Traffic Prediction-Assisted Federated Deep Reinforcement Learning for Service Migration in Digital Twins-Enabled MEC Networks

Citation for published version:
Chen, X, Han, G, Bi, Y, Yuan, Z, Marina, MK, Liu, Y & Zhao, H 2023, 'Traffic Prediction-Assisted Federated Deep Reinforcement Learning for Service Migration in Digital Twins-Enabled MEC Networks', *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 10, 10234717, pp. 3212-3229. https://doi.org/10.1109/JSAC.2023.3310047

Digital Object Identifier (DOI):
10.1109/JSAC.2023.3310047

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
IEEE Journal on Selected Areas in Communications

Traffic Prediction-Assisted Federated Deep Reinforcement Learning for Service Migration in Digital Twins-Enabled MEC Networks

Xiangyi Chen, Guangjie Han, *Fellow, IEEE,* Yuanguo Bi, *Member, IEEE,* Zimeng Yuan,
Mahesh K. Marina, *Member, IEEE,* Yufei Liu, and Hai Zhao

***Abstract*—In Mobile Edge Computing (MEC) networks, dynamic service migration can support service continuity and reduce user-perceived delay. However, service migration in MEC networks faces significant challenges due to the uncertainty in future traffic demands, the distributed architecture of MEC networks, high operating costs and the dynamism of network resources. Digital Twins (DT), which achieve the mapping of physical entities to virtual digital models in cyberspace, provide new perspectives for intelligent and efficient service provisioning in MEC networks. In this paper, we propose a traffic prediction-assisted federated deep reinforcement learning scheme to efficiently migrate services and improve the cost efficiency of DT-enabled MEC networks. Specifically, to address the coupled spatio-temporal dependencies of mobile traffic and the imbalance in traffic data, a Multi-order Spatio-temporal information integration-based distributed Traffic Prediction (MSTP) scheme is proposed, which achieves high-accuracy mobile traffic prediction at a low cost. Then, we propose a Federated Cooperative cost-efficient Service Migration (FCSM) algorithm that adaptively adjusts service migration strategies in a distributed manner to respond to future traffic demands. Moreover, a theoretical model is developed to analyze the convergence of FCSM and derive the upper bound of the time-average squared gradient norm. Finally, extensive simulations demonstrate that the proposed schemes achieve excellent traffic prediction performance, enhance users' Quality of Service (QoS), and significantly reduce the system cost of MEC networks.**

***Index Terms*—Mobile edge computing, digital twins, service migration, mobile traffic prediction, deep reinforcement learning.**

Manuscript received 1 December 2022; revised 15 April 2023; accepted 3 August 2023. Date of publication 7 August 2023; date of current version 7 August 2023. This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFE0114200; in part by the National Natural Science Foundation of China under Grant U1808207; in part by the Fundamental Research Funds for the Central Universities of China under Grant N2224001-7, Grant N2116014, and Grant N2116002. *(Corresponding author: Guangjie Han, Yuanguo Bi.)*

Xiangyi Chen is with the School of Computer Science and Engineering, Northeastern University and Engineering Research Center of Security Technology of Complex Network System, Ministry of Education, Shenyang 110169, China, and also with the School of Informatics, University of Edinburgh, Edinburgh EH8 9AB, U.K. (e-mail: xiangyichen3746@gmail.com).

Yuanguo Bi, Zimeng Yuan, Yufei Liu, and Hai Zhao are with the School of Computer Science and Engineering, Northeastern University, Shenyang 110169, China, and also with the Engineering Research Center of Security Technology of Complex Network System, Ministry of Education, Shenyang 110169, China (e-mail: biyuanguo@mail.neu.edu.cn; yunmengyuanzimeng@outlook.com; liuyufei1119@163.com; zhaoh@cse.neu.edu.cn).

Guangjie Han is with the Department of Internet of Things Engineering, Hohai University, Changzhou, 213022, China (e-mail: hanguangjie@gmail.com).

Mahesh K. Marina is with the School of Informatics, University of Edinburgh, Edinburgh EH8 9AB, U.K. (e-mail: mahesh@ed.ac.uk).

I. Introduction

In recent years, the development of 5G/B5G/6G communication technologies and the increasing number of mobile devices have led to the explosive growth of network traffic [1]. Cisco predicts that by 2023, there will be 5.3 billion internet users with an annual compound growth rate of 6% [2]. On the other hand, advanced mobile services spawned by new communication technologies are constantly emerging, which generally have computation-intensive and latency-sensitive requirements, such as intelligent manufacturing, autonomous driving, augmented reality etc. To cope with the substantial volume of mobile traffic and meet diverse service demands, Mobile Edge Computing (MEC) has emerged as a promising technology. MEC provides rapid response to user requests and low-latency access to computing resources, so as to guarantee users' Quality of Service (QoS) and enhance service experience [1], [3].

In MEC networks, unpredictable user mobility and time-varying wireless channels often lead to frequent handovers in edge access networks. These factors can increase user-perceived delay, potentially causing service interruptions and significantly degrading the QoS for delay-sensitive services [4]. Therefore, as a user moves across Base Stations (BSs), services can dynamically migrate among edge servers to accommodate user mobility and meet dynamically changing service requests. However, frequent service migration introduces additional resource overhead, e.g., Wide Area Network (WAN) bandwidth, and increases network operating cost, e.g., network energy consumption [5]. This further undermines the utility of the MEC network, which already has high operating cost and significant waste of network resources due to the dense deployment of base stations [6]. Therefore, service migration considering cost-efficiency in MEC networks is a critical issue.

Since the cost-efficiency of service migration strategies is influenced by future traffic demands, an intuitive idea is to assist service migration by predicting future traffic demands. For instance, popular services could be migrated to areas with high traffic demands to accommodate an increased number of user requests. This can support service continuity, and also allows idle infrastructure to be put into sleep mode post-migration, thereby reducing network operation cost. With the development of Artificial Intelligence (AI) and distributed learning technologies, a promising approach is to leverage

Edge-AI and Federated Learning (FL) to establish a distributed learning architecture for intelligent service migration in MEC networks [7]–[10].

Service migration based on Edge-AI necessitates a substantial volume of data for the construction of an effective model. Recently, the emerging Digital Twins (DT) technology has been instrumental in this aspect. DT can capture the status information of physical entities using sensor data, create corresponding virtual objects, and accumulate a vast amount of real data in the digital space [11], [12]. This data can be exploited for model training, state analysis, and risk assessment in AI-enabled networks. Therefore, distributed DT have been integrated into MEC networks, sinking real-time data processing to the edge plane [13]. It enables edge servers to leverage large volumes of data for Deep Learning (DL)-based model training, thereby supporting traffic prediction-assisted cost-efficient service migration in Digital Twin-enabled MEC (DT-MEC) networks.

However, traffic prediction-assisted cost-efficient service migration in DT-MEC faces many challenges. 1) To predict future traffic demands, it is necessary to deal with complex and coupled spatio-temporal dependencies in mobile traffic, rooted in variegated user mobility, load fluctuations caused by social activities, and geographically heterogeneous mobile communications [14], [15]. Moreover, large-scale MEC networks need to address the challenges of data scarcity and imbalance in certain regions, which significantly undermine the performance of DL-based traffic prediction methods. 2) Given the distributed architecture of MEC networks and the difficulty in obtaining global information, it is necessary to design distributed and efficient collaborative learning schemes to enhance service migration efficiency. 3) Considering the increased network operating costs associated with service migration, as well as the heterogeneity, limited availability, and time-varying nature of edge resources, it is crucial to consider both resource allocation and network cost optimization during service migration. Therefore, in a large-scale MEC network lacking global information, how to design a cost-efficient service migration scheme that takes into account future traffic demands and reasonably migrates services, to ensure service continuity and reduce long-term network cost is a challenging problem.

To address the aforementioned issues, in this paper, we propose a traffic prediction-assisted cost-efficient service migration scheme based on federated Deep Reinforcement Learning (DRL). This scheme dynamically migrates services, switches BSs state, and allocates resources in response to future traffic demand, thereby aiming to reduce long-term network costs. Specifically, a Multi-order Spatio-temporal information integration-based distributed Traffic Prediction (MSTP) method is proposed to cope with the coupled spatio-temporal dependencies of mobile traffic. Moreover, and a Region Clustering-based edge Model Transfer (RCMT) method is proposed to tackle the imbalance of traffic data. Then, based on the predicted traffic demand, a Federated Cooperative cost-efficient Service Migration (FCSM) algorithm is proposed, which adaptively adjusts service migration strategies in response to the dynamic changes of the edge environment. To the best of our knowledge, this is the first paper that studies cost-efficient service migration considering future traffic demand in DT-MEC networks. Our contributions are four-fold as follows.

- Based on the analysis of real-world mobile traffic dataset, we propose a distributed traffic prediction approach that achieves efficient and low-cost mobile traffic prediction by integrating multi-order spatio-temporal information and edge model transfer.
- To achieve cost-efficient service migration, FCSM is proposed to enable online service migration in large-scale MEC networks, which facilitates multi-agent cooperative learning through intra-cluster parameter sharing and global asynchronous parameter aggregation.
- An analytical model is developed to demonstrate that the features extracted by MSTP contain multi-order spatio-temporal dependencies. Furthermore, the convergence of FCSM is theoretically analyzed and an upper bound of the time-averaged squared gradient norm is proved.
- Extensive simulations demonstrate that the proposed schemes achieve excellent prediction performance and reduce training cost. FCSM enables cost-efficient service migration, enhances users QoS, and significantly reduces long-term system cost in DT-MEC networks.

The rest of the paper is organized as follows. In Section II, we briefly discuss the related work. The system model and problem formulation are described in Section III. Then we describe the data analysis, and introduce the proposed MSTP and RCMT in Section IV. FCSM is described in detail in Section V, then we theoretically analyze MSTP and FCSM in Section VI. In Section VII, we evaluate the performance of MSTP and FCSM through extensive simulations and conclude our work in Section VIII.

II. Related Work

A. Traffic Prediction

Mobile traffic prediction is typically modeled as a time series prediction problem. Till date, several traffic prediction methods have been proposed, primarily including statistical analysis-based methods [16]–[19], traditional machine learning [20], and deep learning-based methods [10], [21]–[29]. For example, statistical analysis based methods include the classic Autoregressive Integrated Moving Average (ARIMA) [16], [17], entropy theory [18], and Holt-Winter's Exponential Smoothing [19]. Some researchers have focused their attention on traffic prediction methods based on traditional machine learning. In [20], Xu *et al.* proposed a scalable Gaussian process framework, as a class of Bayesian non-parametric machine learning models, which achieves large-scale wireless traffic prediction via a customized kernel function. These methods mainly focus on obtaining the temporal relationship of mobile traffic, and do not fully consider the complex spatial dependencies.

With the rapid increase of network traffic and the development of AI technology, traffic prediction methods based on DL with powerful representation ability have attracted more and more attention, such as Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), transfer learning, Convolutional Neural Network (CNN), etc. [10], [21]–[29]. These methods have brought strong vitality to mobile traffic prediction. As a Deep Neural Network (DNN) method, CNN is utilized to capture the spatial correlation of network traffic in some previous works [22], [25], [27]. On the other hand, some existing studies have considered spatial and temporal relationships for mobile traffic prediction [10], [24], [26]. For example, in [10], Wu *et al.* proposed Geographical and Semantic Spatial-Temporal Network to capture spatial relationship via CNN and then capture temporal relationship via LSTM. Crucially, the aforementioned studies mainly focused on centralized traffic prediction schemes, which introduce complex neural network parameters, high training costs, slow convergence, and also incur higher delay and communication overhead for data aggregation. Recently, distributed traffic prediction begins to attract attention. FL-based wireless traffic prediction scheme (FedDA) [29] is a representative work that focused on the aggregation of prediction models.

Although the above discussed works have preliminarily considered the importance of spatio-temporal dependencies for mobile traffic prediction, they mainly focus on adjacent spatial relationships and do not fully consider the coupled spatio-temporal dependencies among multi-order neighbors. Moreover, most existing traffic prediction methods require global information, and cannot cope with insufficient and imbalanced data in some regions of MEC networks. In contrast, we fully consider the spatio-temporal dependencies of mobile traffic by using the integrated multi-order spatio-temporal representation and leverage regional similarity to achieve high-efficiency and low-cost distributed traffic prediction.

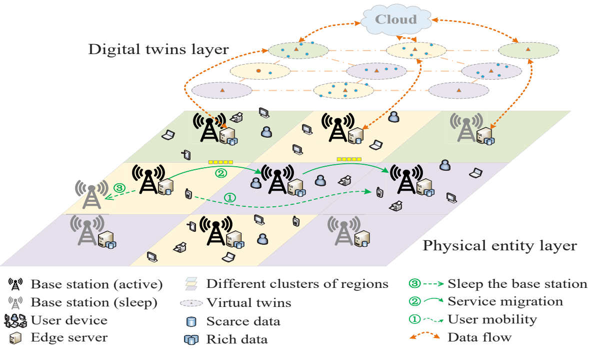


Fig. 1. An illustration of service migration and BS switching in a distributed DT-MEC network.

B. Service Migration

Service migration has also attracted more and more attention of researchers [4], [5], [30]–[37]. The main goal of service migration is to reduce the service response delay by following the user mobility. Existing research mainly includes migration cost optimization [30], [33], migration time optimization [5], [31], [32], [34]–[36] and migration success rate optimization [4], [37]. Some studies focus on the optimization of migration cost, including network bandwidth resource consumption, service and virtual machine installation cost, energy consumption, etc. For example, In [30], Taleb *et al.* constructed a Markov Decision Process (MDP)-based service migration algorithm to optimize service migration cost, and then proposed two alternative schemes including software defined networking technologies or the locator/identifier separation protocol to ensure service continuity and non-disruptive operation. In [33], Liang *et al.* considered the problem of minimizing migration cost and developed an effective solution based on relaxation and rounding. Some research focuses on optimizing service migration time to reduce user-perceived latency. In [5], Ouyang *et al.* considered the computing delay and communication delay of service requests, decomposed the long-term optimization problem of service delay into a series of real-time optimization problems, and designed centralized and distributed approximation schemes. In [35], Mukhopadhyay *et al.* proposed a priority-induced service migration minimization algorithm to optimize service migration time. In [36], Li *et al.* proposed a learning-based intelligent service migration algorithm to minimize the long-term average service delay for all users. Some work leverages DRL and federated learning to optimize latency and network resources in MEC [31]. Last but not least, the optimization of migration success rate aims to maximize the number of satisfied service requests of mobile users. In [4], Chen *et al.* considered the actual characteristics of multi-cell MEC networks, including the constraints of edge server storage and computing capacities, and developed a dynamic service migration method based on random rounding technology to maximize the number of requests accepted by edge servers. In [37], Tang *et al.* took into account user mobility and designed a group migration algorithm to maximize the number of tasks with deadlines.

Although some schemes have been proposed for service migration, existing studies have not taken into account the impact of future traffic demands on service migration decisions, potentially leading to low cost-efficiency during the migration process. Moreover, existing learning-based service migration studies have not sufficiently addressed the challenge posed by the lack of global information in distributed MEC networks, nor have they improved service migration efficiency through effective collaborative learning schemes. In this paper, we propose a federated cooperative multi-agent reinforcement learning framework to achieve cost-efficient service migration based on predicted traffic, which reduces the long-term system cost of distributed DT-MEC networks.

