

Article

Adaptive SFC Management and Orchestration Based on DRL in Edge Intelligence for Computation Efficiency

Seyha Ros ¹, Taikuong Iv ¹, Intae Ryoo ² and Seokhoon Kim ^{1,3,*}

¹ Department of Software Convergence, Soonchunhyang University, Asan 31538, Republic of Korea; rosseyha003@gmail.com (S.R.); taikuongiv@sch.ac.kr (T.I.)

² Department of Computer Engineering, Kyung Hee University, Yongin-si 17104, Republic of Korea; itryoo@khu.ac.kr

³ Department of Computer Software Engineering, Soonchunhyang University, Asan 31538, Republic of Korea

* Correspondence: seokhoon@sch.ac.kr

Abstract

Network functions virtualization (NFV) is an emerging technology that enables flexible service deployment for supporting the Beyond 5G/6G network. NFV transforms physical network devices into virtual network functions (VNF) over Edge Computing capabilities, thereby facilitating the agility of network services and reducing management costs. To effectively monitor Internet of Things (IoT) network resources, service function chaining (SFC) is used for its virtualizations to ensure the multi-service requirements are sufficiently in capability, scalability, and flexibility for computation workloads alignments. However, to satisfy the resource availability requirements and efficiency under several conditions, SFC reconfiguration methods face the challenges in meeting significant latency requirement of delay-sensitive applications while reaching the importance of energy saving on orchestration timespan. In this paper, we propose task management-aware SFC and orchestrating schemes, namely GNN-PPO. In this framework, we utilize the Graph Neural Network (GNN), which relies on the message-passing neural network (MPNN), to capture all the abstraction of physical resource nodes and link capabilities over MEC node states. In particular, GNN is divided construction into two phrases: (1) GNN represents nodes for all the Mobile edge computing (MEC) nodes, which have a global view on resources of computation and communicational capabilities that could serve as carriers; (2) VNFs are transferred into graph networks by using feature-extraction MPNN to manage each VIM that seeks an optimal and reliable analysis of traffic fluctuations. Lastly, Deep Reinforcement Learning (DRL) is used to embrace the network determination in policy strategy, which utilizes a Proximal Policy Gradient (PPO). On the other hand, we propose a novel network architecture based on PPO to perform the design for the optimization of resource utilization and facilitate energy consumption on MEC servers under diverse setting scenarios, which enables continuous policy enforcement for our system. With the experimental results, we compare our proposed solution with reference schemes in terms of rewards with learning rate and batch size, average request acceptance, SFC success, packet delivery, throughput, and resource utilization ratio that confirm the scheme's scalability and practical suitability for IoT network deployment.



Academic Editors: Bartłomiej Płaczek and Marcin Bernaś

Received: 19 May 2026

Revised: 22 June 2026

Accepted: 25 June 2026

Published: 30 June 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article

distributed under the terms and

conditions of the [Creative Commons](https://creativecommons.org/licenses/by/4.0/)

[Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: computational efficiency; deep reinforcement learning; edge intelligence; network functions virtualization; service functions chaining

1. Introduction

With urban deployment of the internet of things (IoT) devices, increasing amounts of resource computation require both communication and computation capacities to fulfill in differentiation resource demands [1–4]. On the other hand, on-demand resources' computing workload is trending into different B5G/6G network-specified types of service priorities and stringent quality of services (QoS) [5,6]. Meanwhile, the European Telecommunication Standards Institute (ETSI) proposed a network functions virtualization (NFV) framework to ensure the resource capabilities closest to end devices [7,8]. NFV has been used to tackle several challenges in terms of providing resource availability, dynamically and agilely consolidating functionality views over physical resources by adopting more declarative management and organization while addressing the end goals in providing more flexibility and easy integration perspectives [9]. On the other hand, multi-access edge computing (MEC) architecture provides the cutting-edge paradigms to deploy the running virtual machine (VM) for facilitating cloud computing capabilities to network edges, efficiently solving the traffic congestion bottleneck and ensuring the reliability service for terminal devices [10,11]. Especially, significant applications in IoT network are under handling within the MEC architecture. Figure 1 depicts the hierarchical resource alignment of diversities of service-based and network function on-demand. In this flow, management and orchestration (MANO) are set to arrange the resource pooling for handling the request from operation support system (OSS) and business support (BSS) [12–15].

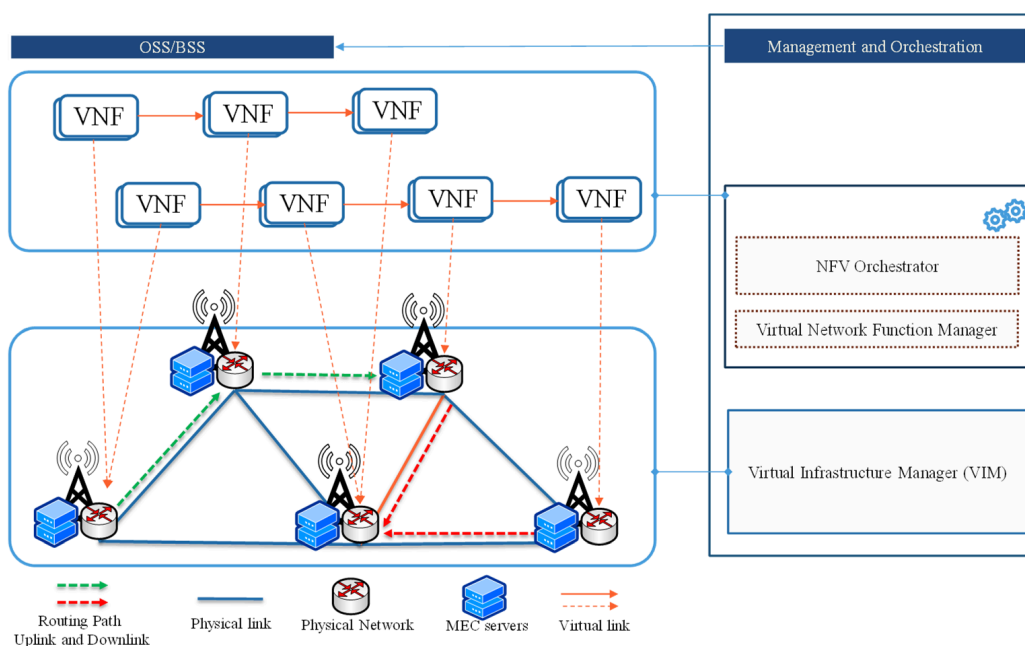


Figure 1. Overall SFC architecture deployed in MEC server for ensuring IoT networks.

However, with a heterogeneous number of IoT devices, traditional approaches are becoming unable to handle the massive traffic requests to satisfy the target requirements of low latency on computation workloads, resulting in resource-intensive, low-efficiency energy consumption in edge computing networks.

This limitation highlights the need for methods that explicitly overcome VNF allocation and SFC placement. Deep reinforcement learning (DRL) is leveraged to handle adaptive resource adjustment and mapping that provides a concise impact to MEC resource management [16–18]. Unlikely with traditional heuristic-based methods, this capability enables a DRL-based framework to learn from network interactions to optimize SFC placement and VNF resource allocation in real time. However, efficient orchestration remains

challenging due to limited edge resources, unpredictable IoT workloads, and conflicting objectives such as QoS and reducing latency and energy consumption targets. Conventional static deployment strategies often fail to adapt to rapidly changing network conditions, as they are simultaneously required to support latency-sensitive and resource-intensive applications in real time. Consequently, the orchestration process must carefully balance multiple conflicting objectives, including minimizing end-to-end latency and energy consumption while maximizing resource utilization, service acceptance ratio, and operational efficiency.

