiving requests. The SRAPON problem is formulated as ILP with the optimization objective:

$Objective$:

$$Minimize \quad \sum_{r_k \in A_{rr}} C^{\gamma_k} \cdot X^{\gamma_k} \tag{20}$$

$Subject\ to$:

$$(15), (16), (17), (18)$$

$$X^{\gamma_k} = \min_{i \in F^{\gamma_k}} \left(\sum_{v \in C} x_{i,v}^k \right) \left[\hat{\mathbb{R}}^{\gamma_k} \geq R^{\gamma_k} \right] \cdot \left[\mathbb{S}^{\gamma_k} \geq S^{\gamma_k} \right] \left[\mathbb{A}^{\gamma_k} \geq A^{\gamma_k} \right] \tag{21}$$

$$A_{cc} = \bigcup_{\gamma_k \in A_{rr}} \gamma_k \left[X^{\gamma_k} > 0 \right] \tag{22}$$

$$\sum_{v \in C} x_{i,v}^k = 1, f_i \in \gamma_k, \gamma_k \in A_{cc} \tag{23}$$

$$\sum_{v \in C} y_{ib_m,v}^k = 1, i \in F^{\gamma_k}, \gamma_k \in A_{cc}, ib_m \in BF_i^{\gamma_k} \tag{24}$$

$$\sum_{f_i \in \gamma_k} cap_{f_i} \cdot \left(x_{i,v}^k + y_{ib_m,v}^k \right) \leq C_v(k-1), v \in C, \gamma_k \in A_{cc} \tag{25}$$

$$acc_{f_i}^r \geq A^{\gamma_k}, f_i \in \gamma_k, \gamma_k \in A_{cc} \tag{26}$$

$$\prod_{f \in F^{\gamma_k}} \hat{r}_f \geq R^{\gamma_k}, \gamma_k \in A_{cc} \tag{27}$$

$$\sum_{v=c\in C} x_{i,v}^{k} \cdot T_{\gamma_k,c} \geq T_{thr}, f_i \in \gamma_k, \gamma_k \in A_{cc} \tag{28}$$

$$\min_{e_{f_i,f_{i+1}} \in F^{\gamma_k}, 1\leq i\leq |F^{\gamma_k}|-1} S_e \geq S^{\gamma_k}, \gamma_k \in A_{cc} \tag{29}$$

$$BN_i^{\gamma_k} \cdot x_{i,v}^{k} = \sum_{ib_m \in BF_i^{\gamma}} y_{ib_m,v}^{k}, i \in F^{\gamma_k}, v \in C, \gamma_k \in A_{cc} \tag{30}$$

$Variable$:

$$X^{\gamma_k} = 1, \gamma_k \in A_{cc} \tag{31}$$

$$X^{\gamma_k} \in \{0,1\}, \gamma \in A_{rr} \tag{32}$$

$$x_{i,v}^{k} \in \{0,1\}, v \in C, i \in F^{\gamma_k}, \gamma_k \in A_{rr} \tag{33}$$

$$y_{ib_m,v}^{k} \in \{0,1\}, v \in C, ib_m \in BF_i^{k}, \gamma_k \in A_{rr} \tag{34}$$

The SRAPOFF problem is formulated as ILP with the optimization objective:

$Objective$:

$$Minimize \quad \sum_{r_k \in A_{rr}} C^{\gamma_k} \cdot X^{\gamma_k}$$

$Subject\ to$:

$$(15) - (18), (21), (22), (24) - (29)$$

$$\sum_{i \in F^{\gamma_k}} x_{i,v}^{k} \leq 1, v \in C, \gamma_k \in A_{rr} \tag{35}$$

$$x_{i,v}^{k} + \sum_{ib_m \in BF_i^{\gamma_k}} y_{ib_m,v}^{k} \leq 1, i \in F^{\gamma_k}, v \in C, \gamma_k \in A_{rr} \tag{36}$$

$Variable$:

$$(31), (32), (33), (34)$$

Constraint (21) indicates that a request is accepted only when requirements on reliability, security, accuracy, and inference sequence are met. Constraint (23) and (24) protect data integrity by limiting each requested (B)VNF to be deployed on a single server. Constraint (25) shows that computing capacity occupied by all VNFs and BVNFs in v cannot exceed the remaining computing capacity of v. Constraint (27), (29), and (26) indicate that the reliability, security, and accuracy of each accepted request are higher than user requirement. Constraint (28) represents all VNFs and BVNFs are deployed on trusted servers. Constraint (30) in SRAPON problem is deployment location constraint, where VNFs and their BVNFs must be deployed on the same edge server. Constraint (35) and (36) in the SRAPOFF problem are deployment location constraints, where the VNFs in an AISC must be deployed on different servers, and VNFs and corresponding BVNFs must also be deployed on different servers.

4 On-site Scenario

4.1 The construction of auxiliary graphs

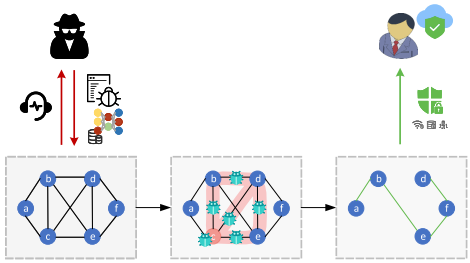


Fig. 2: An auxiliary graph.

Note that the construction of auxiliary graphs is only the first step of our algorithms. To reduce the risk of being attacked, the links and servers whose security and trust are lower than user requirements cannot be used for data traffic and data processing. For clarity, let $G_{S^\gamma} = (V^{S^\gamma}, E^{S^\gamma})$ be an auxiliary graph, constructed according to security level S^γ and trust threshold T_{thr}. Specifically, in terms of edges, the links whose security is lower than S^γ in E^s are first removed to form $E^{S^\gamma} \subset E^s$ through the Breadth-First Search algorithm. G_{S^γ} may contain more than one connected components. In terms of servers, trust values of edge servers for a user in each connected component are calculated based on the formula (8), (9), (10), (11). The servers are next removed to form $V^{S^\gamma} \subset V^s$, when their trust value is lower than T_{thr}. Fig. 2 gives a two-stage auxiliary graph example.

4.2 SRON algorithm

The basic idea behind the SRON algorithm is to provide secure and reliable AISC services through redundancy protection and physical isolation, and the detailed process of Algorithm 1 (SRON) is as follows:

Step 1. Based on the above construction principles, a dedicated auxiliary graph G_{S^γ} for each new arrival request γ is built, namely links with low security and servers with low trust are isolated from AISCs. During constructing G_{S^γ} due to partial link removal, multiple connected components are often formed. To maintain the deployment location constraint, namely VNFs and corresponding BVNFs must be deployed on the same edge server, the request will be rejected when there are fewer than $|F^\gamma|$ trusted servers in the selected connected component. In addition, only the connected component Λ that has maximum trusted server numbers and contains the AP closest to the user is finally considered. In this selection process, the number of hops is used as the primary parameter of distance, but when the number of hops is the same, the actual distance is used as the auxiliary judgment. This is because data packets are often split, copied, and consolidated at every node in its routing path due to the maximum packet segment limit in the TCP/IP protocol during transmission in a multi-hop network [57]; in addition, since a packet would consume limited resources of multiple routers in a multi-hop network, the queuing delay of the whole network will be

Algorithm 1: (SRON) Security-aware and reliability-guaranteed algorithm under on-site scenario

Input: The resource-limited EIC network $G_s = (V^s, E^s)$ with computing capacity C_v, and a group of requests A_{rr} that arrive one by one without any future information
Output: a set of receiving requests A_{cc}

begin
  $A_{cc} = \emptyset$, $BN = \emptyset$;
  for *each new arrival* γ_k **do**
    Construct auxiliary graph $G_{S^\gamma} = (V^{S^\gamma}, E^{S^\gamma})$;
    Select a connected component Λ that is closest to the user, and contains the most trusted server numbers $|C^{\gamma_k}|$ in G_{S^γ};
    if $|C^{\gamma_k}| < |F^{\gamma_k}|$ **then**
      Reject request γ_k;
      Continue;
    Select model of each VNF in γ_k based on (14);
    Calculate the reliability $\mathbb{R}^{\gamma_k}$ of γ_k based (2);
    while $\mathbb{R}^{\gamma_k} < R^{\gamma_k}$ **do**
      $i \leftarrow \min_{f_i \in \gamma} r_{f_i}$, $BN_i^{\gamma_k}$ += 1, update $\hat{r}_{f_i}$, $\hat{\mathbb{R}}^{\gamma_k}$ based (3) and (4);
      $\mathbb{R}^{\gamma_k} \leftarrow \hat{\mathbb{R}}^{\gamma_k}$;
    Construct LGAP instance that is consisted of $|F^{\gamma_k}|$ items with $(BN_i^{\gamma_k} + 1) \cdot cap_f$ size and ψ_v^k weight, and $\left|C^{S^\gamma} = \Lambda \cap C\right|$ bins with $C_v(k-1)$ capacity;
    Solution $\mathbb{S} = \cup C_{s_i}$ is obtained by an algorithm [23];
    if $\exists C_{s_i} < 0$ **then**
      Reject request γ_k;
    else
      for *each VNF and BVNF* f_i *in* γ_k **do**
        Deploy f_i on edge server C_{s_i};
        Update resources and $\bar{\rho}_{f,c}$;
      $A_{cc} \leftarrow A_{cc} \cup \gamma_k$;
  return A_{cc};

amplified greatly, much higher than other delays such as propagation, transmission (distance), and processing [58].

