IEEE ComSoc IEEE Communications Society | IEEE COMPUTER SOCIETY | IEEE Signal Processing Society | VTS | IEEE Transactions on Machine Learning in Communications and Networking

Received 23 January 2024; revised 5 June 2024 and 29 July 2024; accepted 17 August 2024.
Date of publication 26 August 2024; date of current version 2 September 2024.

The associate editor coordinating the review of this article and approving it for publication was G. Yu.

Digital Object Identifier 10.1109/TMLCN.2024.3449831

Objective-Driven Differentiable Optimization of Traffic Prediction and Resource Allocation for Split AI Inference Edge Networks

XINCHEN LYU[1], YUEWEI LI[1], YING HE[2] (Senior Member, IEEE), CHENSHAN REN[3], WEI NI[2] (Fellow, IEEE), REN PING LIU[2] (Senior Member, IEEE), PENGCHENG ZHU[4] (Member, IEEE), AND QIMEI CUI[1] (Senior Member, IEEE)

[1]National Engineering Research Center for Mobile Network Technologies, Beijing University of Posts and Telecommunications, Beijing 100876, China
[2]School of Electrical and Data Engineering, University of Technology Sydney, Ultimo, NSW 2007, Australia
[3]School of Information Engineering, Minzu University of China, Beijing 100081, China
[4]National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China

CORRESPONDING AUTHOR: X. LYU (lvxinchen@bupt.edu.cn)

This work was supported in part by the National Key R&D Program of China under Grant 2021YFB2900302, in part by the National Natural Science Foundation of China under Grant 62371059, in part by the Fundamental Research Funds for the Central Universities under Grant 2242022k60006, and in part by the Joint Funds for Regional Innovation and Development of the National Natural Science Foundation of China under Grant U21A20449.

ABSTRACT Split AI inference partitions an artificial intelligence (AI) model into multiple parts, enabling the offloading of computation-intensive AI services. Resource allocation is critical for the performance of split AI inference. The challenge arises from the time-sensitivity of many services versus time-varying traffic arrivals and network conditions. The conventional prediction-based resource allocation frameworks have adopted separate traffic prediction and resource optimization modules, which may be inefficient due to discrepancies between the traffic prediction accuracy and resource optimization objective. This paper proposes a new, objective-driven, differentiable optimization framework that integrates traffic prediction and resource allocation for split AI inference. The resource optimization problem (aimed to maximize network revenue while adhering to service and network constraints) is designed to be embedded as the output layer following the traffic prediction module. As such, the traffic prediction module can be trained directly based on the network revenue instead of the prediction accuracy, significantly outperforming the conventional prediction-based separate design. Employing the Lagrange duality and Karush-Kuhn-Tucker (KKT) conditions, we achieve efficient forward pass (obtaining resource allocation decisions) and backpropagation (deriving the objective-driven gradients for joint model training) of the output layer. Extensive experiments on different traffic datasets validate the superiority of the proposed approach, achieving up to 38.85% higher network revenue than the conventional predictive baselines.

INDEX TERMS Resource allocation, traffic prediction, differential optimization, split AI inference.

I. INTRODUCTION

THE integration of artificial intelligence (AI) and communication is considered as a crucial usage scenario for the upcoming sixth-generation (6G) networks, as highlighted in the ITU-R report for the framework of IMT-2030 [1]. As recommended in the 3rd generation partnership project (3GPP) TR 22.874 [2], split AI inference can facilitate the computation offloading of AI services in edge computing networks and alleviate the limitation of delay, resources, and the number of devices [3]. The split AI inference involves dividing AI models, such as deep neural networks (DNNs), into multiple parts based on the splitting points, which can be deployed or offloaded to different edge servers [4]. In split AI inference, the intermediate computation results of each part of a model are transmitted to the corresponding edge server. As specified in the 3GPP [2], the optimal splitting points can help reduce the resource and energy consumption of resource-constrained local devices [5], [6], decrease the end-to-end delay, and improve the accuracy, efficiency, and privacy of networks [3], [7].

There are various existing studies to optimize split AI inference with the objectives of communication latency [8], [9], energy consumption [10], throughput [11], and quality of service (QoS) [12]. Resource allocation (including split decisions) is a critical concern in split AI inference for computation-intensive models. One notable approach is Neurosurgeon [3], which introduced an algorithm to identify the optimal layer for splitting DNN models while minimizing energy consumption and latency. Additionally, other studies [13], [14] jointly optimized the allocation of computation and communication resources.

In existing approaches, the challenges of conventional frameworks arise from the lack of time-sensitive service provisioning capability when dealing with time-varying traffic arrivals across a network [15], [16]. First, the static resource allocation methods are inadequate for accommodating fluctuations in traffic inference and, consequently, often result in violations of service level agreements (SLAs) [17]. They fail to tackle fluctuations in network traffic and resource availability in a dynamic network. Second, to tackle the issue of time-varying network traffic, prediction-based resource optimization approaches have been proposed. They predict network traffic using different prediction methods. As illustrated in Fig. 1(a), a traffic prediction module, such as the long short-term memory (LSTM) network, can forecast network traffic patterns, and resources can be proactively allocated to handle incoming traffic according to the prediction. Despite their advantages, existing prediction-based resource allocation approaches suffer from inefficiency due to the disjointed design structure of the traffic prediction and resource allocation modules.