In this paper, we utilize the Graph neural network (GNN) to embed with the hierarchical network in terms of network topology and resource nodes to capture long-term VNF-to-physical node dependency. On the other hand, message-passing neural network (MPNN) variance is employed to execute feature extraction from the physical network resource and SFC states [19,20]. Meanwhile, we design a two-level actor and critic structure with a Proximal Policy Gradient (PPO). The PPO is used to determine the policy for charging SFC deployment in the MEC server, enabling an adaptive allocation of optimal decisions, such as resources in fluctuating IoT networks. This integration with topology-aware learning mechanisms provides a promising direction toward autonomous and adaptive network orchestration for next-generation MEC-enabled IoT systems. Here is the brief of our main motivations and contributions:

- We utilize GNN to capture all spatial network topology by MPNN, including MEC nodes in long-range fluctuations. We design a topology-aware GNN-PPO framework that jointly addresses VNF allocation and mapping, as well as SFC placement by embedding MEC network structures, resource capacities, and service-chain dependencies into graph representations, enabling intelligent policy optimization for dynamic IoT workloads.
- We introduce a normalized, multi-objective System Utility formulation to mathematically reconcile the conflicting physical scales of energy consumption and processing latency. This prevents a reward explosion and ensures stable policy updates.
- We integrate GNN-based feature extraction with PPO-driven decision-making via a clipped surrogate objective, continuously optimizing VNF allocation and SFC routing while preventing the gradient instability observed in traditional heuristic and early-stage DRL baselines under fluctuating scenarios.
- We conduct extensive performance evaluations, demonstrating that the proposed GNN-PPO scheme consistently surpasses conventional DRL and heuristic baselines in service acceptance, convergence stability, throughput, and long-term computation efficiency for MEC-enabled SFC orchestration.

2. Related Work

Many research has investigated the optimization of resource management by using several techniques, such as service function chain offloading algorithm, VNF placement, and resource adjustment, which aim to alleviate network delays and maximize resource utilization to ensure computational capability. Enhancing the proactive precision for VNF resources is crucial in the MANO lifecycle, which involves the increase of energy and delays in orchestrating SFC forms for new templates. In this study, there are several investigations on existing works in terms of approach-based, algorithm-based, and deep learning-based variance in modeling.

By using heuristic algorithms to ensure resource optimization, Cai et al. utilized parallelized SFC techniques with an aim to increase accelerating SFCs. Their proposal utilized heuristic algorithms with distributed NFV architecture to implement SFC and develop parallelized serial SFCs based on the VNF dependency [21]. In [22], the author seeks to minimize the decision time and system's latency for joint VNF deployment and migration

by designing a multi-objective optimization problem. Their proposed algorithm utilizes a collaborative filtering-based fast delay-aware algorithm, which distributes information of deployed/migrated VNFs to assist the current VNF deployment in terms of reducing decision-making time. Thus, in summary, the heuristic algorithm depends on hand-crafted rules that cannot ensure the unseen traffic coming or topology fluctuation and get trapped in local optima.

Recent studies have explored DRL-based methods for dynamically adaptive SFC deployment, resource allocation, energy efficient, and VNF placement in dynamic edge networks environments. Zhang et al. introduced DeepSFC, a distributed self-learning method based on the Proximal Policy Optimization (PPO) algorithm. To overcome the challenges of sparse rewards in DRL, DeepSFC incorporates a curiosity-driven exploration module, enabling agents to make rapid, decentralized orchestration decisions that significantly improve network throughput under complex traffic patterns. While many approaches treat cost, latency, and reliability in isolation, recent frameworks attempt to optimize their complex interdependencies [23]. Xiao et al. introduced A2C-SFC, a two-stage DRL framework designed to meet Ultra-Reliable and Low-Latency Communication (URLLC) standards while minimizing Operational Expenditure (OPEX). In the first stage, A2C-SFC uses an Advantage Actor-Critic (A2C) algorithm integrated with a Karush-Kuhn-Tucker (KKT)-based adaptive resource preemption technique to allow active VNFs to preempt resources from unexecuted VNFs, thereby minimizing processing delays. In the second stage, it applies a reliability-aware redundancy management heuristic to optimally place redundant VNFs, achieving substantial OPEX reductions without violating strict latency and reliability constraints [24]. In [25], this paper proposed an approach for a SFC within an NFV-enabled network by addressing the complexity of both wired and wireless resources. Their proposed aim was to minimize E2E service latency, utilizing a Markov Decision Process (MDP) model of captured dynamics server capacity and radio interference. Additionally, they proposed a natural actor-critic deep reinforcement learning framework, which utilizes natural policy gradients to avoid local optima and improve training stability. On the other hand, to improve the energy-efficient offloading and resource allocation in the edge environment, ref. [26] explicitly presents a multi-user end-edge cloud orchestrated network that designs an optimal computation offloading and resource allocation strategy for computation-intensive and delay-sensitive tasks for particular edges on SBS or MBS, respectively. Additionally, ref. [27] utilizes an intelligent optimization strategy based on DRL for resource allocation strategies across a three-tier cloud-edge-end hierarchy directly relevant to the scalability and architectural positioning, as well as design cache replacement and task scheduling methods for arriving content requests. Their work utilizes asymmetrical control problems caused by a complex service and heterogeneous network environment. This technique is to ensure content distribution for cloud-edge-end nodes. Ref. [28] considers integrating GNN and DRL agents, addressed to overcome complexity of network optimization by using MPNN to perform a feature extraction of the correlation between complex paths and links connection over network topology. However, this proposed method cannot reach QoS guarantees while optimizing resource and energy efficiency. In [29], the author introduces DRL-FJM, a joint mapping approach that simultaneously evaluates node and link resources to maximize the number of accepted service requests. The framework utilizes DRL and PPO to dynamically adapt to shifting network conditions and resource availability. Their technique provides a robust solution for managing the strict delay and capacity demands of modern applications like autonomous driving and IoT.

Through their proposed approaches and solutions for the problem, to push beyond the boundaries of these prior works, this paper proposes a substantive architectural leap. First, we introduce an MPNN variant that explicitly captures spatial link conditions (bandwidth,

propagation delay) alongside node capacities, enabling topology-aware virtual routing. Second, we transition from standard DRL to a PPO architecture. This paper introduces a normalized, dynamically weighted System Utility formulation. This dimensionless objective resolves the mathematical instability caused by conflicting physical scales, allowing the PPO's clipped surrogate objective to safely and consistently converge even under severe, fluctuating network congestion in B5G/6G MEC-enabled environments.

3. System Model and Problem Formulation

This section, we consider the entire network infrastructure for problem formulation and representation of the VNF allocation and SFC placement over MEC servers. As shown, Figure 2 depicts the operational execution from the end-device tier to the control tier. All the VNFs associated with a request are proceeded with MEC servers. The workload on crucial tasks is primarily divided into categories based on network functionalities to achieve resource efficiency and energy consumption in the core network.

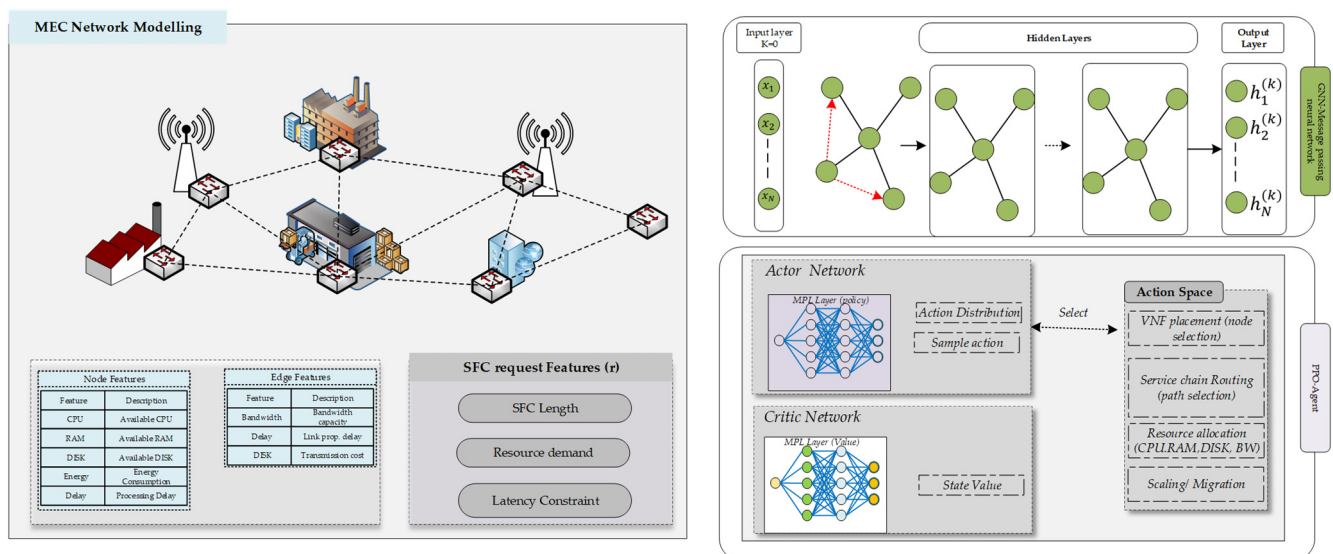


Figure 2. MEC-driven GNN-PPO in adaptively resource allocation and SFC placement.