Step 2. The appropriate model size of each VNF is selected based on the corrected data quality of each e in Λ and accuracy demand A^{γ_k}, and then redundancy BVNFs are iteratively deployed near the least reliable VNF in each unreliable AISC until its reliability exceeds user requirements. The backup process is implemented by solving special LGAP instances, and the steps of constructing an LGAP instance are as follows. We virtualize VNF and its corresponding BVNFs as an item. Each LGAP instance is consisted of $|F^\gamma|$ items with $\sum_{v \in C^{S^\gamma} = \Lambda \cap C} \psi_v^k \cdot x_{i,v}^k$ cost and $(BN_i^{\gamma_k} + 1) \cdot cap_f$ size, and $\left|C^{S^\gamma}\right|$ bins (edge servers) with remaining computing capacity $C_v(k-1)$. The approximate deployment solution $\mathbb{S} = \cup C_{s_i}$ of LGAP instance is obtained by an algorithm [23], whose cost is no more than 3 times of the optimal cost. Each C_{s_i} represents a deployment location of the corresponding item. If the remaining capacity of the server is greater than computing capacity required by the item, C_{s_i} stores the server index; otherwise, C_{s_i} is assigned to -1.

Step 3. The required VNFs and BVNFs are deployed sequentially according to C_s to meet AISC sequence requirements, and update the remaining resources of every edge server.

4.3 Algorithm analysis

In this section, we first prove the feasibility of SRON (algorithm 1). Then, the competitive ratio and time complexity of SRON are analyzed.

Theorem 2. *In a resource-constrained EIC network $G_s = (V \cup C, E^s)$, in which C represents edge servers with computing capacity cap_c, V is a set of APs without computing capacity. Given a set of links e with security levels S_e, and request $\gamma = (L^\gamma, F^\gamma, R^\gamma, S^\gamma, A^\gamma)$ arrive one by one without any future information, there is an online algorithm 1 (SRON) with 3 competitive ratio providing a feasible solution in $O(|A_{rr}| \cdot (V^{s2} + \log_{1-r_{f_{min}}} 1 - R^{\gamma_{max}} + |F^{\gamma_{max}}|\,|C|\,log\,|F^{\gamma_{max}}|))$ time for SRAPON problem, where $r_{f_{min}}$ is the lowest reliability in all VNFs, $|F^{\gamma_{max}}|$ and $R^{\gamma_{max}}$ are the maximum number of VNFs and the highest reliability in all requests, respectively.*

Proof. Firstly, the feasibility of algorithm 1 is proved as follows. In terms of security, we use physical isolation to remove links and servers whose values are lower than threshold by constructing an auxiliary graph. In terms of accuracy, appropriate VNF models are selected. In terms of reliability, iteratively providing BVNFs to the least reliable VNF in each unreliable AISC prevents it from faults. In addition, servers are guaranteed not to violate their capacities during the backup process. In other words, f_i and f_{ib}^m can be deployed only if their required resources are less than or equal to the remaining resource of c.

Then, the competitive ratio of SRON is analyzed. The deployment cost of A_{rr} is no more than 1 + α' times of the optimal cost ($\alpha' = 2$) [23]. Finally, its time complexity is analyzed. It takes $O\left(2 \cdot (V^s)^2\right)$ to construct the auxiliary graph G_{S^γ} and to find all trusted connected components in G_{S^γ} for a request. If AISC reliability is less than the requested threshold, iteratively providing BVNFs would consume $O(\log_{1-r_{f_{min}}} 1 - R^{\gamma_{max}})$. The solution $\mathbb{S}$ obtained by algorithm [23] takes $O(|F^{\gamma_{max}}|\,|C|\,log\,|F^{\gamma_{max}}|)$. Checking solution takes $O(|F^{\gamma_{max}}|)$. In addition, these operations are repeated $O(|A_{rr}|)$ times. Therefore, the total time complexity of algorithm 1 is $O(|A_{rr}| \cdot (V^{s2} + \log_{1-r_{f_{min}}} 1 - R^{\gamma_{max}} + |F^{\gamma_{max}}|\,|C|\,log\,|F^{\gamma_{max}}|))$. □

5 OFF-SITE SCENARIO

In this section, we consider a more complex online backup off-site scenario. That is, for a given set of AISC requests, the VNFs in each AISC must be deployed on different edge servers, and VNFs and corresponding BVNFs must be also deployed on different servers. The scenario effectively alleviates the fault of VNFs running on a single edge server due to server malfunctions or attacks. However, the link securities between VNFs and corresponding BVNFs need to be considered additionally.

5.1 SROFF algorithm

The SROFF algorithm (Algorithm 2) for the SRAP problem under an off-site scenario is described as follows.

Step 1. In the off-site scenario, we first sort a set of new arrival requests in ascending order based on security demands, and merge the requests with the same demand into one group. Instead of constructing auxiliary graphs one by one for each request, we construct one for the request group only based on the security to reduce the running time of the SROFF algorithm. Note that the graphs may have untrusted servers.

Algorithm 2: (SROFF) Security-aware and reliability-guaranteed algorithm under off-site scenario

Input: The resource-limited EIC network $G_s = (V^s, E^s)$ with computing capacity C_v, and a given set of requests A_{rr} without any future information
Output: a set of receiving requests A_{cc}

begin
  $sl = -1, BN = \Lambda = \emptyset, A_{cc} = A_{rr}$;
  Sort requests γ in A_{rr} based on its required security level S^{γ};
  for *each* γ_k *in this order* **do**
    if $sl < S^{\gamma_k}$ **then**
      $sl \leftarrow S^{\gamma_k}$;
      Construct auxiliary $G_{sl} = (V^s, E^{sl})$ only based sercurity level sl for links;
    Calculate the reliability $\mathbb{R}^{\gamma_k}$ of γ_k based (2);
    while $\mathbb{R}^{\gamma_k} < R^{\gamma_k}$ **do**
      $i \leftarrow \min_{f_i \in \gamma} r_{f_i}$, $BN_i^{\gamma_k}$ += 1, update $\hat{r}_{f_i}$, $\hat{\mathbb{R}}^{\gamma_k}$ based (3) and (4);
      $\mathbb{R}^{\gamma_k} \leftarrow \hat{\mathbb{R}}^{\gamma_k}$;
    $\beta = \left\{ \left| F^{\gamma_k} \right|, \max_{f_i \in \gamma_k} BN_i^{\gamma_k} \right\}_{max} + 1$;
    Calculate the trust value of all edge servers based on the formula (8), (9), (10), (11), and delete servers whose trusts are lower than T_{thr};
    Select a connected component Λ that is closest to the user, and contains the most trusted server numbers $\left|C^{\Lambda}\right|$ in G_{sl};
    if $\left|C^{\Lambda}\right| < |\beta|$ **then**
      Reject request γ_k, $A_{cc} \leftarrow A_{cc} \backslash \gamma_k$;
    sort C^{Λ} in ascending based on (19), merge the top-β c into $\Lambda_{k'} = \{\lambda_1 = c_i, ..., \lambda_\beta = c_j\}$;
    Select model of each VNF in γ_k based on (14);
    Pre-deploy VNFs and BVNFs into ordered $\Lambda_{k'}$ according to the inference sequence of AISC, and and record the corresponding deployment location in C_{s_i}. // Note that the same type (B)VNFs cannot be placed in the same edge server c, and positions with the smallest hops and the closest distance from VNF are selected first;
    if *there is one or more edge servers that cannot host (B)VNFs due to insufficient computing resources or* $\sum_{c \in \Lambda_{k'}} \phi_c^k \geq |C|\, S^{\gamma_k}$ **then**
      Reject request γ_k, $A_{cc} \leftarrow A_{cc} \backslash \gamma_k$;
    else
      Deploy γ_k based on AISC order and C_{s_i};
      Update resources and $\bar{\rho}_{f,c}$;
  return A_{cc};

Step 2. The minimum amount BN of BVNFs required by VNFs in each unreliable AISC is checked and calculated based on equation (2), (3), (4). β is denoted by the near-optimal number of trusted servers based on deployment location constraints of SROFF. The untrusted servers for a user are calculated and removed based on the formula (8), (9), (10), (11). If the number of trusted servers is less than β in the closest and largest connected component Λ, request will be rejected due to insufficient deployment location requirement. Otherwise, sort edge servers in ascending order based on the normalized cost (19), and merge top-β trusted servers into $\Lambda_{k'}$.