III. System Model and Problem Formulation

A. Network Model

The architecture of the proposed distributed DT-MEC network is shown in Fig. 1. Each mobile user is denoted by i, with $i \in U = \{1, 2, \cdots, |U|\}$, where U is the set of users. The BS and the MEC server are co-located and referred to as an edge node, denoted by j, where $j \in V = \{1, 2, \cdots, |V|\}$, and V is the set of BSs. We discretize time into multiple time slots, denoted by $t \in \Gamma = \{1, 2, \cdots, T\}$, where Γ is the set of time slots. In time slot t, the service request of user i is defined as: $q^i(t) = \{\iota^i(t), s^i(t), \omega^i(t), \sigma^i_\iota(t)\}$, which represents the service type, the service request size (in bits), the required computing intensity (in CPU cycles/bit), and the service storage size (in bits), respectively. The service type is represented as $\iota \in \boldsymbol{\iota} = \{1, 2, \cdots, |\boldsymbol{\iota}|\}$, and the location of user i is denoted as $p^i(t)$. In our proposed architecture, the distributed DT-MEC network enables the mapping of real network to digital models on edge servers, and provides extensive data and functional descriptions for physical entities to support intelligent service migration.

B. Energy Consumption Model

The energy consumption of a BS mainly includes fixed energy consumption in the active state and load-dependent energy consumption. The fixed energy consumption mainly relates to the rectifier, air conditioner, and microwave link, while the load-dependent energy consumption is mainly associated with digital signal processing, the transceiver, and the power amplifier [38]. Therefore, the energy consumption model of BS j is expressed as

$$e_j(t) = e_j^c + (e_j^{\max} - e_j^c)u_j(t), \tag{1}$$

where e_j^c is the fixed energy consumption when the BS is active, and $e_j^{\max}$ is the maximum energy consumption of the BS. The utilization rate of the BS is $u_j(t) = f_j(t)/F_j$, that is, the ratio of the traffic load to the total capacity of BS j in time slot t. The binary variable $x_j(t) \in \{0, 1\}$ denotes the state of BS j, that is, if j is in active state in time slot t, then $x_j(t) = 1$, otherwise, $x_j(t) = 0$. $V(t)$ denotes the set of BSs activated in time slot t. Therefore, the total energy consumption of the system in time slot t is expressed as $e(t) = \sum_{j \in V(t)} e_j(t)$.

C. Service Migration Model

Due to the unpredictable mobility of users and dynamically changing service requests, services need to be constantly migrated across edge servers to support service continuity and reduce user-perceived latency. Specifically, we define the binary variable for service deployment decision as $y_j^i(t) \in \{0, 1\}$. If the service of user i is deployed at edge server j in time slot t, then $y_j^i(t) = 1$, otherwise $y_j^i(t) = 0$. Therefore, the total cost of service migration $c_m(t)$ in time slot t is calculated as

$$c_m(t) = \sum_{i \in U(t)} \sum_{j' \in V(t-1)} \sum_{j \in V(t)} y_{j'}^i(t-1) y_j^i(t) c_{j'j}^i(t), \tag{2}$$

where $c_{j'j}^i(t) = \kappa \sigma_\iota^i(t) \eta_{j'j}$ and represents the cost of migrating the service from edge server j' to j. The cost of service migration is related to service size $\sigma_\iota^i(t)$ and the number of routing hops $\eta_{j'j}$. κ is the migration cost factor, and $U(t)$ denotes the set of users that have service requests in time slot t.

D. QoS Model

The QoS of the service can reflect the impact of sleeping some BSs and migrating service on service performance. In our system model, we mainly consider user-perceived delay, including transmission delay and computing delay. Regarding the transmission delay, the spectral efficiency between user i and BS j is

$$e_j^i(t) = \log_2(1 + \frac{P g_j^i(t)}{\chi^2}), \tag{3}$$

where $g_j^i(t)$ represents the channel gain, which is related to the distance between user i and BS j in time slot t. χ^2 is the noise power, and P is the user transmit power. Therefore, the uplink transmission rate is

$$r_j^i(t) = W_j b_j^i(t) e_j^i(t), \tag{4}$$

where W_j denotes the bandwidth resource capacity of BS j, and $b_j^i(t) \in [0, 1]$ denotes the spectrum fraction allocated to user i by the BS j in time slot t, that is, the proportion of allocated communication resources. Therefore, the transmission delay can be calculated by $d_j^i(t) = \frac{s^i(t)}{r_j^i(t)}$. Q_j denotes the computing capacity of BS j, and $z_j^i(t) \in [0, 1]$ is the computing resource allocation fraction, then the computing delay can be calculated by

$$\rho_j^i(t) = \frac{s^i(t) w^i(t)}{Q_j z_j^i(t)}. \tag{5}$$

Therefore, the overall QoS degradation cost of the system in time slot t is expressed as

$$c_q(t) = \sum_{i \in U(t)} \sum_{j \in V(t)} (d_j^i(t) + \rho_j^i(t)). \tag{6}$$

E. Problem Formulation

To avoid network instability and machine hardware damage caused by frequent BS switching, the total BS switching cost of the system is calculated as

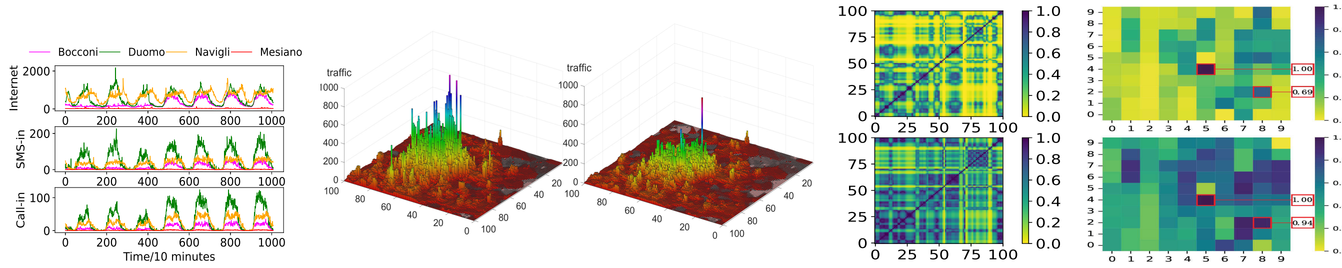
$$c_s(t) = \sum_{j \in V(t)} c_j(t) \mathcal{I}(x_j(t) > x_j(t-1)), \tag{7}$$

where $c_j(t)$ is the factor of base station switching cost, and $\mathcal{I}(*)$ is the indicator function and satisfies

$$\mathcal{I}(x_j(t) > x_j(t-1)) = \begin{cases} 1, \text{ if } x_j(t) > x_j(t-1), \\ 0, \text{ otherwise.} \end{cases} \tag{8}$$

The total cost of the system in time slot t consists of energy consumption, BS switching cost, service migration cost, and QoS degradation cost, which can be calculated as

$$c(t) = \phi_e e(t) + \phi_s c_s(t) + \phi_m c_m(t) + \phi_q c_q(t), \tag{9}$$



(a) (b) (c) (d)

Fig. 2. (a) Time series of mobile traffic for three events. (b) Snapshots of spatial distribution of mobile traffic for 10,000 regions at two timestamps. (c) Spatial correlations of 100 regions at two timestamps. (d) Spatial correlations of grid 45 (id 8155) with surrounding regions at two timestamps.

where $\phi_e, \phi_s, \phi_m, \phi_q \in [0,1]$ are the weight factors, and $\phi_e + \phi_s + \phi_m + \phi_q = 1$. Therefore, the cost optimization problem of the MEC network can be expressed as

$$\min_{\mathbf{x}(t),\mathbf{y}(t),\mathbf{z}(t),\mathbf{b}(t)} \quad c(t), \tag{10a}$$

$$\text{s.t.} \quad x_j(t), y_j^i(t) \in \{0,1\}, \forall i \in U, j \in V, \tag{10b}$$

$$z_j^i(t), b_j^i(t) \in [0,1], \forall i \in U, j \in V, \tag{10c}$$

$$y_j^i(t) \leq x_j(t), \tag{10d}$$

$$z_j^i(t) \leq y_j^i(t), \tag{10e}$$

$$b_j^i(t) \leq y_j^i(t), \tag{10f}$$

$$\sum_{j \in V(t)} y_j^i(t) = 1, \tag{10g}$$

$$\sum_{i \in U(t)} y_j^i(t) \sigma_\iota^i(t) \leqslant S_j, \tag{10h}$$

$$\sum_{i \in U(t)} y_j^i(t) z_j^i(t) = 1, \tag{10i}$$

$$\sum_{i \in U(t)} y_j^i(t) b_j^i(t) = 1, \tag{10j}$$

where BS switching and service deployment are integer decision variables, while communication and computing resource allocation are fractional variables. (10d) represents that the node providing the service must be active, (10e) and (10f) represent the allocation of computing and communication resources by the node providing services. Constraint (10g) indicates that the service request must be responded by one edge server. Constraints (10h)-(10j) represent storage, computing and communication resource capacity constraints. Note that to improve the user's QoS, computing and communication resources can be allocated to their maximum extent.

The above optimization problem faces the following challenges: 1) Accurate traffic prediction requires capturing complex and coupled spatio-temporal dependencies, and designing efficient distributed traffic prediction schemes in MEC networks is necessary. Moreover, overcoming the scarcity of traffic data in certain areas and reducing the cost of model training are also key issues. 2) The cost-efficient service migration problem is a mixed integer non-linear programming problem, which has been proven to be NP-hard [39]. Learning-based service migration schemes need to address the challenges brought by the lack of global information in distributed MEC networks. 3) In the distributed learning architecture, it is also a challenge to improve learning efficiency through effective collaborative learning schemes. Therefore, we propose a traffic prediction-assisted federated DRL framework for cost-efficient service migration in distributed DT-MEC networks.

IV. Distributed Traffic Prediction for DT-MEC Networks

In this section, we first analyze real-world mobile traffic datasets and present our research motivation for distributed traffic prediction. Our distributed traffic prediction method is shown in Fig. 3, which consists of two parts. Firstly, since the arrival of mobile traffic is strongly correlated with the coupled relationship between the temporal sequence and spatial distribution, we propose the MSTP method. Furthermore, to reduce model training cost and address the issue of data scarcity in certain regions, we design the RCMT method for distributed traffic prediction.

A. Data Analysis

We analyze two of the most representative and widely used datasets in the field of traffic prediction, the Telecom Italia Big Data Challenge datasets [40]. These are real cellular traffic datasets of the city of Milan and the Province of Trentino. These two real-world datasets contain traffic data from November 1st, 2013 to January 1st, 2014, and one traffic data record is generated every ten minutes. As multi-source datasets, they contain various types of communication data, including SMS-in activity, SMS-out activity, incoming call activity, outgoing call activity, and Internet traffic. The Milan dataset comprises 10,000 regions, while the Trentino dataset comprises 6,575 regions.

We sample and analyze the datasets, where Fig. 2(a) shows the traffic patterns of call-in events, SMS-in events, and Internet events for one week, and the three different traffic events show a certain periodicity and regularity on the time scale. Fig. 2(b) exhibits the spatial distribution of real traffic for 10,000 regions at two timestamps. It can be seen that the spatial distribution of network traffic is imbalanced and shows significant differences at different times. Furthermore, we use the Dynamic Time Warping (DTW) technique [41] to analyze the similarity of mobile traffic within different regions, and the spatial correlations of 100 regions at two timestamps are shown in Fig. 2(c). We can observe that the spatial correlation of mobile traffic varies at different timestamps, which illustrates the tightly coupled relationships between the temporal and spatial correlations of various regions. The coupled spatio-temporal relationships can affect the accuracy of mobile traffic prediction, but are often ignored in previous works.

We randomly select a region (45th grid) and analyze its spatial correlation with its surrounding regions at two timestamps, as shown in Fig. 2(d). We can observe that mobile traffic in different regions may exhibit a high degree of similarity. For example, the correlations of the 45th grid and the 28th grid at two timestamps are 0.69 and 0.94, respectively, which implies that there is a high consensus on traffic patterns in some regions. The analysis in [14] shows that urban mobile traffic can be characterized by some basic time-domain patterns, which correspond to different functional areas, i.e., residential urban, office urban, transportation urban, touristic and leisure urban. Besides, we can also observe that there is a data scarcity issue in some areas in Fig. 2(b). This issue could seriously affect the performance of DL-based traffic prediction methods, especially those distributed prediction methods that lack global information. An intuitive motivation is that regions can be clustered according to the spatio-temporal similarities in mobile traffic, and then the edge prediction models of the data-rich regions could be transferred to other regions. This approach can address the data scarcity problem and reduce the training cost of the prediction model in certain regions.

B. Multi-order Spatio-temporal Information Integration-based Distributed Traffic Prediction

To clearly represent the network traffic data, the coverage area of the DT-MEC network is divided into $|V|$ grid regions, with the region indexed by n. In each grid region, the corresponding BS is deployed. The traffic volume of region n in time slot t is expressed as $f_n(t) = \sum_{i\in U(t)\cap p^i(t)\in U_n(t)} s^i(t)$, where $U_n(t)$ is the set of users within region n in time slot t. Therefore, the traffic matrix $\boldsymbol{F}(t) \in \mathbb{R}^{|V|\times 1}$ for time slot t is expressed as $\boldsymbol{F}(t) = [f_1(t), ..., f_n(t), ..., f_{|V|}(t)]^\top$. The traffic data spanning T time slots is denoted as $\boldsymbol{F}_{1\sim T} = [\boldsymbol{F}(1), \boldsymbol{F}(2), \dots, \boldsymbol{F}(T)] \in \mathbb{R}^{|V|\times T}$. MSTP benefits from virtual twins deployed on edge servers, which provide a large amount of real traffic data for the training of prediction models.

Traffic prediction in the DT-MEC network is performed in a distributed manner, where the perceptible range of each region contains $|\bar{N}|$ regions, $\bar{N} \subset V$. Denote the traffic matrix of $|\bar{N}|$ regions in time slot t as $\bar{\boldsymbol{F}}(t)$, which is a submatrix of the global traffic matrix. Given the truncated historical traffic data of the previous T time slots $\bar{\boldsymbol{F}}(t-T+1), \dots, \bar{\boldsymbol{F}}(t)$, the sliding window model is used to predict future traffic in region n, which is expressed as

$$\tilde{\boldsymbol{F}}(t+1) = \underset{\tilde{\boldsymbol{F}}(t+1)}{\arg\min}\, L_{\mathrm{p}}\Big(\bar{\boldsymbol{F}}(t+1), \tilde{\boldsymbol{F}}(t+1), \theta_p | \bar{\boldsymbol{F}}(t-T+1), \dots, \bar{\boldsymbol{F}}(t)\Big), \tag{11}$$

where $\tilde{\boldsymbol{F}}(t+1)$ is the predicted traffic for time slot $t+1$. Note that the notation n is omitted for simplicity. This paper proposes a custom-tailored DL method based on spatio-temporal information integration, and our goal is to minimize the loss L_{p} between the predicted traffic value $\tilde{\boldsymbol{F}}(t+1)$ and the real value $\bar{\boldsymbol{F}}(t+1)$ by learning the parameter θ_p.