3.1. Architectural

MEC servers are deployed next to the end-device layer. Additionally, MEC has a control function that enables the SDN/NFV controller. In this context, SDN/NFV is centralized on traffic flow and resource management in terms of required QoS from the IoT request service. We divided it into two phrases: data flow and control flow, which are demonstrated below:

3.1.1. Data Flow

The data flow is to initialize the connection between end-user and network devices by utilizing the communicating protocol and computing in form of the wireless and wired connection.

3.1.2. Control Flow

Upon the arrival of a service request from IoT devices, all data are assumed to be fully offloaded from locally end devices to MEC servers instead. Due to local computing, delays on computing, energy consumption, and workload efficiency are hugely increased. Thus, MEC ensures the ability to overcome resource computing, proving to be reliable and stable in further needs.

3.2. System Models

We demonstrate the resource allocation and mapping into SFC templates to align resources that are demanding under MEC server computation and management. Typically, resource placement is the significant technique for ensuring the resource under controlling for differentiation in providing innovative solutions to the scalability problem of resource constraint.

3.2.1. Network Model

The network model is modeled as directed graph $G = (V, E)$, where $V = \{v_1, v_2, \dots, v_n\}$ is the set number of MEC nodes. E is the set of physical links connecting between MEC. Let $e, v \in V$ represent specific physical nodes and $e_{uv} \in E$ represent the physical link that connected node u to node v . The set of MEC servers possess a specific maximum resource capacity, denoted collectively as C_v^{max} , which encompassed computation resources such as CPU, RAM, and disk storage. Each physical link e_{uv} has a maximum bandwidth capacity denoted by C_{uv}^{bw} . The physical network model hosts various VNFs to satisfy the diverse services required by IoT devices based on OSS/BSS demands. Our system aims to orchestrate these diverse services to strictly adhere to their Quality of Service (QoS) requirements. The unified notations used in our system model and formulations are provided in Table 1.

Table 1. Notation system and variables.

Notations	Description
Network Resources	
$G = (V, E)$	The graph of physical network G where decouples the set of node and link set E
$u, v \in V$	Physical MEC servers/nodes
$e_{uv} \in E$	Physical link connecting node u to node v
N	Set of number IoT devices $N = 1, 2, \dots, n, \forall n \in N$
T	Timeslot – t
Resource capacities and SFC Requests	
C_v^{CPU}	Allocated CPU processing rate (e.g., cycles per second) of MEC node v
C_v^{max}	Maximum available capacity (CPU, RAM, Disk) of MECs node v
C_{uv}^{bw}	Maximum bandwidth capacity of physical link e_{uv}
cap_f^v	Required computation capacity to host a specific VNF f on node v
b_f	Required bandwidth capacity for service request over a virtual link
PT_f	Maximum tolerable processing time for network function f
D_{MAX}, E_{MAX}	Normalization constants for maximum system delay and energy
W_f	Required computational workload to completely process network function f
F, Q	Set of available Network function ($f \in F$) and service requests ($q \in Q$)

Table 1. Cont.

Notations	Description
Binary Variables	
$x_f^v(t)$	Binary variable indicating whether service request that deploy MECs node at time- t , otherwise
$y_f^{uv}(t)$	Binary variable indicating whether virtual link between node at time- t , otherwise
$\lambda_{lat}, \lambda_{ene}, \lambda_{acc}$	Dynamic weight coefficient for latency, energy, and acceptance optimization objectives.

3.2.2. Service Function Chaining Requests Model

In this system, diverse service requests are continuously generated by IoT devices and fully offloaded to the MEC servers for computation in each time slot $t \in T$. The set of service requests is denoted by Q , where each individual request $q \in Q$ demands specific SFCs. An SFC request can be represented as a tuple, defining its operational and resource constraint that shows as $q = (F_q, cap_f^v, b_f, PT_f, src_q, dst_q)$, where $F_q \in F$ denotes the ordered sequence of VNFs that formulate the chain. The variable cap_f^v denotes the required computation capacity to host a specific VNF f on node v , while b_f denotes the required bandwidth capacity for the virtual links connecting the VNFs within the chain. Furthermore, PT_f represents the maximum tolerable processing time for the network function to ensure delay-sensitive QoS requirements are met. Finally, src_q and dst_q define the ingress source and egress destination nodes for the traffic flow traversing the MEC network.

3.3. Problem Statement

Given the MECs with resource and service handling, the main objective is to ensure the service provider maximizes successful placement and orchestration while adhering to the associated constraints.

Constraint on Resource Allocation and Placement

The resource host capacity of proceeds over MECs is composed of utilizing currently available resources. Every MEC can be limited in terms of computation resources; for instance CPU, memory, and disk. The VNF is considered to manage and orchestrate following VIMs for the specific level and type of resource that are necessary to assign to a server on a specification workload. This constraint prevents the overwhelming of physical nodes.

$$\sum_{f \in F} cap_f^v x_f^v(t) \leq C_v^{max}, \forall v \in V, t \in T \quad (1)$$

In Equation (1), all the resources are ensured that they do not exceed their maximum available capacity C_v^{max} .

$$\sum_{q \in Q} \Delta_q^t x_f^v(t) \leq PT_f \quad (2)$$

where Δ_q^t is the end-to-end latency by the service request q in timeslot t . Meanwhile, it can guarantee that resource requirements do not surpass the capacity at destination node following migration. Moreover, the processing load in every single node and bandwidth usage must not exceed in each MEC link, which remains through their respective capacity limits to ensure stable, feasible, and reliable network operation targets.

$$\sum_{f \in F} b_f y_f^{uv}(t) \leq C_{uv}^{bw}, \forall e \in E, t \in T \quad (3)$$

Equation (3) ensures the tolerable bandwidth demanded by b_f , with all virtual links mapping to a physical edge e_{uv} that does not exceed the link's maximum capacity C_{uv}^{bw} .

3.4. Delay Model

3.4.1. Communication Delay

Communication considers the typically carrying packets within end-device and edge networks over the physical medium, which involves service delay criteria, e.g., queuing delays, transition delays, and propagation delays. We ensure they facilitate minimal delays on travers nodes from ingress to egress over chaining VNF nodes.

$$D_{total} = D_{Process} + D_{queue} + D_{trans} + D_{prop} \quad (4)$$

where $D_{Process}$ is the node taken to examine the packet header and check for errors before determining the destination. D_{queue} is in the prose as the dynamic buffer waiting time. This is highly dynamic and relies on network congestion levels. The time required to push all the packet's bits onto the physical wire or wireless medium is determined by the packet length denote as L , and the transmission rate denotes as TR .

$$D_{trans} = \frac{L}{TR} \quad (5)$$

D_{trans} is the set of the period of time for the signal to physically travel from one end of the link to the other. Somehow, it depends on the distance and speed of the signal over the medium interface.