The motivation of this paper lies in addressing the challenge of effectively provisioning services in networks experiencing time-varying traffic arrivals. In conventional frameworks, inefficiencies arise primarily because conventional traffic prediction modules prioritize the accuracy of future traffic prediction by minimizing the mean squared error (MSE) between predicted and ground-truth traffic patterns. To meet the requirements of edge networks for split inference as specified in the 3GPP [2], novel frameworks are anticipated to solve the problem that is difficult to account for the broader network objectives (e.g., maximizing throughput or revenue.) in conventional approaches.

In this paper, we propose a novel, objective-driven, differentiable optimization framework. Distinctively different from the existing separate designs of traffic prediction and resource allocation, the proposed framework integrates these two modules for split AI inference in edge networks. This integration (i.e., the objective-driven design) enables the proposed framework to be trained based on the network revenue, instead of the traffic prediction accuracy. Specifically, we formulate the resource allocation problem to maximize network revenue under computational and bandwidth resource constraints, and propose to embed the problem as the output layer of the traffic prediction AI model. As a result, the model accounts for both traffic prediction and resource allocation. By contrast, the existing models can only capture traffic prediction and optimize traffic prediction accuracy.

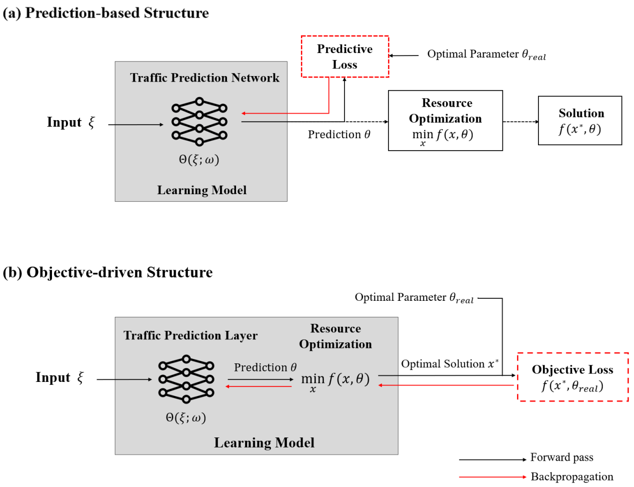

FIGURE 1. Comparison between different frameworks. (a) The existing prediction-based framework, which adopts the separate design of traffic prediction and resource optimization; (b) The objective-driven integrated framework, which embeds the optimization as the output layer of the traffic prediction module. In the objective-driven framework, the traffic prediction module is trained to maximize the resource allocation objective (e.g., network revenue), instead of minimizing prediction errors.

To achieve this integration, the concept of differentiable optimization [18], [19] is leveraged, which enables the generation of gradients between the loss function and the parameters involved in the differentiable optimization of a learning model. By harnessing differentiable optimization, the traffic prediction and resource allocation modules can be seamlessly integrated. This approach allows the traffic prediction module to be trained to optimize network revenue directly, rather than minimize prediction error as in conventional approaches. The proposed framework offers a new solution for establishing a more efficient approach to time-varying traffic provisioning in edge networks.

The key contributions of this paper are listed as follows.

- We formulate a new resource allocation problem for time-sensitive split AI inference network in the presence of uncertain traffic and network re heterogeneity, where the resources must be allocated without knowing the upcoming traffic arrivals and network conditions.
- We propose an objective-driven framework that combines network traffic prediction and resource optimization, hence integrating the optimization problem as the output layer of a learning model. This integration facilitates training the traffic prediction module to maximize

TABLE 1. **The summary of existing resource optimization studies based on reinforcement learning and deep learning.**

Approach	Work	Architecture	Resource Constraints	Objective-driven	Learning Model	Contribution
Reinforcement Learning	[20]	E	C	Yes	Collective DRL	Use MDP to allow nodes to share information.
	[21]	E+D	C+B+E	Yes	DRL	Maximize profits by two-stage Stackelberg game.
	[22]	C+E+D	C+B+E	Yes	QRL	Resource management and task offloading in MEC.
	[23]	E+D	C+B+E	No	LSTM+DRL	Minimize cost by combining prediction and DRL.
	[24]	E	C+L	Yes	DQN	Propose improved DQN to manage the resource.
Deep Learning with Optimization	[25], [26]	BS	B+E	No	LSTM	Introduce BS strategy by LSTM and optimization.
	[27]	C+E	Utility	No	LSTM	Predict user data and allocate the resource on cell.
	[28]	E+D	E+L	No	Attention-based LSTM	Predict network traffic and dispatch different resource dynamically in URLLC.
	[29]	E+D	C+L	No	LSTM	Predict traffic and balance the distributed resource.
	[30]	E+D	L	No	ConvLSTM	Predict slices traffic and solve resource problem to reduce delay by interior-point method.
Differentiable Optimization	Ours	C+E	C+B	Yes	LSTM+ Differentiable layer	Integrate optimization into learning-based model.