The spatial correlation of mobile traffic is not only related to geographical location, but also closely associated with user mobility, traffic patterns, service types, etc. Therefore, traditional distance-based spatial correlation calculation methods cannot capture the true spatial relationship of mobile traffic among regions. As mobile traffic is a time series data, we use DTW to measure the similarity of traffic sequences across different regions. For $|\bar{N}|$ regions in the DT-MEC network, indexed by $n \in \bar{N}$, the similarity between any two regions n and n' is calculated by $s_{nn'} = e^{-\mathrm{DTW}(\boldsymbol{f}_n, \boldsymbol{f}_{n'})}$, where $\boldsymbol{f}_n = [f_n(1), f_n(2), \dots, f_n(T)]$. The similarity of region pairs can be constructed as a similarity matrix $\boldsymbol{S}_{\mathrm{DTW}} \in \mathbb{R}^{|\bar{N}|\times|\bar{N}|}$, where $s_{nn'}$ is the element in the nth row and n'th column of $\boldsymbol{S}_{\mathrm{DTW}}$.

On the other hand, the number of neighbor nodes may vary due to different geographic distributions. We use Graph Convolutional Network (GCN) units to capture the spatial correlation of mobile traffic, which is not limited to a fixed number of neighbors or adjacent regions. Then, we propose a novel traffic prediction method MSTP that integrates multi-order spatio-temporal dependencies of neighbors, as illustrated in Fig. 3. Our motivation is that the information learned by each layer of the GCN unit is different, and MSTP preserves spatial correlations and aims to capture as much spatio-temporal dependencies as possible for effective traffic prediction. MSTP comprises a spatial network and a temporal network.

Spatial network: For $n \in \bar{N}$, select the first λ similarity region n' from $\boldsymbol{S}_{\mathrm{DTW}}$ to establish the neighbor relationship with n, which is expressed as

$$a_{nn'} = \begin{cases} 1, \text{ if } n \text{ and } n' \text{ are neighbors,} \\ 0, \text{ otherwise.} \end{cases} \tag{12}$$

Furthermore, considering the symmetry of the neighbor relationship in an undirected graph, if $a_{nn'} = 1$, we set $a_{n'n} = 1$. Then an adjacency matrix $\boldsymbol{A} \in \{0,1\}^{|\bar{N}|\times|\bar{N}|}$ can be constructed for the set of $|\bar{N}|$ regions. To integrate neighbor information and regional self-information during the process of graph convolution operations, the adjacency matrix $\boldsymbol{A}$ is added to the identity matrix to obtain a self-looped adjacency matrix $\tilde{\boldsymbol{A}} \triangleq \boldsymbol{A} + \boldsymbol{I}$. Furthermore, the degree matrix $\tilde{\boldsymbol{D}}$ corresponding to $\tilde{\boldsymbol{A}}$ is constructed, which is a diagonal matrix and its diagonal element value $d_n \triangleq \sum_{n'=1}^{|\bar{N}|} a_{nn'}$ is the degree of the region n.

Consider that the data of t time slots is independently processed by multi-layer GCN unit to extract the spatial information of each time slot. Therefore, for the data of time slot t, the graph convolution operation through the ℓth layer GCN unit is expressed as

$$\boldsymbol{G}^{(\ell)}(t) = \sigma\Big(\tilde{\boldsymbol{D}}^{-\frac{1}{2}}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-\frac{1}{2}}\boldsymbol{G}^{(\ell-1)}(t)\boldsymbol{W}_g^{(\ell-1)}\Big), \tag{13}$$

where $\sigma(*)$ is the activation function, and $\boldsymbol{G}^{(\ell-1)}(t)$, $\boldsymbol{G}^{(\ell)}(t)$, $\boldsymbol{W}_g^{(\ell-1)}$ are the input representation, output representation,

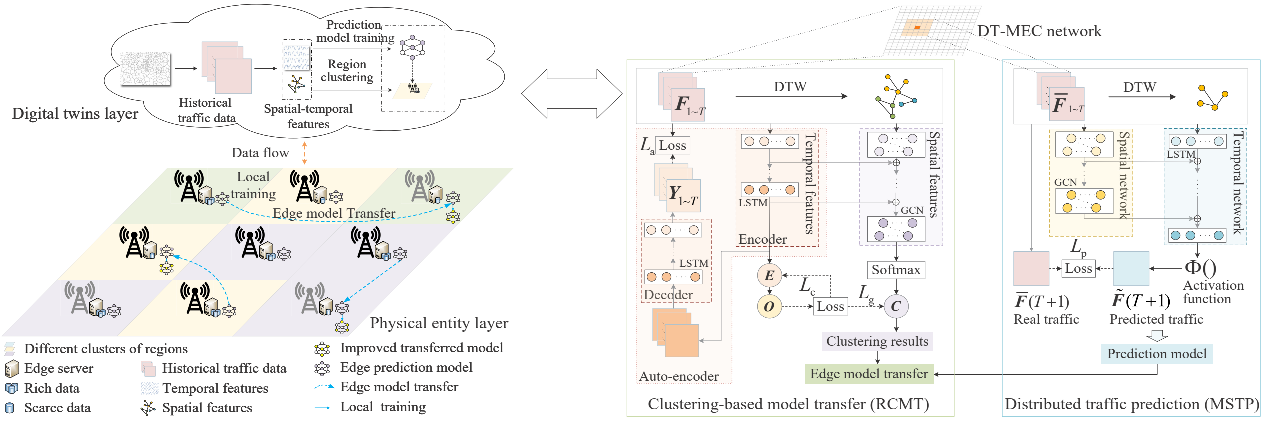


Fig. 3. An illustration of distributed traffic prediction in the DT-MEC network.

and parameter matrix, respectively. GCN can learn the representation of neighbor regions through the standard adjacency matrix $\hat{\boldsymbol{A}} \triangleq \tilde{\boldsymbol{D}}^{-\frac{1}{2}} \tilde{\boldsymbol{A}} \tilde{\boldsymbol{D}}^{-\frac{1}{2}}$. Note that for the first layer of GCN, i.e. when $\ell = 1$, the original input data is $\boldsymbol{G}^{(0)}(t) = \bar{\boldsymbol{F}}(t)$.

Temporal network: Traffic prediction model requires strong capabilities to extract long-range dependencies and handle nonlinear data [9]. In our proposed MSTP, the LSTM unit in the temporal network is used to capture dependencies at different time scales. As a special variant of RNN, LSTM mitigates the vanishing or exploding gradient problem of traditional RNN through improved internal unit. Specifically, LSTM includes an input gate, a forget gate and an output gate, which control the processes of information input, transmission and updating. The LSTM operation of the ℓth layer at time step t is expressed as

$$\boldsymbol{L}^{(\ell)}(t) = \Phi\left(\boldsymbol{L}^{(\ell)}(t-1), \tilde{\boldsymbol{L}}^{(\ell-1)}(t); \boldsymbol{W}_r^{(\ell-1)}\right), \tag{14}$$

where $\Phi(*)$ is the update function, $\boldsymbol{L}^{(\ell)}(t-1)$ and $\tilde{\boldsymbol{L}}^{(\ell-1)}(t)$ are input representations, $\boldsymbol{L}^{(\ell)}(t)$ is the output representation, and $\boldsymbol{W}_r^{(\ell-1)}$ is the parameter matrix, respectively.

Then, we design a multi-order feature integration method to preserve more spatial correlations and capture spatio-temporal dependencies. The spatial information captured by each layer of the GCN unit is passed to the corresponding LSTM unit, aiming to aggregate the spatial and temporal information and obtain a more comprehensive representation of mobile traffic,

$$\tilde{\boldsymbol{L}}^{(\ell-1)}(t) = \varpi \boldsymbol{L}^{(\ell-1)}(t) + (1-\varpi)\boldsymbol{G}^{(\ell-1)}(t), \tag{15}$$

where $\varpi \in (0, 1]$ is the feature integration weight factor, which is used to control the proportion of temporal and spatial information. The improved prediction model can simultaneously learn temporal and spatial information at each layer. Note that for the first layer of LSTM unit, i.e. when $\ell = 1$, the original input data is $\boldsymbol{L}^{(0)}(t) = \bar{\boldsymbol{F}}(t)$. Therefore, MSTP aggregates spatio-temporal information from L layers to obtain a more comprehensive and powerful representation of mobile traffic, thereby facilitating distributed traffic prediction.

C. Structural Deep Region Clustering-based Edge Model Transfer

In distributed DT-MEC networks, due to geographic distribution, user mobility, the effectiveness of data collection, etc., there are insufficient and imbalanced data problems in some regions, and these regions cannot train effective prediction models. Therefore, we propose the RCMT method, which differs from existing region clustering methods that are based on Euclidean distance or independent spatio/temporal relationships. RCMT integrates structured information of mobile traffic into regional deep clustering, which fully considers the spatio-temporal dependencies of mobile traffic.

For deep region clustering and efficient edge model transfer, it is crucial to learn effective data representation. In RCMT, we propose a dual-supervised LSTM autoencoder and GCN model to introduce structural information of mobile traffic. The framework of RCMT is shown in Fig. 3. First, since mobile traffic is time-series data, we use the LSTM autoencoder unit to learn temporal features. The encoder network is a multi-layer LSTM model, and the operation of the ℓth layer encoder at the t time step is expressed as

$$\boldsymbol{E}^{(\ell)}(t) = \Psi\left(\boldsymbol{E}^{(\ell)}(t-1), \boldsymbol{E}^{(\ell-1)}(t); \boldsymbol{W}_e^{(\ell-1)}\right), \tag{16}$$

where $\Psi(*)$ is encoding operation. The encoder contains L layer network, and the output of the last layer is a tensor $\boldsymbol{E}_{1\sim T}^{(L)} = \left[\boldsymbol{E}^{(L)}(1), \ldots, \boldsymbol{E}^{(L)}(t), \ldots, \boldsymbol{E}^{(L)}(T)\right]$, which is the input of the decoder. Note that when $\ell = 1$, its input is the raw traffic data $\boldsymbol{E}^{(0)}(t) = \boldsymbol{F}(t)$.

The decoder network also contains L layers of LSTM to reconstruct the traffic data. The operation of the ℓth layer of the decoder at the t time step can be expressed as

$$\hat{\boldsymbol{E}}^{(\ell)}(t) = \hat{\Psi}\left(\hat{\boldsymbol{E}}^{(\ell)}(t-1), \hat{\boldsymbol{E}}^{(\ell-1)}(t); \boldsymbol{W}_d^{(\ell-1)}\right). \tag{17}$$

When $\ell = 1$, the input of the decoder is the output of the last layer of the encoder. When $\ell = L$, the last layer of the decoder can get a tensor $\boldsymbol{Y}_{1\sim T} \triangleq \hat{\boldsymbol{E}}_{1\sim T}^{(L)} = \left[\hat{\boldsymbol{E}}^{(L)}(1), \ldots, \hat{\boldsymbol{E}}^{(L)}(t), \ldots, \hat{\boldsymbol{E}}^{(L)}(T)\right]$, which has the same size as the input of the encoder. The optimization objective of the LSTM autoencoder is to minimize the loss between the input representation $\boldsymbol{F}_{1\sim T}$ and the reconstructed output representation $\boldsymbol{Y}_{1\sim T}$, and the loss function is $L_{\mathrm{a}}(\boldsymbol{W}_e, \boldsymbol{W}_d) = \|\boldsymbol{Y}_{1\sim T} - \boldsymbol{F}_{1\sim T}\|$, where $\| * \|$ is the Frobenius norm in this paper.

The LSTM autoencoder can capture the feature representation of the traffic data itself, but ignores the relationships among data samples. Therefore, in our proposed RCMT, the GCN unit is used to obtain the structural features of network traffic, that is, the spatial relationship of network traffic. Note that in region clustering, for each region, select the first ξ similar regions to construct the neighbor relationships, and then we can get the adjacency relationship graph. Denote the constructed self-looped adjacency matrix and degree matrix as $\tilde{\boldsymbol{H}}, \tilde{\boldsymbol{Q}} \in \mathbb{R}^{|V|\times|V|}$.

For the GCN unit, based on the constructed adjacency matrix $\tilde{\boldsymbol{H}}$ and its corresponding degree matrix $\tilde{\boldsymbol{Q}}$, the GCN operation is expressed as

$$\boldsymbol{C}^{(\ell)}(t) = \sigma\Big(\tilde{\boldsymbol{Q}}^{-\frac{1}{2}}\tilde{\boldsymbol{H}}\tilde{\boldsymbol{Q}}^{-\frac{1}{2}}\tilde{\boldsymbol{C}}^{(\ell-1)}(t)\boldsymbol{W}_c^{(\ell-1)}\Big), \tag{18}$$

where $\boldsymbol{W}_c^{(\ell-1)}$ is the parameter matrix of the ℓth layer of the GCN. To preserve more diverse representation information, the traffic features learned by each layer of the autoencoder are propagated to the GCN, and a more comprehensive representation is obtained by integrating the spatio-temporal features of mobile traffic

$$\tilde{\boldsymbol{C}}^{(\ell-1)}(t) = \epsilon\boldsymbol{C}^{(\ell-1)}(t) + (1-\epsilon)\boldsymbol{E}^{(\ell-1)}(t), \tag{19}$$

where $\boldsymbol{C}^{(\ell-1)}(t)$ is the output representation of the $(\ell-1)$th layer of GCN, $\boldsymbol{E}^{(\ell-1)}(t)$ is the output representation of the $(\ell-1)$th layer of the LSTM encoder, and $\tilde{\boldsymbol{C}}^{(\ell-1)}(t)$ is the integrated representation learned by GCN and LSTM encoder. $\epsilon \in (0,1]$ is the integration factor to control the weight of spatio-temporal information in RCMT.

The output $\boldsymbol{C}^{(L)}(t)$ obtained through the Lth layer of GCN is then passed through a softmax multi-classifier to obtain the probability distribution matrix $\boldsymbol{C} = \mathrm{softmax}\big(\sum_{t=1}^{T}\omega_t\sigma(\tilde{\boldsymbol{Q}}^{-\frac{1}{2}}\tilde{\boldsymbol{H}}\tilde{\boldsymbol{Q}}^{-\frac{1}{2}}\tilde{\boldsymbol{C}}^{(L)}(t)\boldsymbol{W}_c^{(L)})\big)$, where ω_t is the weight coefficient, which can be set based on the importance of data in different time slots. This further obtains the spatial and temporal similarities of regions and achieves region clustering. c_{nk} is the element located in the nth row and kth column of the probability distribution matrix $\boldsymbol{C}$, which represents the probability that region n belongs to cluster k, $1 \le k \le K$. Therefore, the cluster label assigned to region n is $\psi_n = \arg\max_k c_{nk}$.