Step 3. Model size of each VNF is selected based on the corrected data quality of each e in $\Lambda_{k'}$ and accuracy demand A^{γ_k}. Then, we pre-deploy VNFs and BVNFs required by γ_k into $\Lambda_{k'}$ successively according to deployment location constraints and AISC sequence demand. Note that the number of hops is used as the primary parameter of distance during selecting deployment locations, but when the number of hops is the same, the actual distance is used as the auxiliary judgment.

Step 4. In order to ensure that the SROFF algorithm has a competitive ratio under an off-site scenario, we introduce a receiving control policy. If the sum of the normalized cost of VNFs and BVNFs is no greater than a given threshold $|C|\, S^{\gamma_k}$, the request is accepted and deployed, otherwise rejected. In addition, if servers in $\Lambda_{k'}$ with insufficient computing capacity to accommodate VNFs and BVNFs, the request would also be rejected. Finally, deploy AISCs and update edge server resources.

5.2 Algorithm analysis

In this section, we first show an upper bound cost of all edge servers after deploying requests A_{cc}. Then, a lower bound of the difference between an actual solution and an optimal solution is proved. Finally, the competitive ratio and time complexity of SROFF are analyzed.

Lemma 1. *Denote β as the 1-approximate number of requested servers based on SROFF deployment location constraints. That is, $\beta \leq N_{opt} + 1$.*

Proof. N_{opt} indicates the optimal number of edge servers occupied by VNFs and BVNFs in an AISC request.

$Case$ 1: When $|F^{\gamma_k}| \leq \max_{f_i \in \gamma_k} BN_i^{\gamma_k}$. VNFs and corresponding BVNFs cannot run on the same server. So, at least $N_{opt} = \max_{f_i \in \gamma_k} BN_i^{\gamma_k} + 1$ edge servers are required.

$$\beta = \left\{ \left| F^{\gamma_k} \right|, \max_{f_i \in \gamma_k} BN_i^{\gamma_k} \right\}_{max} + 1 = \max_{f_i \in \gamma_k} BN_i^{\gamma_k} + 1 = N_{opt}$$

$Case$ 2: When $|F^{\gamma_k}| > \max_{f_i \in \gamma_k} BN_i^{\gamma_k}$. The VNFs in each AISC must be deployed on different edge servers. So, at least $N_{opt} = |F^{\gamma_k}|$ edge servers are required.

$$\beta = \left\{ \left| F^{\gamma_k} \right|, \max_{f_i \in \gamma_k} BN_i^{\gamma_k} \right\}_{max} + 1 = |F^{\gamma_k}| + 1 \leq N_{opt} + 1$$

Therefore, β is only one more than N_{opt}. □

Lemma 2. *Given an EIC network $G_s = (V \cup C, E^s)$, where each edge server $v \in C$ with computing capacity C_v. A set of receiving requests A_{cc} is obtained by algorithm 2. When requests A_{cc} arrive, the total deployment cost of all v is:*

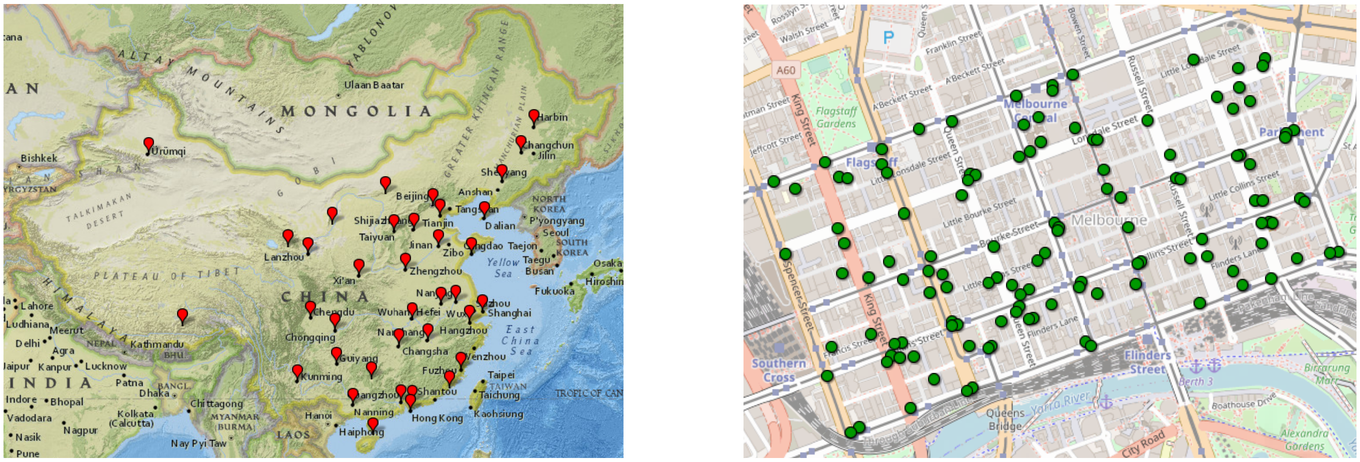
$$\sum_{v \in C, k \in A_{cc}} \psi_v^k \leq 8\,|C| \cdot C_{v_{max}} \cdot \sum_{1 \leq k \leq |A_{cc}|} (S^{\gamma_k} + 1) \quad (37)$$

Where α is a constant with $e\,|C| + 2 \leq \alpha \leq e^{\frac{2C_v}{\xi_{max} C_{f_{max}}}}$ that reflects the workload sensitivity on each server, $\xi_{max} = \lceil \frac{\sum_{i=1}^{|F^{\gamma_k}|} (BN_i^{\gamma_k} + 1)}{\beta} \rceil$ is the maximum number of VNFs and BVNFs running on a single server for each request, and $C_{f_{max}}$ represents the maximum computing capacity cap_f in all VNFs.

The proof can be found in Appendix A.

Lemma 3. *Let AISC request $\gamma_{k''} \in J$ is rejected by SROFF but received by the optimal algorithm. Denote $\Lambda_{k''}^{opt}$ as the set of edge servers selected by the optimal algorithm to host VNFs and BVNFs in $\gamma_{k''}$. For any request $\gamma_{k''} \in J$, the lower bound on deployment cost gap is:*

$$\sum_{v \in \Lambda_{k''}^{opt}} \phi_v^{k''} \geq |C|\, S^{\gamma_{k''}} \quad (38)$$



(a) The CERNET

(b) The Melbourne CBD

Fig. 3: Real network topologies.

Where α is a constant with $e|C|+2 \le \alpha \le e^{\frac{2C_v}{\xi_{max} C_{f_{max}}}}$ that reflects the workload sensitivity on each server, $\xi_{max} = \lceil \frac{\sum_{i=1}^{|F^{\gamma_k}|}(BN_i^{\gamma_k}+1)}{\beta} \rceil$ is the maximum number of VNFs and BVNFs running on a single server for each request, and $C_{f_{max}}$ represents the maximum computing capacity in all VNFs.

The proof can be found in Appendix B.

Theorem 3. *Given a resource-constrained EIC network and a set of new arrival requests, the competitive ratio of online algorithm 2 (SROFF) is a constant, namely, 2 or $C_{v_{max}} S_{max}$.*

The proof can be found in Appendix C.

Theorem 4. *Given a resource-constrained EIC network and a set of new arrival requests, the time complexity of online algorithm 2 (SROFF) is $O(|A_{rr}| \cdot (|A_{rr}| \log |A_{rr}| + \log_{1-r_{f_{min}}} 1 - R^{\gamma_{max}} + C^2))$, where $r_{f_{min}}$ is the lowest reliability in all VNFs, $|F^{\gamma_{max}}|$ and $R^{\gamma_{max}}$ are the maximum number of VNFs and the highest reliability threshold in all requests, respectively.*

Proof. The time complexity of SROFF is analyzed as follows. Sorting a set of arrival requests takes $O(|A_{rr}| \log |A_{rr}|)$. We consume $O(\log_{1-r_{f_{min}}} 1 - R^{\gamma_{max}})$ to check and ensure AISC reliability. It takes $O(C^2)$ and $O(|C^\Lambda| \log |C^\Lambda|)$ to select the closest and largest connected component and to sort it based on (19), respectively. Deploying VNFs and BVNFs into ordered servers successively according to the deployment location constraints takes $O(|C^\Lambda| \beta |F^{\gamma_{max}}|)$. In addition, these operations are repeated $O(|A_{rr}|)$ times for all requests. Therefore, the total time complexity of algorithm 2 is $O(|A_{rr}| \cdot (|A_{rr}| \log |A_{rr}| + \log_{1-r_{f_{min}}} 1 - R^{\gamma_{max}} + C^2))$. □

5.3 VNF consolidation algorithm

The VNF consolidation algorithm (algorithm 3) is designed for scenarios requiring VNF and BVNF sharing. Because each user has different security and accuracy requirements, it is not reasonable to simply consolidate VNFs from multiple servers into a VNF running on a single server [59]. This may result in unroutable or inaccurate situations. To avoid this, algorithm 3 only consolidates each type of VNF on each server, and reserves the suitable number of VNF models with the largest accuracy (lines 8-11). In addition, we also give the time complexity proof of algorithm 3 at last.