3.4.2. Computation Delay on VNF

With computation delay on orchestrating VNFs, MECs are significantly considered for the resource to host virtual resource computing. This delay depends heavily on the computational workload required by the specific network function and the physical CPU resources allocated by the hosting node. To ensure that computation delay is minimized and kept within reasonable time limits for delay-sensitive applications, we mathematically formulate the processing delay for a given VNF as the ratio of its workload to the allocated server capacity.

$$D_{comp} = \sum_{f \in F} \sum_{v \in V} x_f^v(t) \frac{W_f}{C_v^{cpu}} \quad (6)$$

3.4.3. Optimistic Modelling in SFCs

To offer optimal performance in SFCs, we formulate the orchestration problem as a multi-objective optimization model. Rather than minimizing cost directly, we define a "System Utility" that inversely scales with latency and energy. The objective jointly maximizes this service utility while maximizing the service acceptance rate. The objective function is defined as:

$$\max \mathbb{U} = \sum_{t \in T} \lambda_{lat}(t) \left(1 - \frac{D_t}{D_{MAX}} \right) + \lambda_{ene}(t) \left(1 - \frac{E_t}{E_{MAX}} \right) + \lambda_{acc} A_t \quad (7)$$

where D_t and E_t denote the latency and energy consumption at time slot- t , respectively, and D_{MAX} , E_{MAX} are normalization constants. $A_t \in \{0, 1\}$ represents whether the SFC request is successfully accepted. The coefficients $\lambda_{lat}(t)$, $\lambda_{ene}(t)$, and $\lambda_{acc}(t)$ are weighting factors that dynamically balance the optimization objectives, subject to: (1)–(3).

4. Methodology

To address the placement strategy, we utilize actor–critic mechanism for defining VNF allocation and SFC placement by adopting Markov decision process (MDP) framework. Hence, the main comprises six elements: environment, agent, state space, action space, reward, and design policy enforcement.

Agent: within the agent, there consisted of two components: Actor and Critic. Actor is to set the interaction directly with network environments, which performs selecting action relying on current policy $\pi_\theta(a_t|s_t)$, which means choosing the probability an action a_t in state s_t . Meanwhile, critic evaluates the perform strategy by learning a value function $V_\omega(s_t)$ and output of the estimated value by the discounted cumulative reward s_{t+1} .

4.1. Hierarchical DRL Based on Graph-Passing Neural Network Variance

With capacities of network states being crucial in capturing the complex topology and dynamic resource fluctuation distribution of the MEC nodes, we incorporate a GNN based on MPNN variance. In this work, the physical network is represented as directed graph. On each node, v is associated with feature vector:

$$x_v = [CPU, RAM, DISK, Energy_v, Delay_v] \quad (8)$$

where C_{uv}^{bw} and PD_{uv} are available bandwidth and processing delay at node v , respectively. Similarly, each edge e_{uv} is feature vector to capture the spatial dependencies and propagation delay of the link, respectively.

To capture the spatial dependencies and resource interactions among nodes, the GNN adopts a message-passing mechanism. At each layer k , node v receives messages from its neighboring nodes $u \in \mathcal{N}(v)$, where $\mathcal{N}(v)$ denotes the set of adjacency nodes. Additionally, $m_v^{(k)}$ represents the aggregated message for node v at layer κ , and $\phi(\cdot)$ is a learnable message function implemented using a multilayer perceptron (MLP). This function integrates both the neighbor node embedding $h_u^{(k-1)}$ and the corresponding edge features e_{uv} , allowing the model to jointly consider resource states and link conditions during information propagation.

$$e_{uv} = [C_{uv}^{bw}, PD_{uv}] \quad (9)$$

$$m_v^{(k)} = \sum_{u \in \mathcal{N}(v)} \phi(h_u^{(k-1)}, e_{uv}) \quad (10)$$

By following MPNN, each node updates its embedding by combining its previous representation with the aggregated neighborhood information. The update rule at layer k is given by:

$$h_v^{(k)} = \sigma(W^{(k)} \cdot [h_v^{(k-1)} \| m_v^{(k)}] + b^{(k)}) \quad (11)$$

Here, $h_v^{(k)}$ is the embedding of nodes v at layer κ , and $W^{(k)}$ and $b^{(k)}$ are trainable parameters. The function $\sigma(\cdot)$ denotes a nonlinear activation function such as ReLU. Finally, a graph-level embedding is obtained to capture all states of MEC network. It is achieved using a permutation-invariant readout function, which is defined as:

$$z_G = \frac{1}{|V|} \sum_{v \in V} h_v^{(k)} \quad (12)$$

4.1.1. State Space

The state encapsulates VNF resource and requested SFC information from GNN execution, formally defined as:

- $\mathcal{R}_{coord}^t$: set of the coordination of resource (CPU, RAM, Disk) in MEC server to handle computation n tasks.
- z_G : the global graph embedding extracted from the readout layer the GNN, capture the current network topology.
- $\mathcal{U}_{Max}$: the set of upper-bound resource utilization in MEC node- m at timeslot- t .
- $\mathcal{B}_{uv}^t$: state of total bandwidth.
- T_s : a tuple representation of processing task divides into resource consumption on computation VNF, energy, and time spent.
- req : the specific SFC request feature, such as SFC length, resource demand, and latency constraint.

$$S_t = \{z_G, \mathcal{R}_{coord}^t, \mathcal{U}_{Max}, \mathcal{B}_{uv}^t, T_s, req\} \quad (13)$$

By leveraging this GNN-based state representation, the DRL agent can efficiently and adaptively manage the resource-awareness information more concisely of the network, enabling decision for VNF allocation and SFC placement under dynamic network workloads.

4.1.2. Action Space

With action selection, selection placement decision of SFC and VNF allocation is:

$$a_t = \{x_f^v(t), y_f^{uv}(t), alloc_f^v\} \quad (14)$$

Here, $x_f^v(t) \in \{0, 1\}$ sets the decision to place VNF on physical node, while $y_f^{uv}(t) \in \{0, 1\}$ selects the routing decision on link placement between nodes. Lastly, $alloc_f^v$ depicts the decision on service function request for allocating resource service function types.

4.1.3. Reward

During the latency-aware SFC placement and VNF allocation phase, our objective aims to minimize both the MEC server latency for initial VNF instance and the energy for execution. Thus, the reward function is as follows:

$$r_t = \lambda_{lat}(t) \left(1 - \frac{D_t}{D_{MAX}}\right) + \lambda_{ene}(t) \left(1 - \frac{E_t}{E_{MAX}}\right) + \lambda_{acc} A_t - \mathcal{P}_t \quad (15)$$

where $\mathcal{P}_t$ is the penalty applied for constraint violation. λ_{lat} and λ_{ene} are time-varying weights ($\lambda_{lat} + \lambda_{ene} = 1$) that adapt based on current conditions. When the network load is high or delay-sensitive service requests are prioritized, $\lambda_{lat}(t)$ is increased to focus on reducing latency. Conversely, under lower loads or when energy conservation is critical, $\lambda_{ene}(t)$ is emphasized to minimize energy consumption.

4.2. Hierarchical Reinforcement Learning Based on PPO Framework

PPO is widely used for its efficiency and significant performance combined with RL to enhance learning policy. We employ a PPO-based actor-critic architecture consisting of: (1), actor network $\pi_\theta(a|s)$ chooses a probability distribution over action, and (2), critic network $V_\theta(s)$ estimates the state value function. The policies have identified clipped surrogate objective functions as in Equation (16). Otherwise, in (17), generalized advantage estimation is utilized for calculation in $\hat{A}_t$ advantage estimate.

$$L^{CLIP}(\theta) = \hat{E}_t [\min(p_t(\theta) \hat{A}_t, clip(r_t(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_t)] \quad (16)$$

$$\hat{A}_t = \sum_{l=0}^{\infty} (\gamma \lambda)^l \delta_{t+l} \quad (17)$$

$$\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t) \quad (18)$$

$$\theta \leftarrow \theta + \alpha_{\theta} \frac{L^{CLIP}(\theta)}{\partial \theta} \quad (19)$$

4.3. Algorithms Design

Our scheme is to enhance the network resource on executing the management for SFC orchestrations. As shown, Figure 3 depicts the state procedure of initial network components. This objective aims to fulfill reliability requirement handling while minimizing the latency on orchestration and energy consumption within centric. The hierarchical set between GNN is incorporated with DRL to execute policy enforcement in efficient resource management and control the packets from end user on resource demand in SDN/NFV controller.