Since there is no supervision information for the RCMT model, we employ a dual-supervised method to train both the LSTM autoencoder and GCN. A supervised target is constructed from the output of the LSTM encoder, and the loss function of the model is designed based on the supervised target. Then, the parameters are learned in a self-supervised manner. Inspired by the work of unsupervised deep embedding [42], we initialize K cluster centers $\{\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_K\}$ based on the K-means algorithm according to the extracted representation $\boldsymbol{E}^{(L)}(T)$ of the LSTM autoencoder. Using the t-distribution [42] as the kernel function, the similarity between the region representation $\boldsymbol{f}_n$ and the cluster center representation $\boldsymbol{\eta}_k$ is $\big(1 + \|\boldsymbol{f}_n - \boldsymbol{\eta}_k\|^2/\varrho\big)^{-\frac{\varrho+1}{2}}$, where ϱ is the degrees of freedom of the t-distribution. Define e_{nk} as the probability of region n being clustered into class k, which can be calculated by $e_{nk} = \frac{(1+\|\boldsymbol{f}_n-\boldsymbol{\eta}_k\|^2/\varrho)^{-\frac{\varrho+1}{2}}}{\sum_{k'}(1+\|\boldsymbol{f}_n-\boldsymbol{\eta}_{k'}\|^2/\varrho)^{-\frac{\varrho+1}{2}}}$, where $n, n' \in \{1, 2, \cdots, |V|\}$, and $k, k' \in \{1, 2, \cdots, K\}$. Then we can have the soft assignment probability matrix $\boldsymbol{E} \in (0,1)^{|V|\times K}$. Furthermore, to improve the supervised training of e_{nk}, an optimized target soft assignment probability o_{nk} is introduced

$$o_{nk} = \frac{e_{nk}^2/\sum_{n'} e_{n'k}}{\sum_{k'}(e_{nk'}^2/\sum_{n'} e_{n'k'})}. \tag{20}$$

Then the target soft assignment matrix $\boldsymbol{O} \in (0,1)^{|V|\times K}$ can be obtained. Since KL-divergence can be used to measure the difference between two probability distributions, the KL-divergence loss function of $\boldsymbol{E}$ for $\boldsymbol{O}$ is defined as $L_{\mathrm{c}}(\boldsymbol{W}_e) = KL(\boldsymbol{O}\|\boldsymbol{E}) = \sum_{n=1}^{|V|}\sum_{k=1}^{K} o_{nk}\log\frac{o_{nk}}{e_{nk}}$. After performing $\mathrm{softmax}()$ on the output of GCN, we get the probability distribution $\boldsymbol{C}$. The target distribution $\boldsymbol{O}$ constructed in the LSTM encoder is used as the target distribution for GCN. Therefore, the KL-divergence loss function of $\boldsymbol{C}$ for $\boldsymbol{O}$ is $L_{\mathrm{g}}(\boldsymbol{W}_e, \boldsymbol{W}_c) = KL(\boldsymbol{O}\|\boldsymbol{C}) = \sum_{n=1}^{|V|}\sum_{k=1}^{K} o_{nk}\log\frac{o_{nk}}{c_{nk}}$. Therefore, the total loss function for structural deep region clustering is $\mathcal{L} = L_{\mathrm{a}} + \varphi_{\mathrm{c}}L_{\mathrm{c}} + \varphi_{\mathrm{g}}L_{\mathrm{g}}$, which includes the LSTM autoencoder loss L_{a}, the clustering loss L_{c} and the GCN loss L_{g}. Here, φ_{c} and φ_{g} are weight factors. Next, we give two important gradient calculations for parameter updates. According to Stochastic Gradient Descent (SGD) and backpropagation [43], the gradient of the parameter update for the cluster center $\boldsymbol{\eta}_k$ is

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{\eta}_k} = \varphi_{\mathrm{c}}\left(\frac{\varrho+1}{\varrho}\right)\sum_{n}^{|V|}\Bigg[(o_{nk}-e_{nk})(\boldsymbol{f}_n-\boldsymbol{\eta}_k) \left(1+\frac{\|\boldsymbol{f}_n-\boldsymbol{\eta}_k\|^2}{\varrho}\right)^{-1}\Bigg]. \tag{21}$$

The gradient of parameters $\boldsymbol{W}_e$ for the LSTM encoder network can be calculated by

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{W}_e} = \frac{\partial L_{\mathrm{a}}}{\partial\boldsymbol{W}_e} + \varphi_{\mathrm{c}}\left(\frac{\varrho+1}{\varrho}\right)\sum_{k}^{K}\Bigg[(e_{nk}-o_{nk})(\boldsymbol{f}_n-\boldsymbol{\eta}_k) \left(1+\frac{\|\boldsymbol{f}_n-\boldsymbol{\eta}_k\|^2}{\varrho}\right)^{-1}\Bigg]\frac{\partial\boldsymbol{f}_n}{\partial\boldsymbol{W}_e} - \varphi_{\mathrm{g}}\frac{o_{nk}}{c_{nk}}\frac{\partial c_{nk}}{\partial\boldsymbol{W}_e}. \tag{22}$$

Furthermore, based on the results of region clustering, we can transfer the well-trained edge models in the data-rich regions to similar regions. On the one hand, it provides a refined initial state for the prediction model in target regions, rather than parameters that are randomly initialized, thereby alleviating the problem of insufficient data in certain regions. On the other hand, it can also reduce the training cost of the prediction model. Since graph regularization in RCMT can introduce structural information into the model, the representation learned through this process incorporates the multi-order spatio-temporal feature of mobile traffic.

V. Federated DRL Based Cost-efficient Service Migration

The predicted traffic can guide service migration considering cost efficiency and resource optimization in the DT-MEC network. In this section, we first perform a problem transformation of the joint optimization model, then we describe the model training process based on federated multi-agent DRL.

A. Problem Transformation

Learning-based algorithms can dynamically adapt to the network environment with the complex state space, but centralized DRL methods cannot deal with the exponentially increasing system complexity as the number of base stations and users increases. Therefore, we first remodel the proposed joint optimization problem as a multi-agent extension of MDP, denoted as ($\mathcal{J}$, $\mathcal{S}$, $\{\mathcal{A}_j\}$, $\{\mathcal{O}_j\}$, $\mathcal{P}$, $\mathcal{C}$, $\mathcal{O}$), where $\mathcal{J}$ and $\mathcal{S}$ represent agents and system state, respectively. $\mathcal{A}_j$ represents the possible actions of $j \in V$, and $\mathcal{O}_j$ represents the observation of $\mathcal{S}$ by j. $\mathcal{P}$ is the joint state transition probability function, $\mathcal{C}$ is the joint cost function, $\mathcal{O}$ is the joint observation function. Note that $\mathcal{P}$, $\mathcal{C}$ and $\mathcal{O}$ need to be modeled by the interaction between agents and the environment.

Since agents cannot observe actual traffic at the beginning of time slot t, we use the predicted traffic as state observation. Then, base station switching can be executed based on the predicted traffic. Subsequently, service migration, bandwidth resource allocation, and computing resource allocation are performed after observing the real user requests.

State space: The state space is defined as $< s^c(t), s^m(t) >$, where the joint state of BS switching at time slot t is defined as $s^c(t) = (\tilde{f}(t), x(t-1))$, which includes the predicted traffic at the beginning of t and the active/sleep state of BSs at $t-1$. The joint state of service migration is defined as

$$\begin{aligned} s^m(t) =& (\tilde{x}(t), q(t), p(t)) \\ =& \big(\tilde{x}_1(t), \tilde{x}_2(t), ..., \tilde{x}_{|V|}(t), \iota^1(t), \iota^2(t), ..., \iota^{|U|}(t), \\ & s^1(t), s^2(t), ..., s^{|U|}(t), w^1(t), w^2(t), ..., w^{|U|}(t), \\ & \sigma^1(t), \sigma^2(t), ..., \sigma^{|U|}(t), p^1(t), p^2(t), ..., p^{|U|}(t)\big), \end{aligned} \tag{23}$$

which includes all BS active/sleep pre-operations according to the predicted traffic, all service requests and locations of users.

Observation space: The observation space is defined as $< o_j^c(t), o_j^m(t) >$. Since the proposed architecture does not require the coordination of a central controller and frequent information exchange, each agent only needs to observe local information. The observation of BS switching can be described as $o_j^c(t) = (\tilde{f}_j(t), x_j(t-1))$, which includes the predicted local traffic and the active/sleep state at $t-1$. The observation of service migration can be described as

$$\begin{aligned} o_j^m(t) = & \Big(\tilde{x}_j(t), \iota_j^1(t), \iota_j^2(t), ..., \iota_j^{|U|}(t), \\ & s_j^1(t), s_j^2(t), ..., s_j^{|U|}(t), w_j^1(t), w_j^2(t), ..., w_j^{|U|}(t), \\ & \sigma_j^1(t), \sigma_j^1(t), ..., \sigma_j^{|U|}(t), p_j^1(t), p_j^2(t), ..., p_j^{|U|}(t) \ \Big), \end{aligned} \tag{24}$$

which includes local BS active/sleep pre-operations performed according to the predicted traffic, local service requests and locations of users.

Action space: The action space is defined as $< a_j^c(t), a_j^m(t) >$. Each agent selects an action from the action space to execute based on local observation and policy π_j, that is, $a_j^c(t)$ and $a_j^m(t)$, where $a_j^c(t)$ is the local BS active/sleep operation $x_j(t)$, and $a_j^m(t)$ can be expressed as

$$\begin{aligned} a_j^m(t) =& \Big(y_j^1(t), y_j^2(t), ..., y_j^{|U|}(t), z_j^1(t), z_j^2(t), ..., z_j^{|U|}(t), \\ & b_j^1(t), b_j^2(t), ..., b_j^{|U|}(t)\Big). \end{aligned} \tag{25}$$

Reward function: To evaluate the reward of action exploration more accurately and mitigate the cost fluctuation caused by traffic changes, the reward function is defined as $R(t) = -\frac{c(t)}{|U(t)|}$.

B. Federated Multi-Agent DRL Based Model Training

To handle continuous decision variables, we utilize federated Multi-Agent Deep Deterministic Policy Gradient (MADDPG) architecture to solve the multi-agent Markov decision problem in this paper. First, the agent observes the state of the environment, then randomly selects transitions from the experience buffer to learn the optimal policy π_j^*, and each agent contains Actor and Critic components.

Experience replay can improve the learning efficiency and stability of MADDPG, where a transition can be expressed as a tuple $\{s(t), a(t), r(t), s'(t)\}$. In each round of iteration, each agent j randomly samples transitions of B_j size from the experience buffer, and updates the parameters $\boldsymbol{\theta}_j^\mu$ and $\boldsymbol{\theta}_j^Q$ of the evaluation network through training. Specifically, the parameters of the Actor evaluation network are updated by maximizing the policy objective function

$$J\left(\boldsymbol{\theta}_j^\mu\right) = \mathbb{E}\left[Q_j\left(s_j, a_j\right) \mid a_j = \mu_j\left(o_j\right)\right], \tag{26}$$

where $Q_j(s_j, a_j) = \mathrm{E}[\sum_{\tau=0}^{\infty} \gamma^\tau r_j(t+\tau) | \pi_j, s_j = s_j(t), a_j = a_j(t)]$, and γ is the penalty factor. $\mu_j(*)$ is the evaluation network function of the actor, which represents the deterministic policy. The Critic updates the parameters of the evaluation network by minimizing the loss

$$L\left(\boldsymbol{\theta}_j^Q\right) = \mathbb{E}\left[\left(Q_j\left(s_j, a_j\right) - \left(r_j + \gamma Q_j'\left(s_j', a_j'\right)\right)\right)^2\right]. \tag{27}$$

where $Q_j'(*)$ is the state-action function for the target network, which has the same structure as the original evaluation network. The target network is constructed to evaluate future action and reward, which can improve the stability and convergence of the training process. As the evaluation network parameters $\boldsymbol{\theta}_j^\mu$ and $\boldsymbol{\theta}_j^Q$ are updated, the parameters of the target network $\boldsymbol{\theta}_j^{\mu'}$ and $\boldsymbol{\theta}_j^{Q'}$ can be softly updated in a weighted manner

$$\boldsymbol{\theta}_j^{\mu'} = \kappa_j^a \boldsymbol{\theta}_j^\mu + \left(1 - \kappa_j^a\right) \boldsymbol{\theta}_j^{\mu'}, \tag{28}$$

$$\boldsymbol{\theta}_j^{Q'} = \kappa_j^c \boldsymbol{\theta}_j^Q + \left(1 - \kappa_j^c\right) \boldsymbol{\theta}_j^{Q'}, \tag{29}$$

where κ_j^a and κ_j^c are parameter update weights. For the DT-MEC network, we propose FL-based model training to update parameters, which can improve learning efficiency and reduce bandwidth resource cost caused by information exchange while protecting data privacy. It mainly includes intra-cluster parameter sharing and global asynchronous parameter aggregation.

Intra-cluster parameter sharing: Based on the model parameter $\boldsymbol{\theta}^{\tau}$ delivered by the central server, intra-cluster parameter sharing is performed after completing H rounds of local iterations within the cluster. $\boldsymbol{\Theta}_k^{\tau,g} = [\boldsymbol{\theta}_{k,1}^{\tau,g}, \ldots, \boldsymbol{\theta}_{k,N_k}^{\tau,g}]$ denotes the parameter vector of the agents in cluster k after g rounds of intra-cluster parameter sharing based on $\boldsymbol{\theta}^{\tau}$. $\boldsymbol{\Theta}_k^{\tau,g,h} = [\boldsymbol{\theta}_{k,1}^{\tau,g,h}, \ldots, \boldsymbol{\theta}_{k,N_k}^{\tau,g,h}]$ denotes the parameter vector of the agents in cluster k after h rounds of local iterations based on $\boldsymbol{\Theta}_k^{\tau,g}$. For the hth iteration of the local SGD, the parameter update of agent j in cluster k can be expressed as

$$\boldsymbol{\theta}_{k,j}^{\tau,g,h} = \boldsymbol{\theta}_{k,j}^{\tau,g,h-1} - \eta_k \nabla J \left(\boldsymbol{\theta}_{k,j}^{\tau,g,h-1} \right), \tag{30}$$

where η_k is the weight of the local parameter update. Then the gth intra-cluster parameter sharing can be expressed as $\boldsymbol{\Theta}_k^{\tau,g} = \boldsymbol{\Theta}_k^{\tau,g-1}\mathbf{W}$ [44], where $\mathbf{W}$ is defined as

$$\mathbf{W} = \begin{bmatrix} w & \frac{1-w}{N_k-1} & \cdots & \frac{1-w}{N_k-1} \\ \frac{1-w}{N_k-1} & w & \cdots & \frac{1-w}{N_k-1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1-w}{N_k-1} & \frac{1-w}{N_k-1} & \cdots & w \end{bmatrix}, \tag{31}$$

where $w \in [0,1]$ is the weight of agent j updating its own parameters in intra-cluster parameter sharing. Note that intra-cluster parameter sharing is based on the clustering results in IV-C, which collaboratively updates the model parameters within the cluster to improve the efficiency of parameter learning.