Algorithm 3: VNF consolidation

Input: The deployment location C_s for each receiving request in A_{cc}, and the threshold T_{sn} of sharing number.
Output: AISC data routing and VNF sharing scheme.

```
1  begin
2      z_{i,j} = [], zb_{i,j} = [] // Number of VNF f_i and BVNF f_ib in
           server c_j, respectively;
3      N_v r = 0 // Number of VNF to be reserved ;
4      for each γ_k in the A_cc do
5          According to the deployment location C_s, the route among
               VNF and BVNF is established in AISC order by the Dijkstra
               shortest path algorithm;
6          Update z_{i,j} and zb_{i,j} based on C_s and the number and type
               of VNF and BVNF required by γ_k;
7      for each edge server c_j do
8          for each type VNF F_i do
9              if z_{i,j} ≥ 2 then
10                 N_vr = ⌈z_{i,j}/T_sn⌉ // Number of VNF to be
                       reserved;
11                 Retain top-N_vr VNFs that have maximum model
                       accuracy, and release the occupied resources by
                       others;
12         (Optional) Perform lines 9-11 for each type of BVNF.
```

Theorem 5. *Given a resource-constrained EIC network $G_s = (V^s, E^s)$, the deployment location C_s for each receiving request in A_{cc}, and the number $|F|$ of all VNF types, the time complexity of algorithm 3 is $O((|V^s|^2 + |V^s||F|) \cdot |A_{cc}|)$.*

Proof. The time complexity of the VNF consolidation algorithm is analyzed as follows. It takes $O((|V^s|^2 + |F^{\gamma_{max}}|) \cdot |A_{cc}|)$ time to create routes and update share numbers for a set of requests. In addition, merging the same type of VNFs running on all edge servers takes $O(|V^s||F||A_{cc}|)$ time. Therefore, the total time complexity of algorithm 3 is $O((|V^s|^2 + |V^s||F|) \cdot |A_{cc}|)$. □

6 Performance Evaluation

In this section, we first detail the experimental settings and benchmark experiments. Then, the influences of relevant important parameters on SRON and SROFF algorithms are

evaluated based on the real network topologies. All the above experiments are conducted on a personal machine with a 2.30GHz Intel(R) Core(TM) I7-10875 CPU and 16GB of RAM.

6.1 Environment setting

We regard two real network topologies with different scales as EIC networks $G_s = (V^s, E^s)$, as shown in Fig. 3. In terms of a smaller scale, the China Education and Research Network (CERNET) is considered, which consists of 37 physical nodes. In terms of a larger scale, the Melbourne Central Business District (CBD) Network[5] [60] is selected, which is composed of 125 nodes. we randomly select 50% of the network size access points (APs) to coexist with the edge server. The computing capacity of the edge server ranges from 2000 MHz to 4000 MHz [4], and the security level of links ranges from 1 to 6 [14]. The specific link security parameters are as follows: let the packet loss rate and hash error rate range from 0 to 0.2. Data integrity rate S_{di}^e ranges from 0.64 to 1. The availability protection mechanisms of a link range from [1,6]. Security authentication $[S_u \cap S_v \neq \varnothing]$ between servers ranges from 0 to 1. The network topologies are based on the latitude and longitude of the real world, so the link delay D_e is calculated by $D_e = distance/3.0 \times 10^5$, where $3.0 \times 10^5 km/s$ is the propagation rate of electromagnetic waves in free space.

In addition, the EIC networks also provide 50 types of VNF, including basic network functions (such as firewall, deep intrusion detection, and data encryption) and AI inference services (such as natural language processing, face recognition, and medical image processing). The computing capacity of VNFs ranges from 10 MHz to 150 MHz [6], and VNF reliabilities range from 0.8 to 0.99. The model accuracy of AI-based VNF ranges from 0.6 to 1 (default value is 1). In addition, according to the data in article [61], the proportional coefficient between model size and model accuracy is set to 1. For each location-random AISC request that is interconnected by $1 \sim 5$ types of VNFs in a specific order, whose reliability demand ranges from 0.8 to 0.95, security demand ranges from 1 to 6, and accuracy demand ranges from 1 to 6, respectively. Let the security threshold T_{thr} be 0.4 [48]. The workload sensitivity factor α on each server is $e\,|C| + 2$, and the number of new arrival requests is 150. The above values are default values for each trial unless otherwise specified. All data is an average of 50 trials.

6.2 Benchmark

The performances of our algorithms (SRON and SROFF) are compared with three other algorithms in terms of average deployment cost, throughput, running time, and server workload.

SRON and SROFF algorithms: Online algorithms solve the SRAPON and SRAPOFF problems respectively in polynomial time.

5. https://github.com/swinedge/eua-dataset

6. Since AI-based VNFs require more computing resources than traditional ones and each edge server has limited resources, the optimal throughput of the entire network decreases as the number of AI-based VNFs increases.

SRON_ILP and SROFF_ILP algorithms: They provide the lowest deployment cost of instantiating VNFs and BVNFs by using the DOcplex optimizer in Python to solve the SRAP problem under on-site and off-site backup scenarios.

SROFF_WT algorithm: It also uses the DOcplex optimizer but does not use the receiving control policy, thus accepting more requests at a lower deployment cost.

RAD algorithm: Shang [12] proposed a RAD scheme for reliable VNF provisioning problems without considering security parameters. They first provided a BVNF for each VNF in arrival requests, and then, BVNFs were provided dynamically in real-time for unreliable requests. For comparison, we modified it to take into account security and reliability (formulas are the same as in this paper) under the off-site backup scenario.

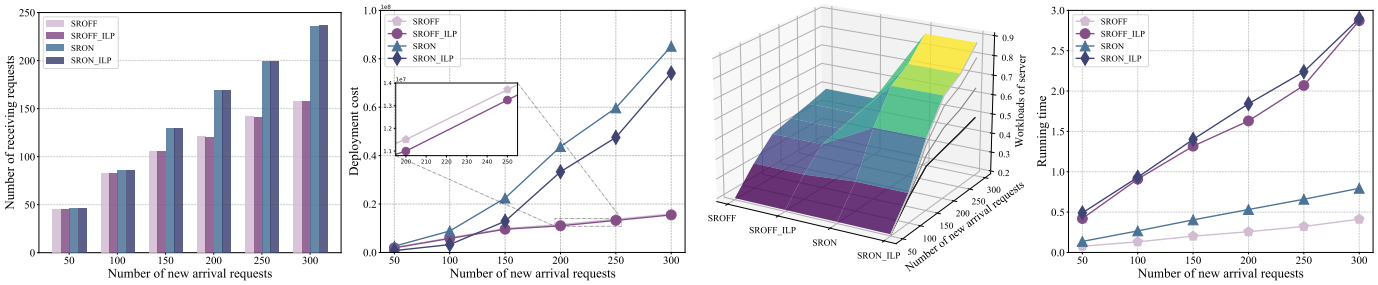
6.3 Performance evaluation

6.3.1 Impact of the number of new arrival requests on performances of different algorithms

We first evaluate different algorithms by varying the number of arrival AISC requests from 50 to 300 in CERNET. Then, the request numbers are raised from 300 to 1300 in Melbourne CBD. Other parameters remain unchanged. Fig. 4(a) shows that SRON (SRON_ILP) receives more AISC requests than SROFF (SROFF_ILP). This is because the deployment location constraints are stricter in off-site than in on-site scenarios. The deployment costs of different schemes are going up, but the rising levels of SRON and SRON_ILP are much higher than that of SROFF and SROFF_ILP in Fig. 4(b). The cost of SRON is not more than 2.729 times the optimal cost, and the SROFF cost does not exceed 1.046 times. The reason for these changes is that VNFs and corresponding BVNFs need to be deployed on the same edge server in the on-site scenario, resulting in higher server workloads and costs than off-site. Fig. 4(c) also illustrates that the server workloads in the on-site scenario are higher than in the off-site, and rise as the number of receiving requests increases. Fig. 4(d) shows that the running time of ILP algorithms is much longer than online algorithms (SRON and SROFF). In addition, the running time of SROFF is shorter than that of SRON because the total construct times of the auxiliary graph are significantly reduced. The above phenomena are also shown in Fig. 10.