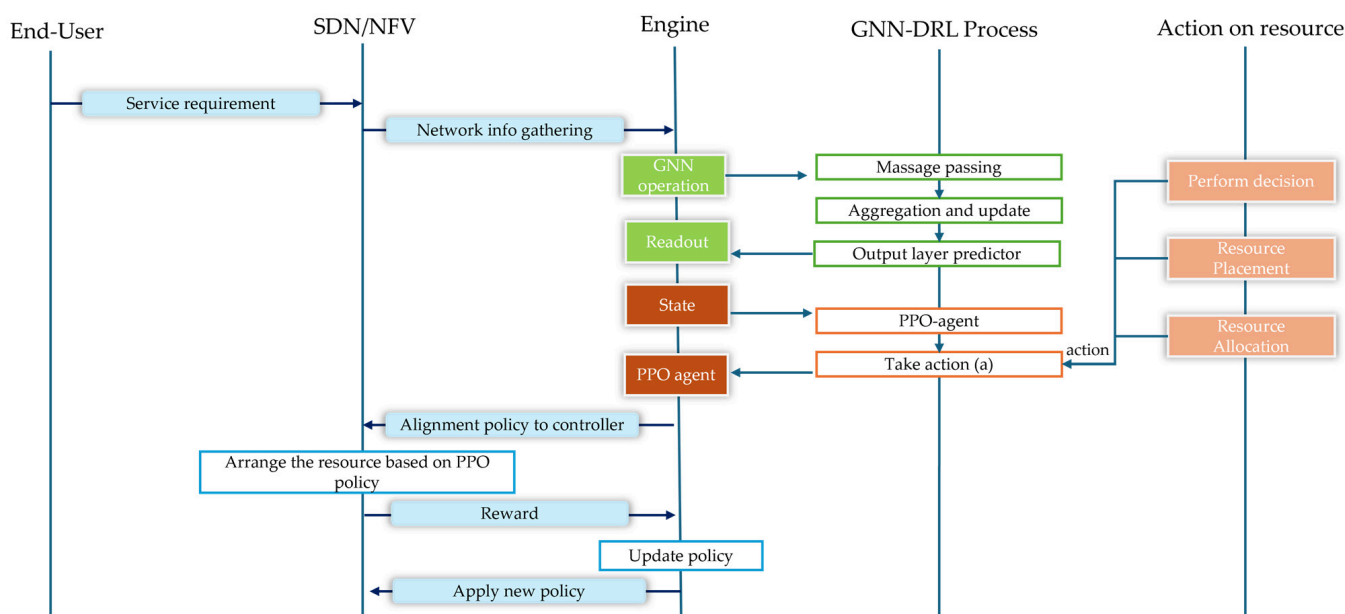


Figure 3. State procedure PPO-enable SDN/NFV for resource orchestration.

4.4. Joint Adaptively Resource Orchestration Based on GNN-PPO Framework

To adaptively orchestrate, GNN-PPO is joined for decision-making for handling resource orchestration and management over MEC servers. Algorithm 1 depicts an execution as a closed-loop optimization cycle to achieve joint VNF allocation and SFC orchestration. At each time step, the current network topology G_t and incoming service request are fed into a K-layer MPNN to extract topology-aware spatial and resource dependencies. Through k iterations of recursive message passing, the MPNN aggregates localized neighborhood attributes to generate refined node embeddings $h_v^{(k)}$. A permutation-invariant readout layer subsequently condenses into a global graph embedding. This graph representation is concatenated with immediate SFC constraints to construct the composite environment state vector. Leveraging this comprehensive state representation, the PPO Actor network yields an orchestration decision $a_t \sim \pi(a_t|s_t)$ defining specific VNF placement locations and traffic routing paths. Following execution within the MEC environment where processing, bandwidth, and latency constraints are verified, the system computes a scalar reward reflecting service acceptance and resource conservation. Trajectory transitions are collected in an on-policy rollout buffer D . Once the rollout buffer reaches the specified batch size threshold, it triggers a joint network update. Here, the Generalized Advantage Estimates (GAE) are computed via the Critic network to optimize the Actor policy and GNN parameters via the clipped PPO objective while minimizing the Critic's Mean Squared Error (MSE) value loss.

Algorithm 1: Adaptively joint VNF allocation and SFC placement-based GNN-PPO framework

Input: Network graph $G = (V, E)$, node/edge resource state, SFCr, GNN parameter and PPO (Actor and Critic)

Output: Optimized VNF placement, SFC offloading policies, and resource allocation

Learning process:

- 1 Initialize GNN parameter, PPO
- 2 Initialize rollout buffer D and environment timestep $t = 0$
- 3 **For each**
 - // topology-aware feature extraction*
 - 4 Observe network topology G_t and SFC request r_t
 - 5 Construct node features x_v and edge features e_{uv}
 - 6 Initialize node embeddings $h_v^{(0)} \leftarrow x_v$
 - 7 **For** $k=1$ to K **do**
 - 8 Aggregate neighbor messages:
$$m_v^{(k)} = \sum_{u \in \mathcal{N}(v)} \phi(h_u^{(k-1)}, e_{uv})$$
 - 9 Update node embeddings:
$$h_v^{(k)} = \sigma(W^{(k)} \cdot [h_v^{(k-1)} \| m_v^{(k)}] + b^{(k)})$$
 - 10 **End for**
 - // state representation*
 - 11 graph embedding $z_G \leftarrow \text{Readout } h_v^{(k)}$
 - 12 Construct state $s_t \leftarrow z_G, r_t$
 - // policy generation (actor)*
 - 13 Selection action $a_t \sim \pi(a_t | s_t)$
 - // environment execution*
 - 14 Execute VNF placement, SFC routing, and resource allocation
 - 15 Verify resource, bandwidth, and latency constraints
 - 16 Compute reward:
$$r_t = \lambda_{lat}(t) \left(1 - \frac{D_{total}}{D_{MAX}}\right) + \lambda_{ene}(t) \left(1 - \frac{E_{total}}{E_{MAX}}\right) + \lambda_{acc} A_t - \mathcal{P}_t$$
 - 17 Store transition (s_t, a_t, r_t, s_{t+1})
 - 18 **if** D reaches update the batch update threshold, **then**
 - 19 Compute advantage $\hat{A}_t$ using GAE
 - 20 Update Actor using PPO clipped objective and Critic using value loss
 - 21 **End for**
 - 22 Return optimized policy π^*

5. Performance Evaluation

To evaluate the GNN-PPO framework, we simulate the dynamic MEC environment using the python program, which utilizes the NetworkX library to create network topologies. We employ PyTorch for the Deep Learning model and Gymnasium for DRL, and the vary network conditions are considered in three types. Additionally, OASIS TOSCA NVF is utilized to create the concept of service temple for topology template in terms of network

function deployment in this study [30,31]. To accurately reflect realistic B5G/6G edge computing scenarios, the network topology is instantiated as a partially meshed graph of five MEC servers with an edge connection probability of 0.6, representing dispersed edge density. Service requests from the 100 IoT devices are dynamically generated following a Poisson arrival process with a rate of (10, 50) requests per timeslot. Each request demands an SFC consisting of two to five VNFs drawn uniformly from five distinct VNF types (e.g., Firewall, NAT, IDS).

Furthermore, to calculate the energy consumption metric for the System Utility objective, we adopt a standard linear-server energy consumption model. The energy consumed by MEC server v at time t is calculated as $E_v = P_{idle} + (P_{max} - P_{idle}) \times U_v(t)$, where P_{idle} and P_{max} are the idle and peak power consumption, and $U_v(t)$ denotes the current CPU utilization ratio of the server [32]. The complete granular configurations for the network resources, service, and DRL hyperparameters are summarized in Table 2.

Table 2. Parameters setting on simulation and hyperparameter.