Global asynchronous parameter aggregation: After G rounds of intra-cluster parameter sharing in cluster k, the cluster parameters are transmitted to the central server for the global asynchronous parameter aggregation in the τth epoch,

$$\boldsymbol{\theta}^{\tau} = (1 - \alpha_{\tau'}^{\tau})\boldsymbol{\theta}^{\tau-1} + \alpha_{\tau'}^{\tau}\boldsymbol{\theta}_k^{\tau'}, \tag{32}$$

where $\alpha_{\tau'}^{\tau} \in (0,1)$ is the weight parameter for global aggregation, and τ' is the last epoch of cluster k participating in global aggregation, thus $\tau - \tau'$ represents the staleness in the global epochs for the parameters of cluster k. When a cluster participates in global aggregation, the average of all agent parameters in that cluster is used as the cluster parameter $\boldsymbol{\theta}_k^{\tau'} = \frac{1}{N_k}\sum_{j=1}^{N_k}\boldsymbol{\theta}_{k,j}^{\tau',G}$, and then the global asynchronous parameter aggregation is performed. The pseudocode of FCSM is shown in Algorithm 1. Each agent independently explores strategies based on local observations, samples from the experience buffer and updates its own Actor network and Critic network parameters through learning. To minimize the long-term cost of the DT-MEC system, the total reward is shared among cooperative agents. On the other hand, FCSM updates network parameters through intra-cluster parameter sharing among agents, as shown in (31). FCSM does not require agents to share input state information, but rather only needs to transmit lightweight Actor network parameters, thereby improving system communication efficiency and model convergence speed. Furthermore, considering the high latency and poor scalability issues caused by the staleness problem in FL-based synchronous update methods, FCSM utilizes (32) to update network parameters through asynchronous parameter aggregation among agents. This further improves learning efficiency and accelerates model convergence. We utilize asymptotic computational time complexity notation $O(*)$ to analyze the complexity of the different execution steps and sub-steps in Algorithm 1. The computational time complexity of Algorithm 1 is $O\big((C_a + C_a^{\top} + C_c^{\top} + |\boldsymbol{\theta}^{\mu}| + |\boldsymbol{\theta}^{Q}|)|\mathcal{J}|iter + (|\boldsymbol{\theta}^{\mu}| + |\boldsymbol{\theta}^{Q}|)\frac{k|\mathcal{J}|^2 iter}{H}\big)$, where $iter$ is the total number of iterations, $C_a^{\top}$ and $C_c^{\top}$ are the simplified representation of the computational loads of parameter update for the Actor and Critic networks, respectively, and C_a is the simplified representation of the computational load of forward propagation for the Actor networks.

Algorithm 1 Federated Cooperative cost-efficient Service Migration

```
Initialize the network parameters of each agent;
for each iteration do
    Agent obtains observation $< o_j^c(t), o_j^m(t) >$ and selects an action $< a_j^c(t), a_j^m(t) >$ according to the policy network, and obtain the input state $< s^c(t), s^m(t) >$;
    Execute action $< a_j^c(t), a_j^m(t) >$ to get reward $R(t)$ and new state $< s^{c'}(t), s^{m'}(t) >$;
    for each agent do
        Store or update Experience Buffer;
        Sample $B_j$ from Experience Buffer;
        Update parameters of actor evaluation network $\boldsymbol{\theta}_j^{\mu}$ and critic evaluation network $\boldsymbol{\theta}_j^{Q}$ through (26) and (27);
        Update parameters of actor target network $\boldsymbol{\theta}_j^{\mu'}$ and critic target network $\boldsymbol{\theta}_j^{Q'}$ through (28) and (29);
    end for
    Updating the true cost of the network;
    if Round of local iteration $C_{iteration} = H$ then
        Execute intra-cluster parameter sharing according to (31);
    end if
    if Round of intra-cluster parameter sharing $C_{cluster} = G$ then
        Execute global asynchronous parameter aggregation according to (32);
    end if
end for
```

VI. Theoretical analysis

A. Multi-order Spatio-temporal Feature Representation of MSTP

We prove that MSTP can extract multi-order spatio-temporal information in Theorem 1. Moreover, the distributed traffic prediction scheme proposed in this paper only aggregates the traffic of $|\bar{N}|$ regions, which greatly reduces the resource consumption and transmission delay of data collection. To illustrate the crucial role of our multi-order feature integration method, we first give the following definitions.

Definition 1: o-order standard adjacency matrix $\boldsymbol{A}^{(o)}$. It is defined as the multiplication of o standard adjacency matrices $\hat{\boldsymbol{A}}^{o}$, that is, $\boldsymbol{A}^{(o)} = \hat{\boldsymbol{A}}^{o}$. This is a symmetric matrix where the n'th element value $a_{n,n'}^{(o)}$ in the nth row vector $\boldsymbol{a}_n^{(o)}$ of $\boldsymbol{A}^{(o)}$ is proportional to the number of o-hop paths between region n and region n'.

Definition 2: o-order similarity $s_{n,n'}^{(o)}$. The o-order similarity of regions n and n' is defined as the cosine similarity of vectors $\boldsymbol{a}_n^{(o)}$ and $\boldsymbol{a}_{n'}^{(o)}$ in the o-order standard adjacency matrix $\boldsymbol{A}^{(o)}$, denoted as $s_{n,n'}^{(o)} = \frac{\left\langle \boldsymbol{a}_n^{(o)}, \boldsymbol{a}_{n'}^{(o)} \right\rangle}{\|\boldsymbol{a}_n^{(o)}\| \cdot \|\boldsymbol{a}_{n'}^{(o)}\|}$. According to the properties of the Cauchy-Schwartz inequality, the o-order similarity between regions n and n' is proportional to the number of their common o-hop neighbors.

Definition 3: o-order graph regularization. The goal of the o-order graph regularization is to minimize the following term $\sum_{n,n'} \frac{1}{2} \|\boldsymbol{f}_n - \boldsymbol{f}_{n'}\|^2 s_{n,n'}^{(o)}$, where the row vector $\boldsymbol{f}_n$ is the representation of region n and $\|\boldsymbol{f}_n - \boldsymbol{f}_{n'}\|$ denotes the Euclidean distance between region representation $\boldsymbol{f}_n$ and $\boldsymbol{f}_{n'}$. If the o-order similarity $s_{n,n'}^{(o)}$ is high, and the goal of the o-order graph regularization is to minimize $\|\boldsymbol{f}_n - \boldsymbol{f}_{n'}\|$. Therefore, o-order graph regularization can make the representations of regions with high similarity more proximate. Consequently, we have the following lemma 1.

Lemma 1: The operation of multiplying the o-order standard adjacency matrix $\boldsymbol{A}^{(o)}$ on the left of the region representation $\boldsymbol{f}_n$ is equivalent to performing o-order graph regularization, and its effect is to introduce multi-order structural information. The detailed proof of Lemma 1 is given in Appendix A, and based on Lemma 1, Theorem 1 can be obtained.

Theorem 1: The features extracted by MSTP contain the multi-order graph regularization term, which means that the learned representation $\boldsymbol{L}^{(L)}$ contains the multi-order spatio-temporal feature representation of mobile traffic. Specifically, $\boldsymbol{L}^{(L)}$ integrates the information representation of LSTM and GCN, can be expressed as

$$\begin{aligned}\boldsymbol{L}^{(L)} &= \boldsymbol{M}\tilde{\boldsymbol{L}}^{(L-1)} + \boldsymbol{N}\boldsymbol{B} \\ &= \boldsymbol{M}\left(\varpi \boldsymbol{L}^{(L-1)} + (1-\varpi)\boldsymbol{G}^{(L-1)}\right) + \boldsymbol{N}\boldsymbol{B} \\ &= \varpi^{L}\boldsymbol{M}^{L}\bar{\boldsymbol{F}}_{1\sim T} + \sum_{\ell=0}^{L-1} \varpi^{L-1-\ell}(1-\varpi)\boldsymbol{M}^{L-\ell}\hat{\boldsymbol{A}}^{\ell}\bar{\boldsymbol{F}}_{1\sim T} \\ &\quad + \sum_{\ell=0}^{L-1} \varpi^{\ell}\boldsymbol{M}^{\ell}\boldsymbol{N}\boldsymbol{B}, \end{aligned} \tag{33}$$

where $\boldsymbol{M} \in \mathbb{N}^{T\times T}$ denotes the lower triangular matrix where all lower triangular elements are 1, $\boldsymbol{N} \in \mathbb{N}^{T\times T}$ denotes diagonal matrix $\mathrm{diag}(1, \ldots, T)$, and $\boldsymbol{B} = [\boldsymbol{b}, \ldots, \boldsymbol{b}]^{\top}$, respectively. In the second term, $\hat{\boldsymbol{A}}^{\ell}\bar{\boldsymbol{F}}_{1\sim T}$ is equivalent to ℓ-order graph regularization and integrates the data of T time slots, which can obtain ℓ-order spatio-temporal information of $\bar{\boldsymbol{F}}_{1\sim T}$. In this paper, multi-order spatio-temporal information is considered for distributed traffic prediction in MEC networks to improve traffic prediction efficiency. The detailed proof of Theorem 1 is given in Appendix B.

TABLE I
PREDICTION PERFORMANCE COMPARISONS THROUGH TWO EVALUATION METRICS OF THREE DIFFERENT TRAFFIC EVENTS.

Method	Internet		Call		SMS	
	RMSE	MAE	RMSE	MAE	RMSE	MAE
ARIMA	4.983	3.453	0.831	0.587	8.630	6.280
SVR	2.193	1.516	0.416	0.275	4.844	3.509
LSTM	3.120	2.217	0.532	0.351	5.119	3.978
CNN-LSTM	2.052	1.386	0.513	0.272	4.302	2.996
FedDA	2.268	1.633	0.349	0.276	4.485	3.317
MSTP	**1.941**	**1.289**	**0.252**	**0.169**	**3.283**	**2.319**

B. Convergence of FCSM

Based on general assumptions, we prove the convergence of the Actor network in FCSM. These assumptions are as follows: 1) the Actor network has β-smoothness, that is, $J(\boldsymbol{\theta}_j) - J(\boldsymbol{\theta}_j') \leq \langle \nabla J(\boldsymbol{\theta}_j'), \boldsymbol{\theta}_j - \boldsymbol{\theta}_j' \rangle + \frac{\beta}{2}\|\boldsymbol{\theta}_j - \boldsymbol{\theta}_j'\|^2$. 2) SGD has unbiased boundedness, that is, $\mathbb{E}[\|\tilde{\boldsymbol{h}}_j - \boldsymbol{h}_j\|^2] \leq C\|\boldsymbol{h}_j\|^2 + \frac{\sigma_j^2}{B}$. 3) The loss function of the Actor network is convex. 4) The squared loss of the Actor network has an upper bound, that is, $\mathbb{E}\left[\|\nabla J(\boldsymbol{\theta})\|^2\right] \leq \delta^2$. 5) The iteration step size η_k and smooth coefficient β satisfy the relationship $\eta_k < \frac{1}{\beta}$. Next, we prove the convergence of the Actor network in FCSM under assumptions 1)-5) in Theorem 2.

Theorem 2: The Actor network in FCSM has convergence with an upper bound of the time-averaged squared gradient norm, that is,

$$\begin{aligned}&\frac{1}{TGN_k}\sum_{\tau=1}^{T}\sum_{g=1}^{G}\sum_{j=1}^{N_k}\mathbb{E}\left[\left\|\nabla J\left(\boldsymbol{\theta}_{k,j}^{\tau,g}\right)\right\|^2\right] \\ &\leq \frac{2\mathbb{E}\left[J\left(\boldsymbol{\theta}^{0}\right) - J\left(\boldsymbol{\theta}^{T}\right)\right]}{\alpha\eta_{\min}TG} + \frac{\sigma_{\max}^2 H}{B}\frac{1+\zeta^2}{1-\zeta^2} \\ &\quad + 2\alpha\delta^2\Upsilon H\sqrt{N_{\max}}\|\mathbf{W}\| \\ &\quad + \alpha^2\beta\delta^2\Upsilon^2 G H^2 N_{\max}\eta_{\max}\|\mathbf{W}\|^2, \end{aligned} \tag{34}$$

where $\sigma_j \leq \sigma_{\max}$, and the number of BSs in k cluster satisfies $N_k \leq N_{\max}$. The step size of local SGD satisfies $\eta_{\min} \leq \eta_k \leq \eta_{\max}$. $\zeta = \frac{N_k w - 1}{N_k - 1}$ is the second largest eigenvalue of the matrix $\mathbf{W}$, and Υ is the upper bound of the number of rounds that any cluster participates in two consecutive global aggregations. The detailed proof of Theorem 2 is given in Appendix C.

VII. Performance Evaluation

We conduct extensive simulations to evaluate the performance of MSTP and FCSM. Firstly, we evaluate MSTP on real-world traffic datasets and compare its performance with that of both classical and state-of-the-art traffic prediction schemes. This evaluation is based on multiple metrics, including Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and training cost. Then, we evaluate the performance of FCSM in the DT-MEC network through multiple performance metrics, including system total cost, service migration cost, service utility, and resource utilization.

A. Simulation Setting

The datasets used in this paper have been illustrated in detail in Section IV-A. In the simulations, we use the traffic of three events (Internet, call and SMS) from these two datasets [40] to evaluate the performance of MSTP and baseline methods, sampling 1008 time-slot traffic records. For MSTP, a three-layer neural network structure is used, and the prediction sliding window is 4. The model training is performed by using Adam [45] as the optimizer with a learning rate of 1e-3. For the MEC network, the gym module [46] in Python is used to build multi-agents. The network consists of 8 fixedly deployed BSs and connected edge servers, and 100 mobile users move and randomly generate service requests. The bandwidth capacity of the BS is 10 MHz, the storage capacity is 35 GB, and the computing capacity is [5, 10] GHz, where each MEC server is equipped with multiple CPU cores [47]. The service request size is uniformly distributed within [0.5, 1] Mb. Each part of the system cost is normalized, and then the weight factors of system costs are set as $\phi_e = \phi_s = \phi_m = \phi_q = 0.25$. Both the Actor and Critic neural networks consist of 4 layers with the learning rate of 1e-4. The number of iterations is 20,000, and the batch-size is 256.

We compare MSTP with some classical and state-of-the-art mobile traffic prediction methods, which are as follows. **ARIMA** [16], [17]: ARIMA is a classic time series prediction method that combines the autoregressive model and moving average model, and introduces a variance factor to detrend the data. It is a representative method of traffic prediction based on statistical analysis. **Support Vector Regression (SVR)** [48]: SVR is a classic machine learning-based prediction method, and its prediction model has excellent generalization ability. **LSTM** [24]: LSTM is a DL-based method that is widely used for various prediction problems. It can deal with long-term dependencies by improving the internal units of RNN. **CNN-LSTM** [25]: CNN-LSTM is a combined prediction model framework, where CNN is used to capture the spatial dependencies and LSTM is used to capture the temporal dependencies, respectively. It achieves the state-of-the-art performance in mobile traffic prediction. **FedDA** [29]: FedDA is the state-of-the-art distributed traffic prediction method based on FL, and achieves excellent traffic prediction performance.

Furthermore, we compare FCSM with the following methods. **SADDPG**: SADDPG uses a single agent to perform policy exploration and evaluation, updating the network parameters in a centralized manner [10], [49]. **MADDPG**: MADDPG is a popular multi-agent reinforcement learning method, which uses the idea of distributed action exploration and centralized reward [50]. **Cluster Parameter Sharing (CPS)**: CPS uses intra-cluster parameter sharing to update Actor network parameters [44].

B. Prediction Performance

Table I shows the performance of various prediction methods, from which we can observe that our proposed MSTP achieves the best traffic prediction performance compared with baseline methods. Specifically, for RMSE of Internet traffic, our scheme achieves 5.41% gains over CNN-LSTM and 11.49% gains over SVR. Furthermore, compared to baseline methods, MSTP achieves more than 27.79% gains for RMSE of call traffic and more than 22.60% gains for MAE of SMS traffic. Fig. 4 and Fig. 5 show the comparison between the predicted values and the ground truths for three different traffic events under two datasets, and Fig. 6 shows the prediction results of MSTP using traffic generated by the MEC platform. The overall prediction error can be observed from the Cumulative Distribution Function (CDF), and these simulation results demonstrate the excellent prediction performance of MSTP. This is because our MSTP method not only considers the temporal and spatial relationships of mobile traffic, but also achieves a more comprehensive and powerful representation of the coupled spatio-temporal dependencies through integrating multi-order spatio-temporal information.