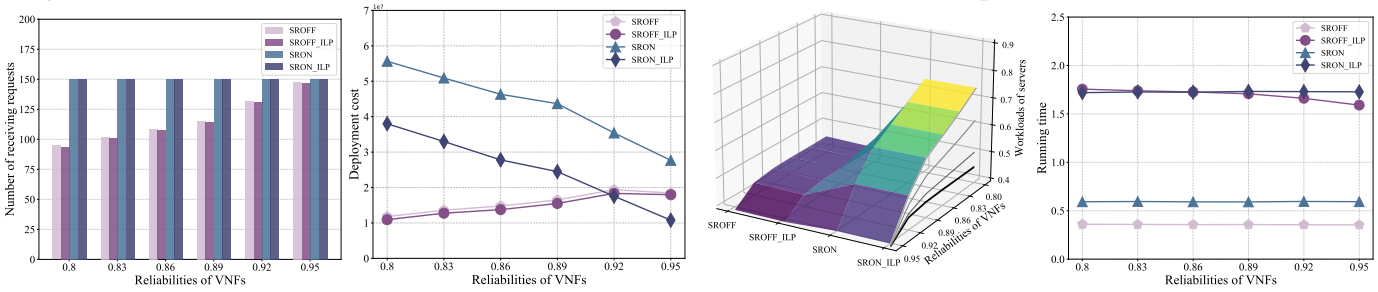
6.3.2 Impact of reliabilities of VNFs on performances of different algorithms

We investigate the impact of VNF reliabilities on the performances of different algorithms by varying the reliabilities from 0.8 to 0.95 in CERNET, while other parameters are not changed. A clear trend in Fig. 5(a) is that the throughput of SROFF and SROFF_ILP algorithms grows as VNF reliabilities increase, while the throughput of SRON and SRON_ILP algorithms remain stable. This is because the VNF reliabilities are inversely proportional to the number of BVNFs required by VNFs, and each edge server can accommodate more requests as reliability increases. Fig. 5(b) shows two distinct trends of deployment cost, because the cost is related not only to BVNF numbers but also to receiving request numbers. As VNF reliabilities increase, the



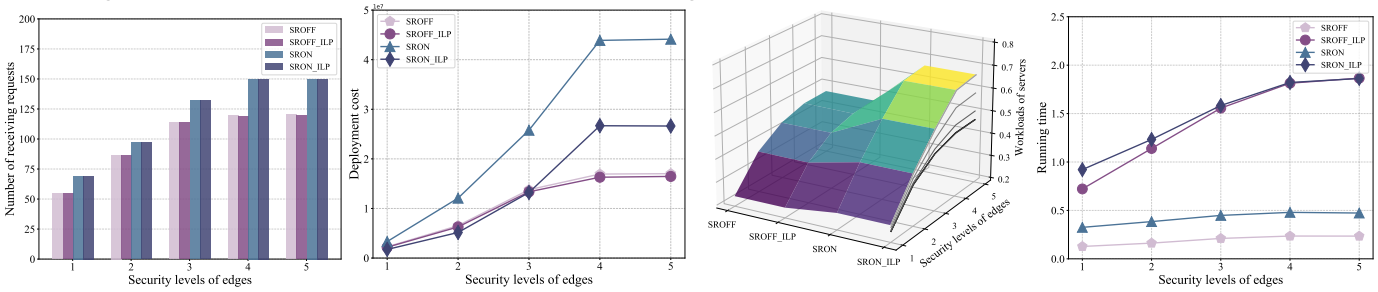
(a) The number of receiving requests (b) The deployment cost (c) The workloads of servers (d) The running time

Fig. 4: Performances of different algorithms by varying the number of new arrival requests from 50 to 300 in CERNET.



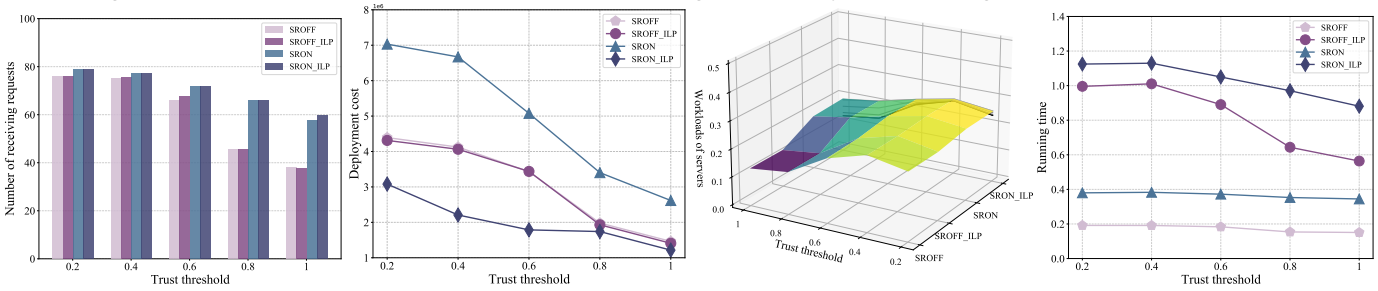
(a) The number of receiving requests (b) The deployment cost (c) The workloads of servers (d) The running time

Fig. 5: Performances of different algorithms by varying the reliabilities of VNFs from 0.8 to 0.95 in CERNET.



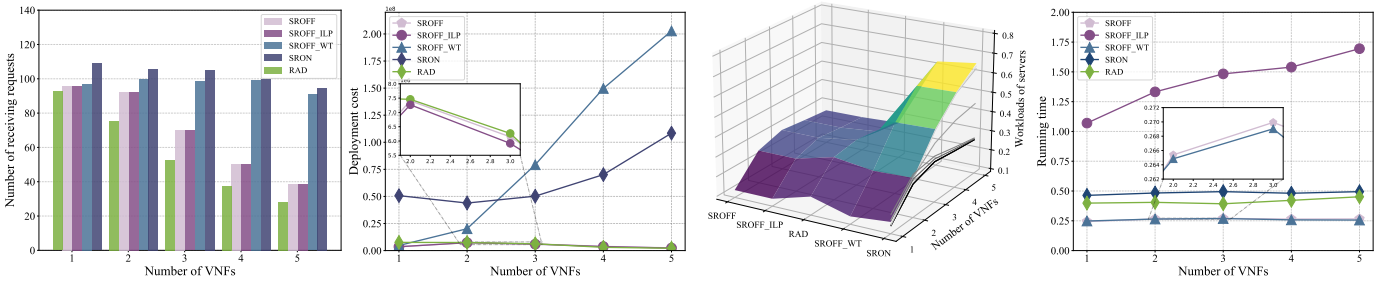
(a) The number of receiving requests (b) The deployment cost (c) The workloads of servers (d) The running time

Fig. 6: Performances of different algorithms by varying the security levels of edges from 1 to 5 in CERNET.



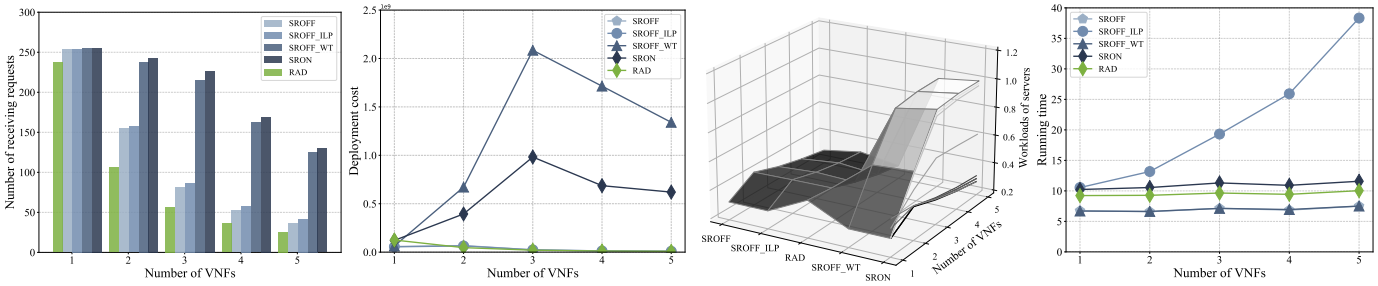
(a) The number of receiving requests (b) The deployment cost (c) The workloads of servers (d) The running time

Fig. 7: Performances of different algorithms by varying the trust threshold from 0.2 to 1 in CERNET.



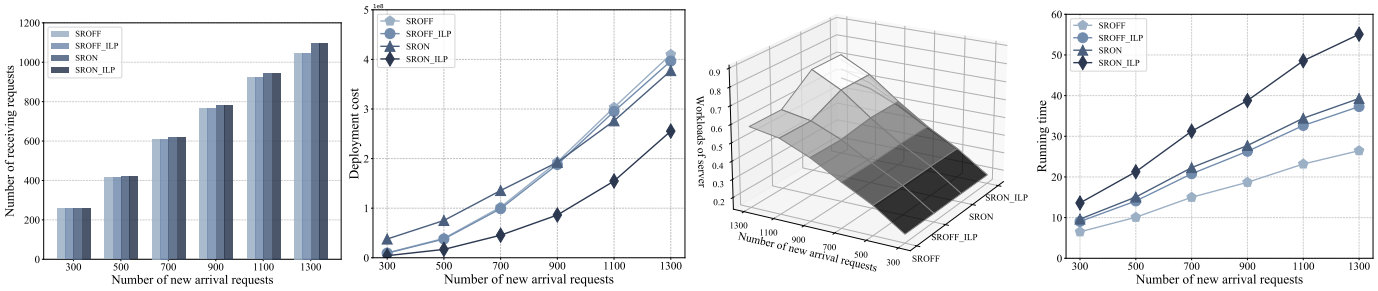
(a) The number of receiving requests (b) The deployment cost (c) The workloads of servers (d) The running time

Fig. 8: Performances of different algorithms by varying the number of VNFs from 1 to 5 in CERNET.