Parameters	Value
Hosting network infrastructure	Intel(R) i7-12700 CPU @ 2.10 GHz, Ram 32 GB @ 3200 Mhz
Number of IoT (end-devices)	100
Number of MECs	5
Bandwidth on link	Random set (100–1000 Mbps)
Link propagation delay	(1–5 ms)
Topology and edge density	Partially meshed graph, edge connection probability $p = 0.6$
Service Configuration	
VNF	VNF 5 types
Request arrival rate	Poisson distribution (10–50) req/timeslot
SFC type of length	(2–5)
Virtual link demand/Flow rate	(64 Kbps–4 Mbps)
The configuration of DRL models	
Learning rate	Value subset (0.00001, 0.0001, 0.001)
Hidden size	Value subset (32, 64, 128)
Discount factor	0.95
GAE trace-decay	0.95
PPO clipping	0.2
Buffer size	1×10^5
Batch size	Value subset (32, 64, 128)
Number of episodes	10,000
Python platform	PyTorch v2.11.0

5.1. Comparison of Propose and Reference Schemes

In this study, we validate our performance method compared with the four baselines.

- DQN utilizes a Deep Q-Network to learn control policies from network states, but its performance declines in complex topologies, making it more suitable for lightweight network environments.

- Load-balancing focuses on distributing workloads evenly across available resources by selecting target networks and assigning VNFs to SFCs based on priority. While it improves utilization and prevents overload, it struggles to adapt to strict or dynamic VNF priority constraints.
- Greedy reflects a traditional SDN/NFV-based MANO strategy for MEC resource management, relying on centralized topology-aware optimization, traffic conditions, and service policies. It performs adequately for simple, non-complex applications but lacks intelligence for dynamic large-scale orchestration.
- Static-on resource uses stochastic MEC service selection from ingress nodes through predefined VNF instances and routing paths for each IoT request, offering simple service chaining but limited adaptability and optimization efficiency.

5.2. Results and Discussions

In this section, we present the results for the proposed GNN-PPO and reference schemes, namely DQN, Load-balance, Greedy, and Static. In this stage, we perform our proposal explicitly in terms of learning rate and batch size.

In Figure 4a, the influence of learning rate (LR) on the convergence behavior of the proposed GNN-PPO framework is evaluated under three configurations (0.00001, 0.0001, and 0.001). During the initial training stage, all configurations exhibit highly negative rewards, indicating inefficient policy exploration as the GNN encoder and PPO policy network are still learning to interpret topological states, resource availability, and SFC demands within the MEC-enabled IoT environment. At this phase, GNN has not yet fully captured the structural dependencies between network nodes, while PPO's actor-critic policy remains unstable due to random action selection. As training progresses, LR (0.00001) demonstrates extremely slow adaptation, with severe reward oscillations persisting across episodes, suggesting that the PPO optimizer updates the GNN-extracted embeddings too conservatively, thereby limiting effective policy refinement for VNF placement and resource orchestration. LR (0.0001) provides improved convergence by enabling more balanced policy gradient updates, allowing the PPO agent to better utilize GNN-based topology-aware state embeddings for adaptive resource scheduling. However, moderate fluctuations remain, indicating slower responsiveness to dynamic workload variations. In contrast, LR (0.001) achieves the fastest and most stable reward convergence, rapidly improving within early episodes and consistently maintaining the highest reward levels. Figure 4b illustrates the effect of batch size on the training efficiency of the proposed GNN-PPO model, where BatchSizes (32, 64, and 128) are compared. In early episodes, all batch sizes begin with substantially negative rewards due to incomplete policy awareness of network topology and computational resource distribution. However, because the proposed model integrates GNN-based message passing to encode node connectivity, link capacity, and distributed MEC resource states, PPO gradually improves decision quality as batch updates refine both actor and critic networks. BatchSize (32) initially converges faster because smaller batches enable more frequent PPO parameter updates, allowing the GNN-PPO agent to quickly adapt to short-term resource orchestration patterns. Nevertheless, this configuration also suffers from noticeable reward instability, including sudden performance drops, indicating that overly frequent updates may amplify variance in policy gradients and reduce long-term stability. BatchSize at 64 demonstrates the most effective balance, achieving robust convergence with fewer severe oscillations. This suggests that moderate batch sizes allow the GNN to generate sufficiently informative topology embeddings while PPO benefits from stable gradient estimation, producing consistent optimization of SFC placement, node utilization, and energy-aware scheduling. BatchSize at 128, although initially slower and exhibiting deeper instability, eventually stabilizes as larger batch aggregation improves

PPO's gradient consistency; however, delayed responsiveness reduces adaptation speed to dynamic MEC workload changes.

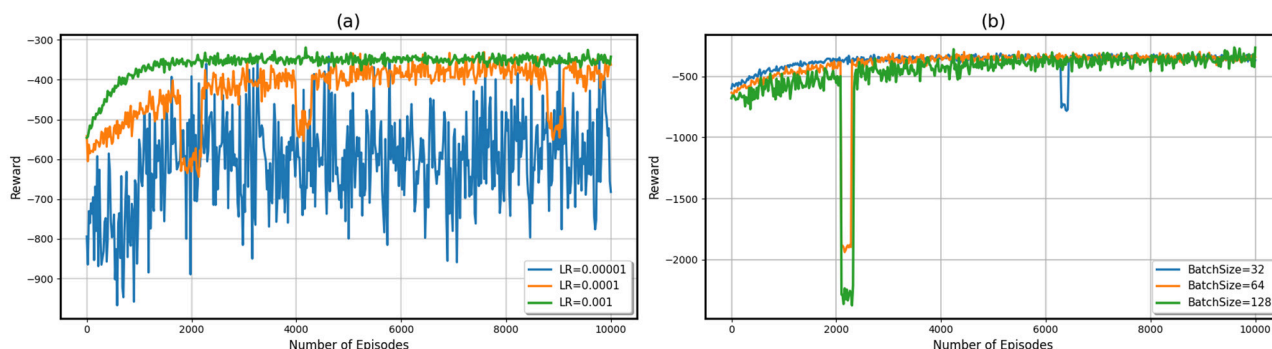


Figure 4. Performance metrics on rewards of (a) learning rate and (b) batch size.

The results are demonstrated in Figure 5a,b for average request acceptance and SFC success rate, respectively, where network states are configured consecutively by setting different network conditions, congestion levels, and application tasks. Figure 5a shows the average SFC request acceptance ratio as service demand increases from 100 to 500 requests over simulation time. All methods experience declining acceptance rates due to higher resource contention and network saturation; however, the proposed method consistently outperforms all baseline schemes across all traffic levels. At 100 requests, GNN-PPO achieves nearly perfect acceptance (99.998%), surpassing DQN (98.49%), Load-balancing (96.11%), Greedy (94.92%), and Static (89.93%). Even under heavy workload at 500 requests, the proposed method maintains 93.42% acceptance, significantly higher than DQN (87.99%) and conventional approaches. Meanwhile, Figure 5b demonstrates a consistently better performance than DQN, Load-balancing, Greedy, and Static approaches across all simulation times. At 100 simulation time, GNN-PPO achieves the highest success ratio at 99.91%, demonstrating superior topology-aware SFC placement and resource orchestration through the integration of GNN-based network representation and PPO policy optimization. Although all methods decline as simulation time increases due to higher resource contention and network complexity, GNN-PPO maintains the strongest performance, retaining 88.42% at 500 simulation time, compared to DQN (82.17%), Load-balancing (79.99%), Greedy (78.01%), and Static (73.23%).

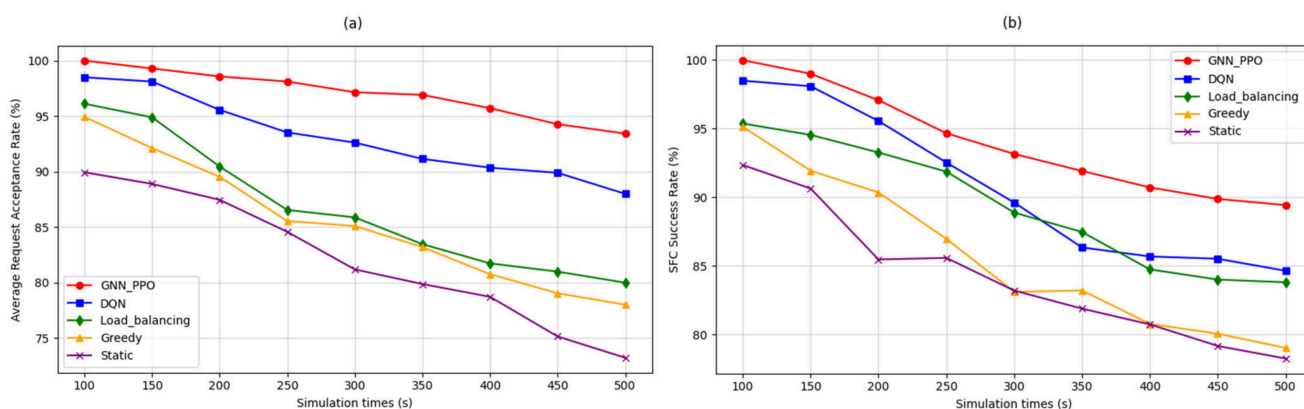


Figure 5. Performance metrics on (a) average request acceptance and (b) SFC success rate.