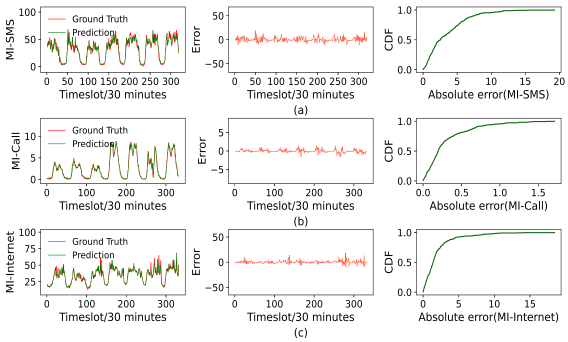


Fig. 4. Prediction results of SMS, call, and Internet traffic in Milan dataset.

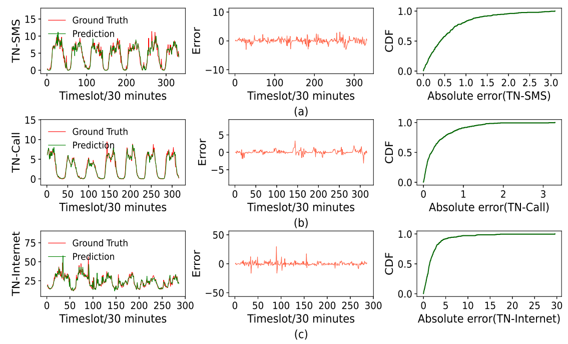


Fig. 5. Prediction results of SMS, call, and Internet traffic in Trentino dataset.

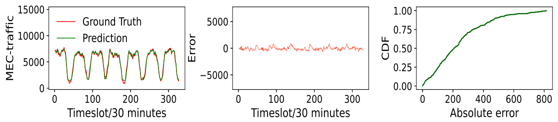


Fig. 6. Traffic prediction results under the MEC network.

C. Clustering and Edge Model Transfer

Furthermore, we analyze the results of region clustering and the performance of edge model transfer. Fig. 7 shows traffic patterns based on real mobile traffic data in some regions for different clusters and the associated training loss, where the regional traffic of the same category shows certain similarities. Fig. 8 shows the comparisons among the predicted traffic by the local model, the transferred edge model, the improved edge model, and the ground-truth traffic in the transferred target regions. Note that the local model is trained with scarce local

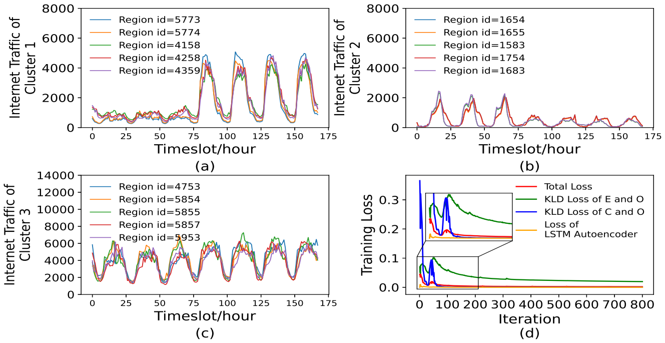


Fig. 7. Traffic patterns for different clusters and clustering training loss.

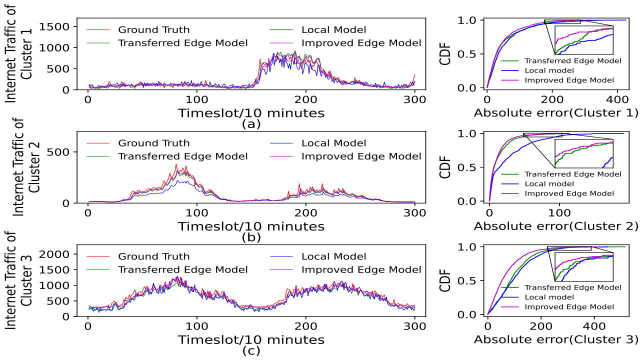


Fig. 8. Prediction results of transferred edge models for different clusters.

data, the transferred edge model is directly transferred from a similar region, and the improved edge model is the model that is transferred from a similar region and then further trained with local data. RCMT achieves excellent traffic prediction performance without the use of global information. Specifically, compared with the local models, the prediction errors of the transferred edge models of three clusters are reduced by 8.49%, 48.20%, and 7.48%, respectively, while the prediction errors of the improved edge models are reduced by 18.33%, 54.38%, and 27.35%, respectively. These results demonstrate that the transferred model can effectively reduce the prediction error for data scarcity regions compared to the local model. This is because our proposed method introduces structured information to regional deep clustering, which can capture the spatio-temporal similarities of regions to support efficient edge model transfer. Therefore, our method overcomes the limitation of insufficient data in some regions and improves the traffic prediction performance for the distributed DT-MEC network through effective region clustering and edge model transfer.

D. Training Cost and Effect of Parameters

We evaluate the training cost of edge models and analyze the effect of parameters on prediction performance, as shown in Fig. 9. When weight parameter ϖ increases from 0.1 to 0.5, the prediction performance of different traffic events is significantly improved in Fig. 9(a), which indicates that capturing the spatio-temporal dependencies of traffic can bring benefits to traffic prediction. When ϖ increases to 0.7, the prediction performance shows a downtrend, indicating that the introduction of too much spatial correlation can interfere with the acquisition of traffic features and thus affect the traffic prediction performance. We also evaluate the effect of the number of similar regions λ on the prediction performance, as shown in Fig. 9(b). We can observe that the optimal λ is different for different traffic events and geographic settings. Overall, with the increase of λ, the RMSE of 6 traffic events show a downward trend to varying degrees. However, with the excessive increase of λ, e.g., for TN-Internet traffic, when λ is greater than 5, the RMSE increases significantly. Simulation results show that with a carefully chosen λ to capture the correlations among similar regions can account for spatial dependencies and indeed improve the prediction performance. The training loss of MSTP is shown in Fig. 9(c), and the average training time of our scheme is 42s, as shown in Fig. 9(d). We can observe that our proposed MSTP can achieve excellent prediction performance with low model training cost, which is suitable for MEC servers with limited capacity.

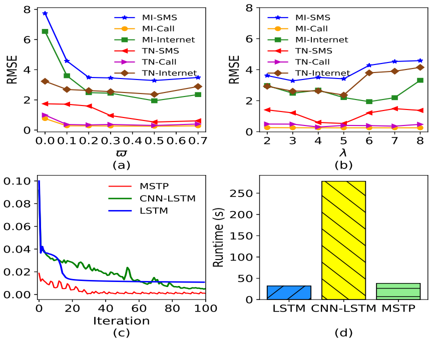


Fig. 9. Weight parameter ϖ, number of similarity regions λ, number of iterations, and training time.

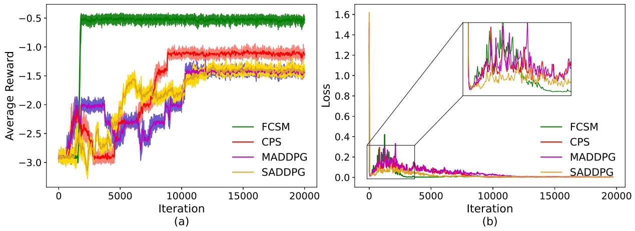


Fig. 10. Actor network training reward and training loss.

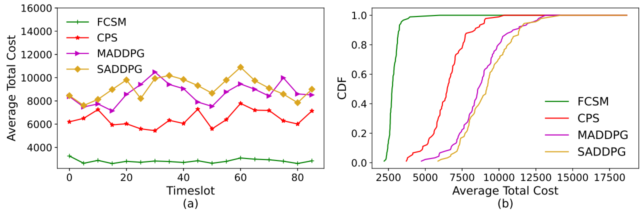


Fig. 11. Total cost and CDF of total cost.

E. Actor Network Convergence and Training Loss

The convergence and loss of the Actor network can reflect the learning efficiency and stability of network model.

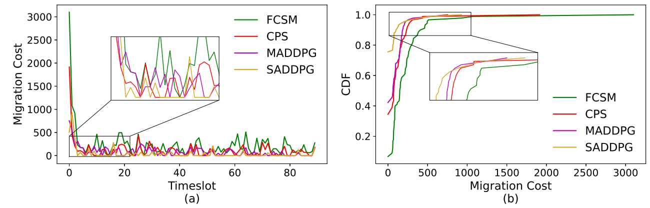


Fig. 12. Service migration cost and CDF of service migration cost.

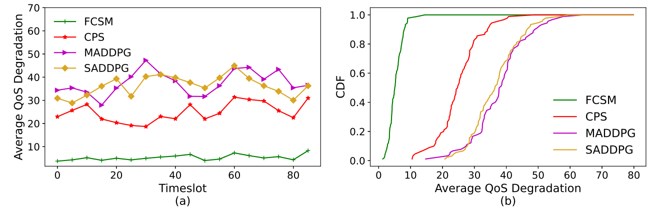


Fig. 13. Service QoS degradation cost and CDF of service QoS degradation cost.

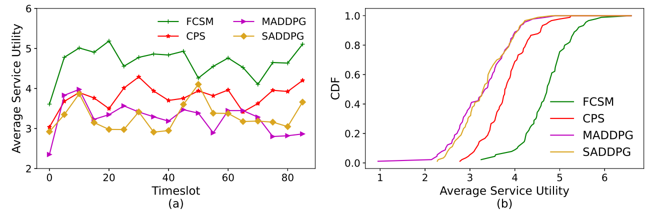


Fig. 14. Service utility and CDF of service utility.

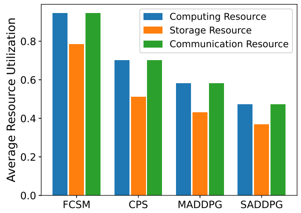


Fig. 15. Computing, storage and communication resource utilization of different algorithms.

Fig. 10 shows the training reward and loss of the Actor network. Although the stochastic property of the MEC network causes certain reward and loss fluctuations, FCSM achieves the best convergence at about 2200 iterations, which significantly outperforms the other comparison algorithms. This is because FCSM employs a distributed learning scheme that only requires a lightweight network architecture, and achieves efficient model training through intra-cluster parameter sharing and global asynchronous parameter aggregation among agents. FCSM does not require the sharing of input state information among agents, and only lightweight Actor network parameters need to be transferred. This improves system communication efficiency and model convergence speed. In addition, the convergence of FCSM is proved in Theorem 2, which provides a theoretical guarantee for the convergence of FCSM. We can observe that SADDPG has the worst convergence with around 12000 iterations. This is because the centralized exploration with single agent requires a neural network model with complex architecture and a large number of parameters to process global state information.

F. System Total Cost and Service Migration Cost

System total cost and service migration cost are key performance metrics for network operators to deploy and operate MEC networks. The total system cost and service migration cost are shown in Fig. 11 and Fig. 12. We can observe that the total cost of FCSM is significantly lower than that of other comparison methods with an average cost of 2855.05. This is because FCSM adopts a federated DRL-based parameter update method. Through intra-cluster parameter sharing and global asynchronous parameter aggregation, it can learn global information and optimize the long-term total system cost. Specifically, FCSM puts base stations into sleep mode during low-traffic periods to reduce network energy cost. Meanwhile, FCSM meets local service requests by proper service migration and improves QoS through appropriate resource allocation. We can also observe that our average migration cost is 227.78, which is higher than that of the comparison algorithms. This is because FCSM migrates services to meet local service requests and causes higher migration cost.

G. QoS Degradation Cost, Service Utility and Resource Utilization

QoS degradation cost and service utility are important metrics, which represents the ability of the MEC network to admit mobile user requests. As shown in Fig. 13 and Fig. 14, FCSM can reduce QoS degradation cost and improve service utility of the MEC network. Specifically, the QoS degradation cost of FCSM is 5.37, while that of MADDPG is as high as 38. This is because FCSM reasonably migrates services and greatly satisfies the service requests of mobile users. Furthermore, the efficient allocation of network resources can reduce communication and computing delay, thereby reducing QoS degradation cost. Overall, FCSM significantly improves the QoS of users while responding to service requests. Furthermore, we evaluate the computing, storage and communication resource utilization of different methods, which reflects whether the algorithm fully make use of the network resources of edge servers. As shown in Fig. 15, the average computing, storage and communication resource utilization of FCSM are 94.45%, 78.30% and 94.44%, respectively. FCSM improves the resource utilization of active infrastructure by switching base stations according to the changes in traffic load. Moreover, FCSM properly migrates services, significantly satisfying the service requests of mobile users, and improves resource utilization through efficient resource allocation.

VIII. Conclusion

In this paper, we have proposed a traffic prediction-assisted federated DRL framework for service migration in the DT-MEC network. Firstly, a distributed traffic prediction scheme for large-scale MEC networks has been proposed, which achieves high-efficiency and low-cost traffic prediction through multi-order feature integration of spatio-temporal information and edge model transfer. Then, we have formulated the cost-efficient service migration problem as a multi-agent Markov model, and have proposed a federated collaboration-based cost-efficient service migration scheme. Furthermore, through theoretical analysis, we have proved the convergence of FCSM. Finally, we have evaluated the performance of MSTP and FCSM. Simulation results show that MSTP yields superior traffic prediction performance relative to prior approaches, and FCSM reduces the long-term cost of MEC networks and shows excellent convergence. Since the learning efficiency of a multi-agent system is greatly affected by the stochastic properties of networks, our future work will focus on the optimization of the multi-agent learning process to improve robustness.