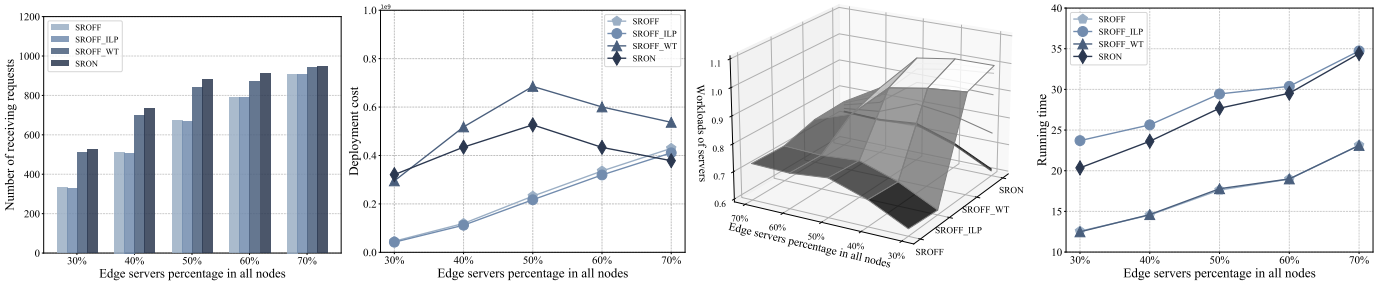
(a) The number of receiving requests (b) The deployment cost (c) The workloads of servers (d) The running time

Fig. 9: Performances of different algorithms by varying the number of VNFs from 5 to 25 in CBD.



(a) The number of receiving requests (b) The deployment cost (c) The workloads of servers (d) The running time

Fig. 10: Performances of different algorithms by varying the number of new arrival requests from 300 to 1300 in CBD.



(a) The number of receiving requests (b) The deployment cost (c) The workloads of servers (d) The running time

Fig. 11: Performances of different algorithms by varying the percentage of edge servers from 30% to 70% in CBD.

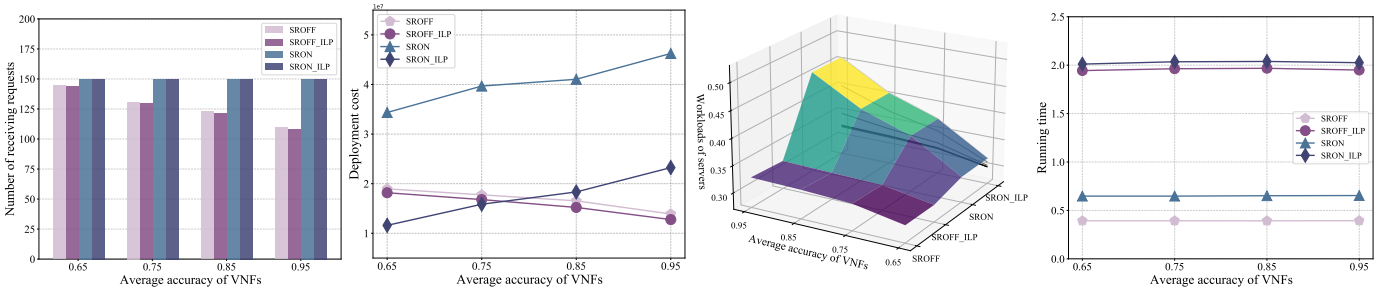
costs of SROFF and SROFF_ILP grow due to the increasing number of receiving requests, while the costs of SRON and SRON_ILP reduce due to the decreasing number of BVNFs. It can also be seen from workloads in Fig. 5(c). Fig. 5(d) shows that the running time of different algorithms decreases slightly as reliability rises.

6.3.3 Impact of security levels of edges on performances of different algorithms

We study the effect of link security levels on the performances of four algorithms by increasing the security from 1 to 5 in CERNET, while security demand ranges from 1 to 4, and other parameters remain unchanged. Fig. 6 shows that the rise of all performances lasted for security level 4 and then began to level off in 5. This is because when the lowest link security level is higher than the maximum user requirement, it is no longer a bottleneck restricting performance. Fig. 6(a), 6(b) and 6(c) show that the number of accepting requests, deployment cost, and server workloads increase with the improvement of link security level, where the cost of SRON can achieve 1.658 times of the optimal solution. From 6(d), the running time of algorithms in on-site scenarios is generally longer than those in off-site due to the gap in the number of constructing auxiliary graphs. In addition, the running time of ILP algorithms is much longer than online algorithms due to the increase of receiving requests, thus SRON_ILP is the longest and SROFF is the shortest.

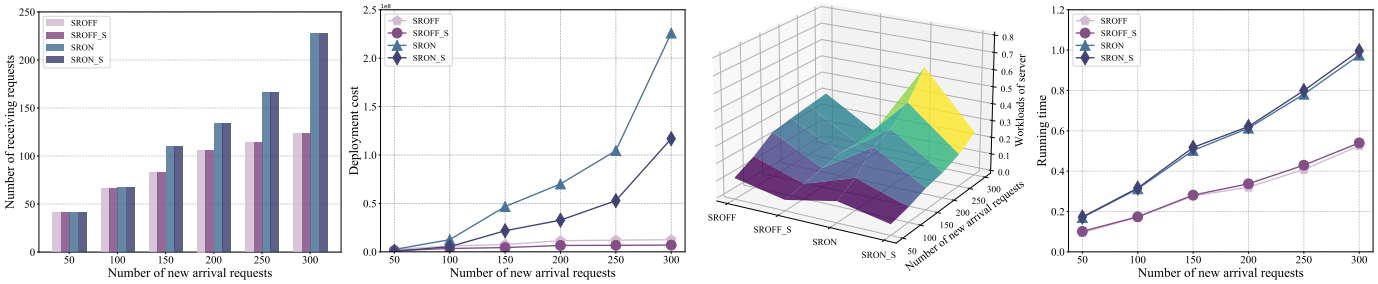
6.3.4 Impact of number of VNFs in an AISC on performances of different algorithms

The effect of the number of VNFs in an AISC on performances of SRON, SROFF, SROFF_ILP, SROFF_WT, and RAD algorithms by changing the number from 1 to 5 in CERNET, while other parameters are not changed. In Fig. 8, the RAD performance is worse than that of our algorithms in the off-site scenario due to an improper backup selection mechanism. Specifically, the RAD scheme receives the fewest requests, but has a slightly higher deployment cost than SROFF and SROFF_ILP, and a higher uptime than SROFF. Fig. 8(a) shows that the throughput of the EIC system decreases gradually with the growth of AISC request numbers. SROFF_WT accepts slightly fewer requests than SRON, but more than SROFF and SROFF_ILP, due to the removal of the receiving control policy. From Fig. 8(b), two opposite patterns are presented that SRON and



(a) The number of receiving requests (b) The deployment cost (c) The workloads of servers (d) The running time

Fig. 12: Performances of different algorithms by varying the average accuracy of VNFs from 0.65 to 0.95 in CERNET.



(a) The number of receiving requests (b) The deployment cost (c) The workloads of servers (d) The running time

Fig. 13: Performances of sharing algorithms by varying the number of new arrival requests from 50 to 300 in CERNET.

SROFF_WT generally improve with the raising of VNF numbers, whereas SROFF and SROFF_ILP grow first and then decrease. The reason is that deployment cost is related to the number of VNFs and BVNFs contained in all requests. Specifically, total costs are reduced when the reduction in accepting request numbers is greater than the growth in VNF numbers, and vice versa. A similar trend of server workloads is found in Fig. 8(c). As can be seen from Fig. 8(d) that the running time of SROFF_WT is shorter compared with SROFF because receiving judgment is not required. Then, the VNF numbers are raised from 5 to 25 in Melbourne CBD where arrival requests are set to 300, and the above phenomena are also shown in Fig. 9.

6.3.5 Impact of other parameters on performances of different algorithms

The impact of trust thresholds on performances of different algorithms is first evaluated in CERNET, by increasing the thresholds from 0.2 to 1, while other parameters remain unchanged. Fig. 7 shows that all performances drop as trust thresholds climb. When the thresholds increase, the number of trusted servers declines, which incurs decreases in the total receiving number.

Then, we study the effect of edge server percentage in all nodes on performances of different algorithms in Melbourne CBD, by increasing the percentage from 30% to 70% and setting arrival requests to 1100. The throughput and runtime of algorithms climb as total numbers of edge servers increase, as shown in Fig. 11(a)(d). However, Fig. 11(b) shows a trend of deployment costs of SRON and SROFF_WT rising and then falling. The reason behind this can be found in Fig. 11(c). Specifically, workloads are one of the main cost parameters, and the workloads on each server would drop when the total server number increases. The cost of SROFF and SROFF_ILP rises because receiving requests continue to increase while server workloads remain stable.