With the results presented in Figure 6a, GNN-PPO achieves the highest PDR at 99.98% during the initial stage and maintains superior stability with 99.96% at 500 simulation time, demonstrating more reliable packet transmission under dynamic MEC conditions. While all

methods experience slight degradation as network complexity increases, GNN-PPO shows the smallest performance decline due to its integration of GNN-based topology awareness and PPO-driven adaptive policy optimization. Figure 6b shows that the proposed GNN-PPO method achieves the highest and most stable performance, with throughput decreasing only slightly from 805.8 Mb/s to 801.7 Mb/s (drop of 4.1 Mb/s), indicating strong adaptability and efficient resource orchestration under dynamic network conditions. DQN performs similarly, but it experiences a slightly larger decline (4.9 Mb/s), while Load Balancing, Greedy, and Static show significantly greater throughput degradation due to less adaptive decision-making.

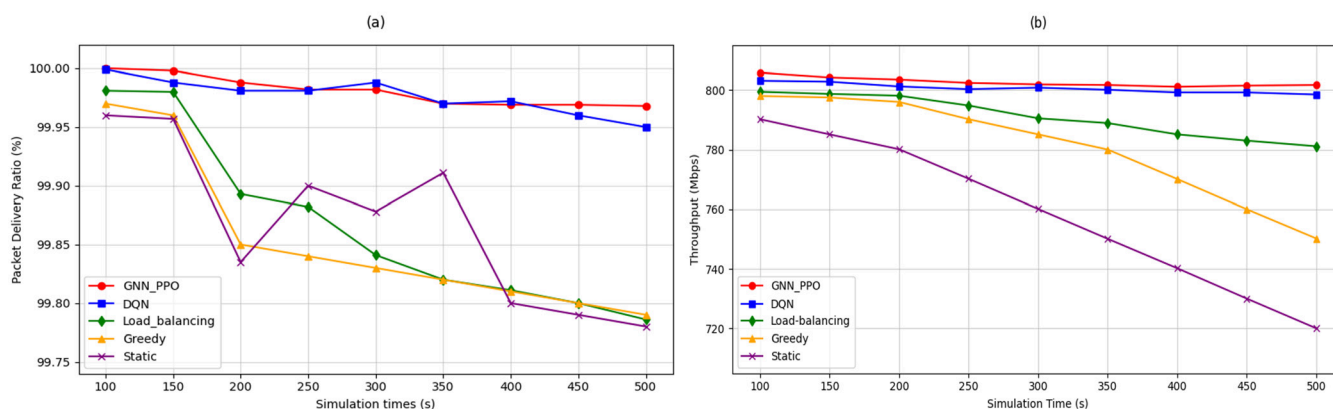


Figure 6. Performance metrics of (a) packet delivery and (b) throughput.

Figure 7 depicts the GNN-PPO framework consistently minimizing resource consumption across all simulation intervals, demonstrating superior efficiency in VNF placement, SFC orchestration, and adaptive resource allocation. Specifically, GNN-PPO reduces resource utilization from an initial 67.99% to 55.42% by simulation time 500, highlighting its capacity to intelligently allocate computational and network assets with minimal overhead. Conversely, DQN maintains higher utilization levels, declining only to 63.42% (from 69.99%). Traditional baselines Load Balancing, Greedy, and Static exhibit the poorest performance, ending at 69.42%, 70.42%, and 72.42%, respectively, due to their rigid orchestration.

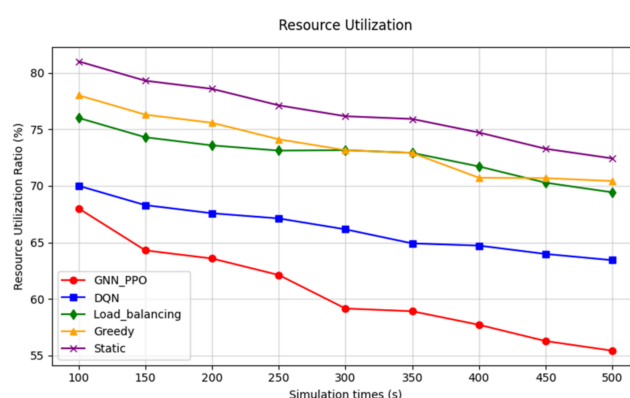


Figure 7. Resource utilization ratio (note: lower values indicate better performance via reduced resource overhead).

These outcomes confirm that the GNN successfully captures topology-aware resource dependencies, while PPO dynamically optimizes VNF placement and SFC offloading, thereby maximizing scalability, congestion handling, and resource utilization. In contrast, DQN shows moderate degradation, while conventional baselines (Load-balancing, Greedy, and Static) decline sharply due to limited adaptability.

6. Conclusions

In this study, we proposed a novelty approach for enhancing VNF placement and SFC orchestration in MEC server environment through integration of GNN incorporated with the PPO framework. The proposed method framework demonstrated its effectiveness by reducing the time of orchestrating resource intensive and energy consumption in dynamic resource workload over MEC scenarios, while GNN-PPO provides a significantly better solution for fast response capacities on resource-constraint edge devices. By integrating a message passing neural network into a PPO (actor, critic) architecture, in this phrase, MPNN captures the state of SFC features and network topologies, thereby achieving coupled decision-making for node mapping. Otherwise, PPO determines the policy for charging SFC deployment in the MEC server, enabling adaptive allocation of optimal decisions for resource instances. Our simulation result outperforms the reference scheme in terms of reducing latency and energy consumption targets in complex task scenarios.

Author Contributions: Conceptualization, S.R., T.I., I.R. and S.K.; methodology, S.R.; software, S.R.; validation, S.R.; formal analysis, S.R., T.I. and I.R.; investigation, S.K.; resources, S.K.; data curation, S.R.; writing—original draft preparation, S.R. and S.K.; writing—review and editing, S.R.; visualization, S.R.; supervision, S.K.; project administration, S.K.; funding acquisition, S.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by BK21 FOUR (Fostering Outstanding Universities for Research) under Grant 5199990914048; and in part by the Soonchunhyang University Research Fund.