Appendix A
Proof of Lemma 1

Proof: According to the definition of $\boldsymbol{A}^{(o)}$, the element $a_{n,n'}^{(o)}$ in the nth row and n'th column is

$$a_{n,n'}^{(o)} = \sum_{k^{(1)},\dots,k^{(o-1)}=1}^{|\bar{N}|} \frac{1}{\sqrt{d_n}\sqrt{d_{n'}}} \frac{a_{nk^{(1)}}\left(\Pi_{q=1}^{o-2} a_{k^{(q)}k^{(q+1)}}\right) a_{k^{(o-1)}n'}}{d_{k^{(1)}}\cdots d_{k^{(o-1)}}} = \frac{1}{\sqrt{d_n}\sqrt{d_{n'}}}\hat{a}_{n,n'}^{(o)}. \quad (35)$$

Hence, let $\boldsymbol{F} = [\boldsymbol{f}_1^\top, \dots, \boldsymbol{f}_{|\bar{N}|}^\top]^\top$, and then we have

$$\hat{\boldsymbol{f}}_n = \boldsymbol{a}_n^{(o)}\boldsymbol{F} = \sum_{k=1}^{|\bar{N}|} \frac{1}{\sqrt{d_n}\sqrt{d_k}}\hat{a}_{n,k}^{(o)}\boldsymbol{f}_k = \sum_{k=1}^{|\bar{N}|} a_{n,k}^{(o)}\boldsymbol{f}_k. \quad (36)$$

Furthermore, the representation of regions n and n' after left multiplying by $\boldsymbol{A}^{(o)}$ is $\hat{\boldsymbol{f}}_n$ and $\hat{\boldsymbol{f}}_{n'}$. The distance between $\hat{\boldsymbol{f}}_n$ and $\hat{\boldsymbol{f}}_{n'}$ is

$$\left\|\hat{\boldsymbol{f}}_n - \hat{\boldsymbol{f}}_{n'}\right\| = \left\|\sum_{k=1}^{|\bar{N}|} a_{n,k}^{(o)}\boldsymbol{f}_k - \sum_{k=1}^{|\bar{N}|} a_{n',k}^{(o)}\boldsymbol{f}_k\right\| \leq \sum_{k=1}^{|\bar{N}|}\left\|a_{n,k}^{(o)} - a_{n',k}^{(o)}\right\| \|\boldsymbol{f}_k\|. \quad (37)$$

From (37), we can see that when the values of $a_{n,k}^{(o)}$ and $a_{n',k}^{(o)}$ are close, the magnitude of $\left\|\hat{\boldsymbol{f}}_n - \hat{\boldsymbol{f}}_{n'}\right\|$ can be small, which is consistent with the idea of o-order graph regularization. Since graph regularization can introduce structural information into the model, the operation of $\boldsymbol{A}^{(o)}\boldsymbol{f}_n$ can introduce o-order structural information.

Appendix B
Proof of Theorem 1

Proof: The update of the ℓth layer LSTM at the t time step is expressed as

$$\boldsymbol{L}^{(\ell)}(t) = \sigma\left(\tilde{\boldsymbol{L}}^{(\ell-1)}(t)\boldsymbol{W}_{r1}^{(\ell-1)} + \boldsymbol{L}^{(\ell)}(t-1)\boldsymbol{W}_{r2}^{(\ell-1)} + \boldsymbol{b}^{(\ell-1)}\right). \quad (38)$$

To simplify the proof, we assume that the activation function $\sigma(x) = x$, the parameter $\boldsymbol{W}_{r1}^{(\ell-1)} = \boldsymbol{W}_{r2}^{(\ell-1)} = \boldsymbol{I}, \boldsymbol{b}^{(\ell-1)} = \boldsymbol{b}, \forall \ell \in \{1, 2, \cdots, L\}$, and let the initial hidden state of LSTM $\boldsymbol{L}^{(\ell)}(0) = \boldsymbol{0}$, then (38) can be simplified as

$$\begin{aligned}\boldsymbol{L}^{(\ell)}(t) &= \tilde{\boldsymbol{L}}^{(\ell-1)}(t) + \boldsymbol{L}^{(\ell)}(t-1) + \boldsymbol{b} \\ &= \sum_{\tau=1}^{t}\tilde{\boldsymbol{L}}^{(\ell-1)}(\tau) + t\boldsymbol{b}. \end{aligned} \quad (39)$$

$\boldsymbol{L}^{(\ell)}$ contains the output of T time series $\boldsymbol{L}^{(\ell)}(1), \dots, \boldsymbol{L}^{(\ell)}(T)$, then we have $\boldsymbol{L}^{(\ell)} = \boldsymbol{M}\tilde{\boldsymbol{L}}^{(\ell-1)} + \boldsymbol{N}\boldsymbol{B}$.

On the other hand, for GCN, we also simplify $\sigma(x) = x$ and $\boldsymbol{W}_g^{(\ell-1)} = \boldsymbol{I}, \forall \ell \in \{1, 2, \cdots, L\}$, and $\tilde{\boldsymbol{D}}^{-\frac{1}{2}}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{-\frac{1}{2}}$ is abbreviated as $\hat{\boldsymbol{A}}$. Hence, the graph convolution operation of the ℓth layer is $\boldsymbol{G}^{(\ell)}(t) = \hat{\boldsymbol{A}}\boldsymbol{G}^{(\ell-1)}(t) = \hat{\boldsymbol{A}}^{\ell}\boldsymbol{G}^{(0)}(t) = \hat{\boldsymbol{A}}^{\ell}\bar{\boldsymbol{F}}(t)$. After integrating the information representation of LSTM and GCN, we have (33) and Theorem 1 is proved.

Appendix C
Proof of Theorem 2

Proof: The expectation of the difference between the global losses for two consecutive epochs is

$$\begin{aligned}&\mathbb{E}\left[J\left(\boldsymbol{\theta}^{\tau}\right) - J\left(\boldsymbol{\theta}^{\tau-1}\right)\right] \\ &\leq \mathbb{E}\left[(1-\alpha)J\left(\boldsymbol{\theta}^{\tau-1}\right) + \alpha J(\boldsymbol{\theta}_k^{\tau'}) - J\left(\boldsymbol{\theta}^{\tau-1}\right)\right] \\ &= \alpha\mathbb{E}\left[J(\boldsymbol{\theta}_k^{\tau'}) - J(\boldsymbol{\theta}^{\tau'})\right] + \alpha\mathbb{E}\left[J(\boldsymbol{\theta}^{\tau'}) - J(\boldsymbol{\theta}^{\tau-1})\right], \end{aligned} \quad (40)$$

where $\boldsymbol{\theta}_k^{\tau'}$ is the result of G rounds of intra-cluster parameter update from the model parameters $\boldsymbol{\theta}^{\tau'}$ of the previous epoch in cluster k. For the first term in (40), we have

$$\mathbb{E}\left[J(\boldsymbol{\theta}_k^{\tau'}) - J(\boldsymbol{\theta}^{\tau'})\right] \leq \frac{1}{N_k}\sum_{j=1}^{N_k}\mathbb{E}\left[J\left(\boldsymbol{\theta}_{k,j}^{\tau',G}\right) - J\left(\boldsymbol{\theta}_{k,j}^{\tau',0}\right)\right]. \quad (41)$$

According to conclusion in [44] and $\eta_k\beta < 1$, we have

$$\begin{aligned}&\mathbb{E}\left[J(\boldsymbol{\theta}_k^{\tau'}) - J\left(\boldsymbol{\theta}^{\tau'}\right)\right] \\ &\leq -\frac{\eta_k}{2N_k}\sum_{g=1}^{G}\sum_{j=1}^{N_k}\mathbb{E}\left[\left\|\nabla J\left(\boldsymbol{\theta}_{k,j}^{\tau',g}\right)\right\|^2\right] + \frac{\eta_k\sigma_{\max}^2 GH}{2B}\frac{1+\zeta^2}{1-\zeta^2}. \end{aligned} \quad (42)$$

For the second term in (40), since the β-smoothness of the loss function, we have

$$\begin{aligned}&\mathbb{E}\left[J(\boldsymbol{\theta}^{\tau'}) - J(\boldsymbol{\theta}^{\tau-1})\right] \\ &\leq \mathbb{E}\left[\left\|\nabla J(\boldsymbol{\theta}^{\tau-1})\right\|\|\boldsymbol{\theta}^{\tau'} - \boldsymbol{\theta}^{\tau-1}\|\right] + \frac{\beta}{2}\mathbb{E}\left[\|\boldsymbol{\theta}^{\tau'} - \boldsymbol{\theta}^{\tau-1}\|^2\right]. \end{aligned} \quad (43)$$

For $\|\boldsymbol{\theta}^{\tau'} - \boldsymbol{\theta}^{\tau-1}\|$ in (43), according to federated global parameters aggregation in (32), we have

$$\begin{aligned}&\|\boldsymbol{\theta}^{\tau'} - \boldsymbol{\theta}^{\tau-1}\| \\ &\leq \|\boldsymbol{\theta}^{\tau'} - \boldsymbol{\theta}^{\tau'+1}\| + \|\boldsymbol{\theta}^{\tau'+1} - \boldsymbol{\theta}^{\tau'+2}\| + \cdots + \left\|\boldsymbol{\theta}^{\tau-2} - \boldsymbol{\theta}^{\tau-1}\right\| \\ &= \alpha\left\|\boldsymbol{\theta}_k^{(\tau'+1)'} - \boldsymbol{\theta}^{\tau'}\right\| + \cdots + \alpha\left\|\boldsymbol{\theta}_k^{(\tau-1)'} - \boldsymbol{\theta}^{\tau-2}\right\|, \end{aligned} \quad (44)$$

where $\boldsymbol{\theta}_k^{(\tau'+1)'}$ can be obtained from $\boldsymbol{\theta}^{(\tau'+1)'}$ through intra-cluster update for G rounds. For the first term in (44), simplify the notation $(\tau'+1)'$ as τ_k, and assume that the gap between $\boldsymbol{\theta}_k^{\tau_k}$ and the global parameter $\boldsymbol{\theta}^{\tau'}$ is smaller than the update gap in cluster k, then we have

$$\begin{aligned}\|\boldsymbol{\theta}_k^{\tau_k} - \boldsymbol{\theta}^{\tau'}\| &\le \|\boldsymbol{\theta}_k^{\tau_k} - \boldsymbol{\theta}^{\tau_k}\| \\ &= \left\| \frac{1}{N_k}\sum_{j=1}^{N_k} \boldsymbol{\theta}_{k,j}^{\tau_k,G} - \frac{1}{N_k}\sum_{j=1}^{N_k} \boldsymbol{\theta}_{k,j}^{\tau_k,0} \right\| \\ &\le \frac{1}{N_k}\sum_{j=1}^{N_k}\left[\left\|(\boldsymbol{\theta}_{k,j}^{\tau_k,G} - \boldsymbol{\theta}_{k,j}^{\tau_k,G-1})\right\| + \dots \right. \\ &\qquad \left. + \left\|(\boldsymbol{\theta}_{k,j}^{\tau_k,1} - \boldsymbol{\theta}_{k,j}^{\tau_k,0})\right\|\right]. \end{aligned} \tag{45}$$

Without loss of generality, we analyze the difference between the $(g+1)$th and gth intra-cluster parameter update in (45),

$$\begin{aligned} &\left\|\boldsymbol{\theta}_{k,j}^{\tau_k,g+1} - \boldsymbol{\theta}_{k,j}^{\tau_k,g}\right\| \\ &\le \left\|(\boldsymbol{\Theta}_k^{\tau_k,g,H} - \boldsymbol{\Theta}_k^{\tau_k,g,H-1})\mathbf{W}\mathbf{I}_j\right\| + \dots \\ &+ \left\|(\boldsymbol{\Theta}_k^{\tau_k,g,2} - \boldsymbol{\Theta}_k^{\tau_k,g,1})\mathbf{W}\mathbf{I}_j\right\| + \left\|(\boldsymbol{\Theta}_k^{\tau_k,g,1}\mathbf{W} - \boldsymbol{\Theta}_k^{\tau_k,g,0})\mathbf{I}_j\right\| \\ &\le \sqrt{N_k}\eta_k\delta\left\|\mathbf{W}\right\| + \dots + \sqrt{N_k}\eta_k\delta\left\|\mathbf{W}\right\| \\ &\le H\sqrt{N_k}\eta_k\delta\left\|\mathbf{W}\right\|, \end{aligned} \tag{46}$$

where $\mathbf{I}_j \in \mathbb{N}^{N_k\times 1}$ is a 0-1 column vector, and only the jth element is 1. Since a single local iteration has a minor impact on the parameters, we simplify the expression for the last term of the first inequality in (46). Substituting the result from (46) back into (45), we have

$$\|\boldsymbol{\theta}_k^{\tau_k} - \boldsymbol{\theta}^{\tau'}\| \le GH\sqrt{N_k}\eta_k\delta\left\|\mathbf{W}\right\|. \tag{47}$$

Substituting the result from (47) successively back into (44) and (43), we have

$$\begin{aligned} &\mathbb{E}\left[J(\boldsymbol{\theta}^{\tau'}) - J(\boldsymbol{\theta}^{\tau-1})\right] \\ &\le \Upsilon\alpha GH\sqrt{N_k}\eta_k\delta^2\left\|\mathbf{W}\right\| + \frac{\beta}{2}\Upsilon^2\alpha^2G^2H^2N_k\eta_k^2\delta^2\left\|\mathbf{W}\right\|^2. \end{aligned} \tag{48}$$

According to (40), (42), and (48), we have

$$\begin{aligned} &\mathbb{E}\left[J\left(\boldsymbol{\theta}^{\tau}\right) - J\left(\boldsymbol{\theta}^{\tau-1}\right)\right] \le \\ &-\alpha\frac{\eta_k}{2N_k}\sum_{g=1}^{G}\sum_{j=1}^{N_k}\mathbb{E}\left[\left\|\nabla J\left(\boldsymbol{\theta}_{k,j}^{\tau',g}\right)\right\|^2\right] + \alpha\frac{\eta_k\sigma_{\max}^2GH}{2B}\frac{1+\zeta^2}{1-\zeta^2} \\ &+ \Upsilon\alpha^2GH\sqrt{N_k}\eta_k\delta^2\left\|\mathbf{W}\right\| + \frac{\beta}{2}\Upsilon^2\alpha^3G^2H^2N_k\eta_k^2\delta^2\left\|\mathbf{W}\right\|^2. \end{aligned} \tag{49}$$

For (49), multiplying both sides by $\frac{2}{\alpha\eta_kG}$, then we rearrange the terms to obtain

$$\begin{aligned} &\frac{1}{GN_k}\sum_{g=1}^{G}\sum_{j=1}^{N_k}\mathbb{E}\left[\left\|\nabla J\left(\boldsymbol{\theta}_{k,j}^{\tau',g}\right)\right\|^2\right] \\ &\le \frac{2\mathbb{E}\left[J\left(\boldsymbol{\theta}^{\tau-1}\right) - J\left(\boldsymbol{\theta}^{\tau}\right)\right]}{\alpha\eta_kG} + \frac{\sigma_{\max}^2H}{B}\frac{1+\zeta^2}{1-\zeta^2} \\ &\quad + 2\Upsilon\alpha H\sqrt{N_k}\delta^2\left\|\mathbf{W}\right\| + \beta\Upsilon^2\alpha^2GH^2N_k\eta_k\delta^2\left\|\mathbf{W}\right\|^2. \end{aligned} \tag{50}$$

Therefore, we obtain the expectation of squared gradient norm in (50) after local iteration and intra-cluster update in each epoch. Furthermore, for all $\tau \in \{1,\dots,T\}$, an upper bound of the time-averaged squared gradient norm can be obtained in (34) with global asynchronous parameter aggregation.

References

[1] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1657–1681, 2017.

[2] U. Cisco, "Cisco annual internet report (2018–2023) white paper," *Cisco: San Jose, CA, USA*, 2020.

[3] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing: A key technology towards 5G," https://bit.ly/3b9KBfc, pp. 1–16, Sept 2015, ETSI.

[4] X. Chen, Y. Bi, X. Chen, H. Zhao, N. Cheng, F. Li, and W. Cheng, "Dynamic service migration and request routing for microservice in multicell mobile-edge computing," *IEEE Internet Things J.*, vol. 9, no. 15, pp. 13 126–13 143, 2022.

[5] T. Ouyang, Z. Zhou, and X. Chen, "Follow me at the edge: Mobility-aware dynamic service placement for mobile edge computing," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 10, pp. 2333–2345, 2018.

[6] X. Ge, S. Tu, G. Mao, C. Wang, and T. Han, "5G ultra-dense cellular networks," *IEEE Wireless Commun.*, vol. 23, no. 1, pp. 72–79, 2016.