Next, we study the effect of average VNF accuracy on the performances of different algorithms in CERNET by varying the accuracy range, namely [0.6, 0.7], [0.7, 0.8], [0.8, 0.9], [0.9, 1], while other parameters remain unchanged. For SROFF_ILP and SROFF, Fig. 12(a), (b) show that the number of receiving requests reduces as the average accuracy of VNF increases, resulting in reduced deployment cost. For SRON_ILP and SRON, Fig. 12(a), (b), (c) show that server workload increases with the model accuracy, which increases the deployment cost when the number of receiving requests is constant.

Finally, we study the effect of VNF sharing on performances of SRON and SROFF algorithms in CERNET, by increasing arrival requests from 50 to 300. Fig. 13 shows that as the number of requests increases, the algorithms using the VNF consolidation mechanism (SROFF_S and SRON_S) can save more resources (over 50%) by sacrificing only a little extra time (no more than 6.5%).

7 Conclusion

EIC is gradually becoming the next stage of mobile edge cloud, whose reliability and security are extremely significant but easily overlooked. To this end, we first study AISCs provisioning in resource-limited and vulnerable EIC from a new perspective of antagonistic effects among reliability, security, and accuracy, aiming to maximize the throughput of receiving requests while minimizing deployment cost. Then, the problem is formulated as ILP under on-site

and off-site backup scenarios. Two online algorithms with constant competitive ratios are proposed for solving the problem in polynomial time, respectively. Finally, extensive theoretical analyses and experiments are conducted on real network topologies. The results show that our algorithms are promising in insecure and unreliable EIC networks.

REFERENCES

[1] Z. Lin, S. Bi, and Y.-J. A. Zhang, "Optimizing ai service placement and resource allocation in mobile edge intelligence systems," *IEEE Transactions on Wireless Communications*, vol. 20, no. 11, pp. 7257–7271, 2021.

[2] M. Li, J. Gao, C. Zhou, X. S. Shen, and W. Zhuang, "Slicing-based artificial intelligence service provisioning on the network edge: Balancing ai service performance and resource consumption of data management," *IEEE Vehicular Technology Magazine*, vol. 16, no. 4, pp. 16–26, 2021.

[3] Y. Han, X. Wang, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, pp. 869–904, 2020.

[4] M. Huang, W. Liang, X. Shen, Y. Ma, and H. Kan, "Reliability-aware virtualized network function services provisioning in mobile edge computing," *IEEE Transactions on Mobile Computing*, vol. 19, no. 11, pp. 2699–2713, 2019.

[5] Y. Ma, W. Liang, M. Huang, W. Xu, and S. Guo, "Virtual network function service provisioning in mec via trading off the usages between computing and communication resources," *IEEE Transactions on Cloud Computing*, pp. 1–1, 2020.

[6] J. Liang and F. Tian, "An online algorithm for virtualized network function placement in mobile edge industrial internet-of-things," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2022.

[7] K. Liu, W. Quan, N. Cheng, X. Zhang, L. Guo, D. Gao, and H. Zhang, "Deadline-constrained multi-agent collaborative transmission for delay-sensitive applications," *IEEE Transactions on Cognitive Communications and Networking*, 2023.

[8] F. Tian, X. Zhang, J. Liang, and Z. Yang, "Bidirectional service function chain embedding for interactive applications in mobile edge networks," *IEEE Transactions on Mobile Computing*, pp. 1–17, 2023.

[9] M. Mukherjee, R. Matam, C. Mavromoustakis, H. Jiang, G. Mastorakis, and M. Guo, "Intelligent edge computing: Security and privacy challenges," *IEEE Communications Magazine*, vol. 58, pp. 26–31, 2020.

[10] Z. Fu, J. Yang, C. Bai, X. Chen, C. Zhang, Y. Zhang, and D. Wang, "Astraea: Deploy ai services at the edge in elegant ways," in *2020 IEEE International Conference on Edge Computing (EDGE)*, 2020, pp. 49–53.

[11] Z. Xu, L. Zhao, W. Liang, O. F. Rana, P. Zhou, Q. Xia, W. Xu, and G. Wu, "Energy-aware inference offloading for dnn-driven applications in mobile edge clouds," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 4, pp. 799–814, 2021.

[12] X. Shang, Y. Huang, Z. Liu, and Y. Yang, "Reducing the service function chain backup cost over the edge and cloud by a self-adapting scheme," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2021.

[13] S. Lal, T. Taleb, and A. Dutta, "Nfv: Security threats and best practices," *IEEE Communications Magazine*, vol. 55, pp. 211–217, 2017.

[14] P. Zhang, C. Jiang, X. Pang, and Y. Qian, "Stec-iot: A security tactic by virtualizing edge computing on iot," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2459–2467, 2020.

[15] S. Liu, Z. Cai, H. Xu, and M. Xu, "Towards security-aware virtual network embedding," *Comput. Networks*, vol. 91, pp. 151–163, 2015.

[16] B. Yi, X. Wang, K. Li, S. k. Das, and M. Huang, "A comprehensive survey of network function virtualization," *Computer Networks*, vol. 133, pp. 212–262, 2018.

[17] F. L. de Mello, "A survey on machine learning adversarial attacks," *Journal of Information Security and Cryptography (Enigma)*, vol. 7, no. 1, pp. 1–7, 2020.

[18] X. Xian, T. Wu, S. Qiao, W. Wang, C. Wang, Y. Liu, and G. Xu, "Deepec: Adversarial attacks against graph structure prediction models," *Neurocomputing*, vol. 437, pp. 168–185, 2021.

[19] K. Liu, W. Quan, N. Cheng, W. Wu, Z. Xu, L. Guo, D. Gao, and H. Zhang, "Reliable ppo-based concurrent multipath transfer for time-sensitive applications," *IEEE Transactions on Vehicular Technology*, 2023.

[20] J. Li, W. Liang, M. Huang, and X. Jia, "Reliability-aware network service provisioning in mobile edge-cloud networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, pp. 1545–1558, 2020.

[21] R. Doriguzzi-Corin, S. Scott-Hayward, D. Siracusa, M. Savi, and E. Salvadori, "Dynamic and application-aware provisioning of chained virtual security network functions," *IEEE Transactions on Network and Service Management*, vol. 17, pp. 294–307, 2020.

[22] R. Nauss, "Solving the generalized assignment problem: An optimizing and heuristic approach," *INFORMS J. Comput.*, vol. 15, pp. 249–266, 2003.

[23] R. Cohen, L. Katzir, and D. Raz, "An efficient approximation for the generalized assignment problem," *Information Processing Letters*, vol. 100, no. 4, pp. 162–166, 2006.

[24] B. Han, V. Gopalakrishnan, G. Kathirvel, and A. Shaikh, "On the resiliency of virtual network functions," *IEEE Communications Magazine*, vol. 55, pp. 152–157, 2017.

[25] M. Herker and A. Khan, "Survey on survivable virtual network embedding problem and solutions," 2013.

[26] D. Cotroneo, L. Simone, A. K. Iannillo, A. Lanzaro, R. Natella, J. Fan, and W. Ping, "Network function virtualization: Challenges and directions for reliability assurance," *2014 IEEE International Symposium on Software Reliability Engineering Workshops*, pp. 37–42, 2014.

[27] L. Bays, R. R. Oliveira, L. Buriol, M. Barcellos, and L. Gaspary, "Security-aware optimal resource allocation for virtual network embedding," *2012 8th international conference on network and service management (cnsm) and 2012 workshop on systems virtualiztion management (svm)*, pp. 378–384, 2012.

[28] D. Dwiardhika and T. Tachibana, "Optimal construction of service function chains based on security level for improving network security," *IEEE Access*, vol. 7, pp. 145 807–145 815, 2019.

[29] M. Bagaa, T. Taleb, J. B. Bernabé, and A. Skarmeta, "Qos and resource-aware security orchestration and life cycle management," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2020.

[30] P. Zhang, C. Wang, C. Jiang, and A. Benslimane, "Security-aware virtual network embedding algorithm based on reinforcement learning," *IEEE Transactions on Network Science and Engineering*, vol. 8, pp. 1095–1105, 2021.

[31] P. Zhang, C. Wang, C. Jiang, N. Kumar, and Q. Lu, "Resource management and security scheme of icpss and iot based on vne algorithm," *IEEE Internet of Things Journal*, pp. 1–1, 2021.

[32] J. Fan, Z. Ye, C. Guan, X. Gao, K. Ren, and C. Qiao, "Grep: Guaranteeing reliability with enhanced protection in nfv," *Proceedings of the 2015 ACM SIGCOMM Workshop on Hot Topics in Middleboxes and Network Function Virtualization*, 2015.