Data Availability Statement: Derived data supporting the findings of this study are available from the corresponding author on request.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Shah, S.H.; Yaqoob, I. A Survey: Internet of Things (IOT) Technologies, Applications and Challenges. In Proceedings of the 2016 IEEE Smart Energy Grid Engineering (SEGE), Oshawa, ON, Canada, 21–24 August 2016; pp. 381–385. Available online: <https://ieeexplore.ieee.org/abstract/document/7589556/> (accessed on 12 January 2026).
- Lee, S.K.; Bae, M.; Kim, H. Future of IoT Networks: A Survey. *Appl. Sci.* **2017**, *7*, 1072. [\[CrossRef\]](#)
- López-Benítez, M.; Alhulayil, M. On the Computational Efficiency of Fading Models: The Fluctuating Two-Ray Model for MmWave Bands. *IEEE Open J. Commun. Soc.* **2025**, *6*, 8175–8189. [\[CrossRef\]](#)
- Chopra, A.; Aydin, H.; Rafatirad, S.; Homayoun, H. Optimal Allocation of Computation and Communication in an IoT Network. *ACM Trans. Des. Autom. Electron. Syst.* **2018**, *23*, 1–22. [\[CrossRef\]](#)
- Sandeepa, C.; Siniarski, B.; Kourtellis, N.; Wang, S.; Liyanage, M. A Survey on Privacy for 5G/6G: New Privacy Challenges, and Research Directions. *J. Ind. Inf. Integr.* **2022**, *30*, 100405. [\[CrossRef\]](#)
- Wang, J.; Li, Y.; Liu, J.; Kato, N. Intelligent Network Slicing for 5G and 6G: Resource Allocation, Service Provisioning, and Security. *IEEE Wirel. Commun.* **2024**, *31*, 271–277. [\[CrossRef\]](#)
- Huang, Y.X.; Chou, J. A Survey of NFV Network Acceleration from ETSI Perspective. *Electronics* **2022**, *11*, 1457. [\[CrossRef\]](#)
- Ros, S.; Tam, P.; Song, I.; Kang, S.; Kim, S. Handling Efficient VNF Placement with Graph-Based Reinforcement Learning for SFC Fault Tolerance. *Electronics* **2024**, *13*, 2552. [\[CrossRef\]](#)
- Aelken, J.; Triay, J.; Chatras, B.; de Nicolas, A.M. Toward Cloud-Native VNFs: An ETSI NFV Management and Orchestration Standards Approach. *IEEE Commun. Stand. Mag.* **2024**, *8*, 12–19. [\[CrossRef\]](#)
- Fulber-Garcia, V.; Flauzino, J.; Dos Santos, C.R.P.; Duarte, E.P. An ETSI-Compliant Architecture for the Element Management System: The Key for Holistic NFV Management. In Proceedings of the 2023 19th International Conference on Network and Service Management (CNSM), Niagara Falls, ON, Canada, 30 October–2 November 2023; pp. 1–9. [\[CrossRef\]](#)
- Pereira, P.; Escalera, P.; Gomes, D.; Aguiar, R. Automatic Service Migration in a MEC-NFV Environment. In Proceedings of the 2024 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), Natal, Brazil, 5–7 November 2024; pp. 1–7. [\[CrossRef\]](#)
- Chan, Y.C.; Yen, L.H.; Wang, T.H.; Tseng, C.C. MEC Implementation in NFV Architecture Supporting 5G End-To-End Network Slicing. In Proceedings of the 2023 2nd International Conference on 6G Networking (6GNet), Paris, France, 18–20 October 2023; pp. 1–4. [\[CrossRef\]](#)

13. Mijumbi, R.; Serrat, J.; Gorricho, J.; Latre, S.; Charalambides, M.; Lopez, D. Management and Orchestration Challenges in Network Functions Virtualization. *IEEE Commun. Mag.* **2016**, *54*, 98–105. [\[CrossRef\]](#)
14. Tam, P.; Ros, S.; Song, I.; Kang, S.; Kim, S. A Survey of Intelligent End-To-End Networking Solutions: Integrating Graph Neural Networks and Deep Reinforcement Learning Approaches. *Electronics* **2024**, *13*, 994. [\[CrossRef\]](#)
15. Doan, T.V.; Nguyen, G.T.; Reisslein, M.; Fritzek, F.H. SAP: Subchain-Aware NFV Service Placement in Mobile Edge Cloud. *IEEE Trans. Netw. Serv. Manag.* **2022**, *20*, 319–341. [\[CrossRef\]](#)
16. Ros, S.; Kang, S.; Iv, T.; Song, I.; Tam, P.; Kim, S. Priority-Aware Resource Allocation for VNF Deployment in Service Function Chains Based on Graph Reinforcement Learning. *Comput. Mater. Contin.* **2025**, *83*, 1649. [\[CrossRef\]](#)
17. Zhang, H.; Wang, H.; Li, Y.; Long, K.; Nallanathan, A. DRL-Driven Dynamic Resource Allocation for Task-Oriented Semantic Communication. *IEEE Trans. Commun.* **2023**, *71*, 3992–4004. [\[CrossRef\]](#)
18. Ros, S.; Ryoo, I.; Kim, S. DRL-Driven Intelligent SFC Deployment in MEC Workload for Dynamic IoT Networks. *Sensors* **2025**, *25*, 4257. [\[CrossRef\]](#) [\[PubMed\]](#)
19. Tam, P.; Song, I.; Kang, S.; Ros, S.; Kim, S. Graph Neural Networks for Intelligent Modelling in Network Management and Orchestration: A Survey on Communications. *Electronics* **2022**, *11*, 3371. [\[CrossRef\]](#)
20. Zhang, X.; Li, S.; Li, H.; Gong, P.; Shi, J.; Shen, J.; Tian, C.; Zhang, D.; Zhu, Q. A Survey on Graph Neural Networks. *IEEE Access* **2026**, *14*, 27491–27512. [\[CrossRef\]](#)
21. Cai, J.; Huang, Z.; Luo, J.; Liu, Y.; Zhao, H.; Liao, L. Composing and Deploying Parallelized Service Function Chains. *J. Netw. Comput. Appl.* **2020**, *163*, 102637. [\[CrossRef\]](#)
22. Liao, Z.; Deng, W.; He, S.; Tang, Q. Collaborative Filtering-Based Fast Delay-Aware Algorithm for Joint VNF Deployment and Migration in Edge Networks. *Comput. Netw.* **2024**, *243*, 110300. [\[CrossRef\]](#)
23. Zhang, W.; Zhang, Y. Adaptive Service Function Chain Orchestration Algorithm Based on Deep Reinforcement Learning. In Proceedings of the 2024 2nd International Conference on Signal Processing and Intelligent Computing (SPIC), Guangzhou, China, 20–22 September 2024; pp. 564–569. [\[CrossRef\]](#)
24. Xiao, Y.; He, Z.; Peng, Q. Cost-Efficient and Reliable SFC Orchestration in Mobile Edge Computing. In *Proceedings of the 9th Asia-Pacific Workshop on Networking, Shanghai, China, 7–8 August 2025*; Association for Computing Machinery: New York, NY, USA, 2025; pp. 192–199. [\[CrossRef\]](#)
25. Wang, R.; Li, J.; Wang, K.; Liu, X.; Lit, X. Service Function Chaining in NFV-Enabled Edge Networks with Natural Actor-Critic Deep Reinforcement Learning. In Proceedings of the 2021 IEEE/CIC International Conference on Communications in China (ICCC), Xiamen, China, 28–30 July 2021; pp. 1095–1100. [\[CrossRef\]](#)
26. Dai, Y.; Zhang, K.; Maharjan, S.; Zhang, Y. Edge Intelligence for Energy-Efficient Computation Offloading and Resource Allocation in 5G Beyond. *IEEE Trans. Veh. Technol.* **2020**, *69*, 12175–12186. [\[CrossRef\]](#)
27. Fang, C.; Zhang, T.; Huang, J.; Xu, H.; Hu, Z.; Yang, Y.; Wang, Z.; Zhou, Z.; Luo, X. A DRL-Driven Intelligent Optimization Strategy for Resource Allocation in Cloud-Edge-End Cooperation Environments. *Symmetry* **2022**, *14*, 2120. [\[CrossRef\]](#)
28. Almasan, P.; Suárez-Varela, J.; Rusek, K.; Barlet-Ros, P.; Cabellos-Aparicio, A. Deep Reinforcement Learning Meets Graph Neural Networks: Exploring a Routing Optimization Use Case. *Comput. Commun.* **2022**, *196*, 184–194. [\[CrossRef\]](#)
29. Wu, Y.; Hu, H.; Zhang, Z. DRL-Based Fast Joint Mapping Approach for SFC Deployment. *Electronics* **2025**, *14*, 2408. [\[CrossRef\]](#)
30. OASIS. OASIS Topology and Orchestration Specification for Cloud Applications (TOSCA) TC. Available online: https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=tosca (accessed on 9 February 2026).
31. TOSCA Simple Profile for Network Functions Virtualization (NFV) Version 1.0. Available online: <http://docs.oasis-open.org/tosca/tosca-nfv/v1.0/tosca-nfv-v1.0.html> (accessed on 24 January 2026).
32. Jin, C.; Bai, X.; Yang, C.; Mao, W.; Xu, X. A Review of Power Consumption Models of Servers in Data Centers. *Appl. Energy* **2020**, *265*, 114806. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.