[7] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 869–904, 2020.

[8] D. C. Nguyen, P. Cheng, M. Ding, D. López-Pérez, P. N. Pathirana, J. Li, A. Seneviratne, Y. Li, and H. V. Poor, "Enabling AI in future wireless networks: A data life cycle perspective," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 1, pp. 553–595, 2021.

[9] X. Chen, X. Wang, B. Yi, Q. He, and M. Huang, "Deep learning-based traffic prediction for energy efficiency optimization in software-defined networking," *IEEE Syst. J.*, vol. 15, no. 4, pp. 5583–5594, 2021.

[10] Q. Wu, X. Chen, Z. Zhou, L. Chen, and J. Zhang, "Deep reinforcement learning with spatio-temporal traffic forecasting for data-driven base station sleep control," *IEEE/ACM Trans. Netw.*, vol. 29, no. 2, pp. 935–948, 2021.

[11] R. Dong, C. She, W. Hardjawana, Y. Li, and B. Vucetic, "Deep learning for hybrid 5G services in mobile edge computing systems: Learn from a digital twin," *IEEE Trans. Wireless Commun.*, vol. 18, no. 10, pp. 4692–4707, 2019.

[12] L. Zhao, Z. Bi, A. Hawbani, K. Yu, Y. Zhang, and M. Guizani, "ELITE: An intelligent digital twin-based hierarchical routing scheme for softwarized vehicular networks," *IEEE Trans. Mobile Comput.*, Early Access, May 31, 2022, doi: 10.1109/TMC.2022.3179254.

[13] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Low-latency federated learning and blockchain for edge association in digital twin empowered 6G networks," *IEEE Trans. Ind. Informatics*, vol. 17, no. 7, pp. 5098–5107, 2021.

[14] A. Furno, M. Fiore, R. Stanica, C. Ziemlicki, and Z. Smoreda, "A tale of ten cities: Characterizing signatures of mobile traffic in urban areas," *IEEE Trans. Mobile Comput.*, vol. 16, no. 10, pp. 2682–2696, 2017.

[15] D. Naboulsi, M. Fiore, S. Ribot, and R. Stanica, "Large-scale mobile traffic analysis: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 124–161, 2016.

[16] F. Xu, Y. Lin, J. Huang, D. Wu, H. Shi, J. Song, and Y. Li, "Big data driven mobile traffic understanding and forecasting: A time series approach," *IEEE Trans. Services Comput.*, vol. 9, no. 5, pp. 796–805, 2016.

[17] H. Kim, J. Lee, Y. Choi, Y. Chung, and H. Lee, "Dynamic bandwidth provisioning using ARIMA-based traffic forecasting for mobile wimax," *Comput. Commun.*, vol. 34, no. 1, pp. 99–106, 2011.

[18] R. Li, Z. Zhao, X. Zhou, J. Palicot, and H. Zhang, "The prediction analysis of cellular radio access network traffic: From entropy theory to networking practice," *IEEE Commun. Mag.*, vol. 52, no. 6, pp. 234–240, 2014.

[19] D. J. A. Clemente, D. F. S. Fernandes, R. Cortesão, G. E. Soares, P. Sebastião, and L. S. Ferreira, "Assessment of traffic prediction models for mobile communication networks," in *Proc. 22nd Int. Symp. Wireless Pers. Multimedia Commun. (WPMC)*, 2019, pp. 1–4.

[20] Y. Xu, F. Yin, W. Xu, J. Lin, and S. Cui, “Wireless traffic prediction with scalable gaussian process: Framework, algorithms, and verification,” *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1291–1306, 2019.

[21] X. Wang, Z. Zhou, F. Xiao, K. Xing, Z. Yang, Y. Liu, and C. Peng, “Spatio-temporal analysis and prediction of cellular traffic in metropolis,” *IEEE Trans. Mobile Comput.*, vol. 18, no. 9, pp. 2190–2202, 2019.

[22] C. Zhang, H. Zhang, J. Qiao, D. Yuan, and M. Zhang, “Deep transfer learning for intelligent cellular traffic prediction based on cross-domain big data,” *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1389–1401, 2019.

[23] K. He, X. Chen, Q. Wu, S. Yu, and Z. Zhou, “Graph attention spatial-temporal network with collaborative global-local learning for citywide mobile traffic prediction,” *IEEE Trans. Mobile Comput.*, vol. 21, no. 4, pp. 1244–1256, 2022.

[24] C. Qiu, Y. Zhang, Z. Feng, P. Zhang, and S. Cui, “Spatio-temporal wireless traffic prediction with recurrent neural network,” *IEEE Wireless Commun. Lett.*, vol. 7, no. 4, pp. 554–557, 2018.

[25] C. Zhang, H. Zhang, D. Yuan, and M. Zhang, “Citywide cellular traffic prediction based on densely connected convolutional neural networks,” *IEEE Commun. Lett.*, vol. 22, no. 8, pp. 1656–1659, 2018.

[26] J. Wang, J. Tang, Z. Xu, Y. Wang, G. Xue, X. Zhang, and D. Yang, “Spatiotemporal modeling and prediction in cellular networks: A big data enabled deep learning approach,” in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, 2017, pp. 1–9.

[27] C. Zhang and P. Patras, “Long-term mobile traffic forecasting using deep spatio-temporal neural networks,” in *Proc. 19th ACM Int. Symp. Mobile Ad Hoc Netw. Comput. (MobiHoc)*, 2018, pp. 231–240.

[28] N. Zhao, A. Wu, Y. Pei, Y. Liang, and D. Niyato, “Spatial-temporal aggregation graph convolution network for efficient mobile cellular traffic prediction,” *IEEE Commun. Lett.*, vol. 26, no. 3, pp. 587–591, 2022.

[29] C. Zhang, S. Dang, B. Shihada, and M. Alouini, “Dual attention-based federated learning for wireless traffic prediction,” in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, 2021, pp. 1–10.

[30] T. Taleb, A. Ksentini, and P. A. Frangoudis, “Follow-me cloud: When cloud services follow mobile users,” *IEEE Trans. Cloud Comput.*, vol. 7, no. 2, pp. 369–382, 2019.

[31] S. Yu, X. Chen, Z. Zhou, X. Gong, and D. Wu, “When deep reinforcement learning meets federated learning: Intelligent multitimescale resource management for multiaccess edge computing in 5G ultradense network,” *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2238–2251, 2021.

[32] M. R. Anwar, S. Wang, M. F. Akram, S. Raza, and S. Mahmood, “5G-enabled MEC: A distributed traffic steering for seamless service migration of internet of vehicles,” *IEEE Internet Things J.*, vol. 9, no. 1, pp. 648–661, 2022.

[33] Z. Liang, Y. Liu, T. Lok, and K. Huang, “Multi-cell mobile edge computing: Joint service migration and resource allocation,” *IEEE Trans. Wirel. Commun.*, vol. 20, no. 9, pp. 5898–5912, 2021.

[34] R. A. Addad, D. L. C. Dutra, M. Bagaa, T. Taleb, and H. Flinck, “Fast service migration in 5G trends and scenarios,” *IEEE Netw.*, vol. 34, no. 2, pp. 92–98, 2020.

[35] A. Mukhopadhyay, G. Iosifidis, and M. Ruffini, “Migration-aware network services with edge computing,” *IEEE Trans. Netw. Serv. Manag.*, vol. 19, no. 2, pp. 1458–1471, 2022.

[36] X. Li, S. Chen, Y. Zhou, J. Chen, and G. Feng, “Intelligent service migration based on hidden state inference for mobile edge computing,” *IEEE Trans. Cogn. Commun. Netw.*, vol. 8, no. 1, pp. 380–393, 2022.

[37] F. Tang, C. Liu, K. Li, Z. Tang, and K. Li, “Task migration optimization for guaranteeing delay deadline with mobility consideration in mobile edge computing,” *J. Syst. Archit.*, vol. 112, p. 101849, 2021.

[38] M. Deruyck, W. Joseph, and L. Martens, “Power consumption model for macrocell and microcell base stations,” *Trans. Emerging Telecommun. Technol.*, vol. 25, no. 3, pp. 320–333, 2014.

[39] A. Shahini, A. Kiani, and N. Ansari, “Energy efficient resource allocation in eh-enabled CR networks for IoT,” *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3186–3193, 2019.

[40] G. Barlacchi, M. De Nadai, R. Larcher, A. Casella, C. Chitic, G. Torrisi, F. Antonelli, A. Vespignani, A. Pentland, and B. Lepri, “A multi-source dataset of urban life in the city of milan and the province of trentino,” *Scientific data*, vol. 2, no. 1, pp. 1–15, 2015.

[41] M. Müller, “Dynamic time warping,” *Information retrieval for music and motion*, pp. 69–84, 2007.

[42] J. Xie, R. B. Girshick, and A. Farhadi, “Unsupervised deep embedding for clustering analysis,” in *Proc. 33nd Int. Conf. Mach. Learn. (ICML)*, 2016, pp. 478–487.

[43] Y. Ren, K. Hu, X. Dai, L. Pan, S. C. H. Hoi, and Z. Xu, “Semi-supervised deep embedded clustering,” *Neurocomputing*, vol. 325, pp. 121–130, 2019.

[44] Z. Zhu, S. Wan, P. Fan, and K. B. Letaief, “Federated multiagent Actor-Critic learning for age sensitive mobile-edge computing,” *IEEE Internet Things J.*, vol. 9, no. 2, pp. 1053–1067, 2021.

[45] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proc. 3rd Int. Conf. Learn. Representations (ICLR)*, 2015, pp. 1–15.

[46] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, “OpenAI gym,” *arXiv preprint arXiv:1606.01540*, 2016.

[47] Y. Sun, S. Zhou, and J. Xu, “EMM: energy-aware mobility management for mobile edge computing in ultra dense networks,” *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2637–2646, 2017.

[48] H. Feng, Y. Shu, S. Wang, and M. Ma, “SVM-based models for predicting WLAN traffic,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2006, pp. 597–602.

[49] J. Ye and Y. A. Zhang, “DRAG: deep reinforcement learning based base station activation in heterogeneous networks,” *IEEE Trans. Mobile Comput.*, vol. 19, no. 9, pp. 2076–2087, 2020.

[50] H. Peng and X. Shen, “Multi-agent reinforcement learning based resource management in MEC- and UAV-assisted vehicular networks,” *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 131–141, 2021.

Xiangyi Chen received the M.S. degree in computer science from Northeastern University, Shenyang, China, in 2019. She is currently pursuing the Ph.D. degree in computer science at Northeastern University, Shenyang, China. She is a visiting Ph.D. student with the Institute for Computing Systems Architecture (ICSA) School of Informatics, University of Edinburgh, Edinburgh, UK. Her research interests include edge computing, edge-AI and reinforcement learning, etc.

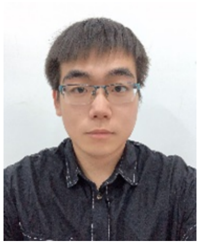
Guangjie Han (Fellow, IEEE) is currently a Professor with the Department of Internet of Things Engineering, Hohai University, Changzhou, China. He received his Ph.D. degree from Northeastern University, Shenyang, China, in 2004. In February 2008, he finished his work as a Postdoctoral Researcher with the Department of Computer Science, Chonnam National University, Gwangju, Korea. From October 2010 to October 2011, he was a Visiting Research Scholar with Osaka University, Suita, Japan. From January 2017 to February 2017, he was a Visiting Professor with City University of Hong Kong, China. From July 2017 to July 2020, he was a Distinguished Professor with Dalian University of Technology, China. His current research interests include Internet of Things, Industrial Internet, Machine Learning and Artificial Intelligence, Mobile Computing, Security and Privacy. Dr. Han has over 500 peer-reviewed journal and conference papers, in addition to 160 granted and pending patents. Currently, his H-index is 63 and i10-index is 267 in Google Citation (Google Scholar). The total citation count of his papers raises above 14600+ times.

Dr. Han is a Fellow of the UK Institution of Engineering and Technology (FIET). He has served on the Editorial Boards of up to 10 international journals, including the IEEE TII, IEEE TCCN, IEEE Systems, etc. He has guest-edited several special issues in IEEE Journals and Magazines, including the IEEE JSAC, IEEE Communications, IEEE Wireless Communications, Computer Networks, etc. Dr. Han has also served as chair of organizing and technical committees in many international conferences. He has been awarded 2020 IEEE Systems Journal Annual Best Paper Award and the 2017-2019 IEEE ACCESS Outstanding Associate Editor Award. He is a Fellow of IEEE.

Yuanguo Bi (M'11) received the Ph.D. degree in computer science from Northeastern University, Shenyang, China, in 2010. He was a Visiting Ph.D. Student with the BroadBand Communications Research (BBCR) lab, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada from 2007 to 2009. He is currently a Professor with the School of Computer Science and Engineering, Northeastern University. He has authored/coauthored more than 50 journal/conference papers, including high quality journal papers, such as IEEE JSAC, IEEE TWC, IEEE TITS, IEEE TVT, IEEE IoT Journal, IEEE Communications Magazine, IEEE Wireless Communications, IEEE Network, and mainstream conferences, such as IEEE Global Communications Conference, IEEE International Conference on Communications. His research interests include medium access control, QoS routing, multihop broadcast, and mobility management in vehicular networks, software-defined networking, and mobile edge computing. Dr. Bi has served as an Editor/Guest Editor for IEEE Communications Magazine, IEEE Wireless Communications, IEEE ACCESS. He has also served as the Technical Program Committee member for many IEEE conferences.

Zimeng Yuan received the B.S. degree in Computer Science from Northeastern University, Shenyang, China, in 2021. He is currently working toward the M.S. degree in Artificial Intelligence from Northeastern University, Shenyang, China. His research interests include edge computing and reinforcement learning.

Mahesh K. Marina (S'00–M'04–SM'13) received the Ph.D. degree in computer science from the State University of New York at Stony Brook, Stony Brook, NY, USA. He had a two-year post-doctoral position with the Computer Science Department, University of California at Los Angeles, Los Angeles, CA, USA. He is a Reader with the School of Informatics, University of Edinburgh, Edinburgh, U.K. In 2013, he was a Visiting Researcher with ETH Zurich, Zürich, Switzerland, and Ofcom, London, U.K. More information about him and his current research can be found at http://homepages.inf.ed.ac.uk/mmarina/. Dr. Marina is a Senior Member of the ACM.



Yufei Liu received the B.S. degree in Computer Science and Technology from Shenyang Aerospace University, China, in 2021. Currently he is pursuing the PH.D. degree in computer science at Northeastern University, Shenyang, China. His research interests mainly include VANETs, SDVN, Traffic Prediction and Service Migration.

Hai Zhao received the B.S. degree in electrical engineering from Dalian Maritime University, Dalian, China, in 1982, and the M.S. and Ph.D. degrees in computer science from Northeastern University, China, in 1987 and 1995, respectively. He is currently a professor with the school of Computer Science and Engineering, Northeastern University, Shenyang, China. He is also the Director of the Liaoning Provincial Key Laboratory of Embedded Technology. His current research interests include embedded Internet technology, wireless sensor networks, vehicular ad hoc networks, body area networks, pervasive computing, operating systems, data and information fusion, computer simulation, and virtual reality.