[33] Y. Kanizo, O. Rottenstreich, I. Segall, and J. Yallouz, "Optimizing virtual backup allocation for middleboxes," *IEEE/ACM Transactions on Networking*, vol. 25, no. 5, pp. 2759–2772, 2017.

[34] L. Qu, M. J. Khabbaz, and C. Assi, "Reliability-aware service chaining in carrier-grade softwarized networks," *IEEE Journal on Selected Areas in Communications*, vol. 36, pp. 558–573, 2018.

[35] Y. Wang, L. Zhang, P. Yu, K. Chen, X. song Qiu, L. Meng, M. Kadoch, and M. Cheriet, "Reliability-oriented and resource-efficient service function chain construction and backup," *IEEE Transactions on Network and Service Management*, vol. 18, pp. 240–257, 2021.

[36] T. Park, Y. Kim, J. Park, H. Suh, B. Hong, and S. Shin, "Qose: Quality of security a network security framework with distributed nfv," in *2016 IEEE International Conference on Communications (ICC)*, 2016, pp. 1–6.

[37] S. Feng, Z. Xiong, D. Niyato, P. Wang, Z. Han, and D. I. Kim, "Joint traffic routing and virtualized security function activation in wireless multihop networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 9, pp. 9205–9219, 2019.

[38] P. K. Thiruvasagam, A. Chakraborty, A. Mathew, and C. S. R. Murthy, "Reliable placement of service function chains and virtual monitoring functions with minimal cost in softwarized 5g networks," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1491–1507, 2021.

[39] J. Wang and J. Liu, "Secure and reliable slicing in 5g and beyond vehicular networks," *IEEE Wireless Communications*, vol. 29, no. 1, pp. 126–133, 2022.

[40] M. Huang, W. Liang, Y. Ma, and S. Guo, "Maximizing throughput of delay-sensitive nfv-enabled request admissions via virtualized network function placement," *IEEE Transactions on Cloud Computing*, vol. 9, no. 4, pp. 1535–1548, 2021.
[41] Y. Qiu, J. Liang, V. C. M. Leung, X. Wu, and X. Deng, "Online reliability-enhanced virtual network services provisioning in fault-prone mobile edge cloud," *IEEE Transactions on Wireless Communications*, vol. 21, no. 9, pp. 7299–7313, 2022.
[42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.
[43] H. Tang, C. Yuan, Z. Li, and J. Tang, "Learning attention-guided pyramidal features for few-shot fine-grained recognition," *Pattern Recognit.*, vol. 130, p. 108792, 2022.
[44] K. Luo, L. Chen, W. Liang, and H. Weng, "A dual-scale morphological filtering method for composite damage identification using fbp," *Mechanical Systems and Signal Processing*, vol. 184, p. 109683, 2023.
[45] W. Zhang, D. Yang, H. Peng, W. Wu, W. Quan, H. Zhang, and X. Shen, "Deep reinforcement learning based resource management for dnn inference in industrial iot," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 8, pp. 7605–7618, 2021.
[46] D. Li, P. Hong, K. Xue, and J. Pei, "Availability aware vnf deployment in datacenter through shared redundancy and multi-tenancy," *IEEE Transactions on Network and Service Management*, vol. 16, no. 4, pp. 1651–1664, 2019.
[47] F. Gandino, R. Ferrero, and M. Rebaudengo, "A key distribution scheme for mobile wireless sensor networks: q - s -composite," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 1, pp. 34–47, 2017.
[48] T. Wang, G. Zhang, A. Liu, M. Z. A. Bhuiyan, and Q. Jin, "A secure iot service architecture with an efficient balance dynamics based on cloud and edge computing," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4831–4843, 2019.
[49] T. Wang, P. Wang, S. Cai, Y. Ma, A. Liu, and M. Xie, "A unified trustworthy environment establishment based on edge computing in industrial iot," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6083–6091, 2020.
[50] T. Wang, H. Luo, X. Zeng, Z. Yu, A. Liu, and A. K. Sangaiah, "Mobility based trust evaluation for heterogeneous electric vehicles network in smart cities," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1797–1806, 2021.
[51] J. Jiang, G. Han, F. Wang, L. Shu, and M. Guizani, "An efficient distributed trust model for wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 5, pp. 1228–1237, 2015.
[52] T. Wang, Y. Li, W. Fang, W. Xu, J. Liang, Y. Chen, and X. Liu, "A comprehensive trustworthy data collection approach in sensor-cloud systems," *IEEE Transactions on Big Data*, vol. 8, no. 1, pp. 140–151, 2022.
[53] E. Paraskevas, T. Jiang, and J. S. Baras, "Trust-aware network utility optimization in multihop wireless networks with delay constraints," in *2016 24th Mediterranean Conference on Control and Automation (MED)*, 2016, pp. 593–598.
[54] W. Wu, P. Yang, W. Zhang, C. Zhou, and X. Shen, "Accuracy-guaranteed collaborative dnn inference in industrial iot via deep reinforcement learning," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 7, pp. 4988–4998, 2021.
[55] S. Xu, I. Koren, and C. M. Krishna, "Adaptive workload adjustment for cyber-physical systems using deep reinforcement learning," *Sustainable Computing: Informatics and Systems*, vol. 30, p. 100525, 2021.
[56] W. Ren, J. Wu, X. Zhang, R. Lai, and L. Chen, "A stochastic model of cascading failure dynamics in communication networks," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 65, no. 5, pp. 632–636, 2018.
[57] F. Tian, J. Liang, and J. Liu, "Joint vnf parallelization and deployment in mobile edge networks," *IEEE Transactions on Wireless Communications*, pp. 1–1, 2023.
[58] R. D. Yates, Y. Sun, D. R. Brown, S. K. Kaul, E. Modiano, and S. Ulukus, "Age of information: An introduction and survey," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 5, pp. 1183–1210, 2021.
[59] M. Savi, M. Tornatore, and G. Verticale, "Impact of processing-resource sharing on the placement of chained virtual network functions," *IEEE Transactions on Cloud Computing*, vol. 9, no. 4, pp. 1479–1492, 2021.
[60] B. Li, Q. He, G. Cui, X. Xia, F. Chen, H. Jin, and Y. Yang, "Read: Robustness-oriented edge application deployment in edge computing environment," *IEEE Transactions on Services Computing*, pp. 1–1, 2020.
[61] K. Zhao, Z. Zhou, X. Chen, R. Zhou, X. Zhang, S. Yu, and D. Wu, "Edgeadaptor: Online configuration adaption, model selection and resource provisioning for edge dnn inference serving at scale," *IEEE Transactions on Mobile Computing*, pp. 1–16, 2022.

Yu Qiu received the B.E. degree from Tianjin University of Technology, Tianjin, China, in 2020. He has been studying for the M.S. degree in electronic and information engineering at Guangxi University, Nanning, China, since 2020. His research interests include mobile edge cloud, network function virtualization, and optimization theory.

Junbin Liang received B.E. and M.S. degrees from Guangxi University, Nanning, China, and the Ph.D. degree from Central South University, Changsha, China, in 2000, 2005, and 2010, respectively. He was a visiting professor in University of British Columbia from 2019 to 2020. He is currently a professor at Guangxi University, Nanning, China. His research interests include sensor-cloud systems, fog computing, and distributed computing.

Victor C.M. Leung [Life Fellow, IEEE] received the B.A.Sc. (Hons.) degree in electrical engineering from the University of British Columbia (UBC), Vancouver, BC, Canada, in 1977, and the Ph.D. degree in electrical engineering from UBC in 1982. He is a Distinguished Professor of Computer Science and Software Engineering at Shenzhen University, China. He is also an Emeritus Professor of Electrical and Computer Engineering and Director of the Laboratory for Wireless Networks and Mobile Systems at the UBC. He is serving on the editorial boards of the IEEE Transactions on Green Communications and Networking, IEEE Transactions on Cloud Computing, IEEE Access, IEEE Network, and several other journals. He is a Life Fellow of IEEE, and a Fellow of the Royal Society of Canada (Academy of Science), Canadian Academy of Engineering, and Engineering Institute of Canada. He is named in the current Clarivate Analytics list of "Highly Cited Researchers".

Min Chen [Fellow, IEEE] has been a Full Professor with School of Computer Science and Engineering, South China University of Technology. He is also the director of Embedded and Pervasive Computing (EPIC) Lab at Huazhong University of Science and Technology. He is the founding Chair of IEEE Computer Society Special Technical Communities on Big Data. He was an assistant professor in School of Computer Science and Engineering at Seoul National University before he joined HUST. He is the Chair of IEEE Globecom 2022 eHealth Symposium. His Google Scholar Citations reached 40,000+ with an h-index of 96. His top paper was cited 4,304+ times. He was selected as Highly Cited Researcher from 2018 to 2022. He got IEEE Communications Society Fred W. Ellersick Prize in 2017, the IEEE Jack Neubauer Memorial Award in 2019, and IEEE ComSoc APB Oustanding Paper Award in 2022. He is a Fellow of IEEE and IET.