Architecture: {C:center, E:edge, D:device, BS:base station}, Resource: {C:computation, B:bandwidth, E:energy, L:latency}

network revenue, instead of only minimizing the prediction error.

- We generate efficient resource allocation decisions and derive the objective-driven gradients for joint model training by employing the Lagrange duality and Karush-Kuhn-Tucker (KKT) conditions of the output layer (i.e., the differentiable optimization module).
- Extensive experiments are conducted on different network traffic datasets to validate the superiority of the proposed objective-driven framework to the existing methods. As compared to the conventional prediction-based separate design, the proposed framework can achieve up to 38.85% higher network revenue.

The rest of the paper is organized as follows. Section II provides an overview of the related works. Section III presents the system model in edge intelligence. formulates the resource management problem. Section IV formulates the resource allocation problem and introduces the objective-driven approach and different models of the proposed framework. The experiment results are analyzed in Section V, followed by a conclusion in Section VI.

II. RELATED WORK

This section provides an overview of research studies related to split AI inference and compares different methods in the topics of predictive resource allocation. A comparison study of the existing studies is presented in Table 1, where existing resource allocation frameworks are based on their structures, network resources, and learning models.

A. REINFORCEMENT LEARNING-BASED RESOURCE ALLOCATION

In resource-constrained systems, efficient resource management and balancing network traffic among different AI services are crucial. Markov decision process (MDP) provides a suitable framework for addressing sequential decision-making problems. Reinforcement learning (RL) offers various methods to solve MDP problems [24]. Xiong et al. [24] formulated a resource allocation problem at the edge node as an MDP and implemented a deep Q-network (DQN) algorithm that utilizes multiple replay memories to reduce resource consumption.

Several studies have applied RL to address resource allocation problems. These frameworks considered computing capability and channel conditions by treating resource consumption minimization or network revenue maximization as system objectives. They monitored the network state and selected optimal allocation schemes [20], [21]. Wang et al. [22] applied RL to an edge intelligence network, considering various resources, such as computational resource, bandwidth, and energy consumption, to reduce latency. Tu et al. [23] proposed an Online Predictive Offloading algorithm that combines a LSTM network and a deep reinforcement learning (DRL) model. However, these RL-based resource allocation approaches may be ineffective, as historical traffic data are not fully exploited.

B. DEEP LEARNING-BASED RESOURCE ALLOCATION

DNN models are also efficient for solving traffic prediction and resource optimization problems. Fattore et al. [29] explored the use of LSTM to predict user mobility and make distributed resource allocation decisions in the mobile edge computing (MEC) network. Zhu and Wang [25] developed an intelligent resource allocation algorithm for base stations (BSs) by utilizing LSTM to predict traffic distributions. The prediction results provided insights into traffic demand and served as a constraint in their resource allocation problem.

Some studies have developed advanced predictive models based on LSTM for efficient resource allocation. Li et al. [26] combined LSTM with historical average values to predict the traffic demands of a BS to solve a power control problem. To optimize resource allocation, convex problems can be combined with the predictive models [27], [30]. To capture the interaction between services and users, attention-based LSTM models can be utilized to predict traffic flow and adaptive resource allocation models can improve efficiency under resource constraints based on the prediction results [28]. Kavehmadavani et al. [31] proposed a joint intelligent framework that integrates traffic prediction and flow split distribution. However, as illustrated in Fig. 1(a), the efficiency of the prediction-based resource allocation framework may be compromised due to the separate design architecture of traffic prediction and resource allocation.

As shown in Table 1, we compare the existing resource optimization studies based on different RL or deep learning frameworks. Several limitations of existing works are identified. RL-based approaches addressed different sequential decision-making problems and failed to predict the traffic data using the historical data. These approaches cannot output accurate prediction results. In deep learning-based approaches, the efficiency may be compromised because the traffic prediction and resource allocation modules are separated. In other words, deep learning-based approaches make it difficult to generate an objective-driven framework. They only aim at accurate predictive traffic, not the network revenue.

C. DIFFERENTIABLE OPTIMIZATION

The OptNet framework [18] has introduced a promising breakthrough by incorporating constrained optimization as a layer within deep learning architectures. This framework establishes the relationship between model-based optimization problems and AI models, enabling the use of exact gradients through differentiation of the KKT conditions of a quadratic program [32]. This allows for objective-driven model training. Although the OptNet framework has been applied in various research areas, such as adversarial behavior learning [33], power provisioning [34], and graph matching [35], its application in resource allocation has not been adequately explored in the existing literature. Our work aims to bridge this research gap by integrating prediction and optimization modules and proposing an objective-driven, differentiable optimization approach for traffic prediction and resource allocation in the context of split AI inference. This integration allows for a unified framework that combines the benefits of deep learning-based prediction models with the optimization capabilities of the OptNet framework.

III. SYSTEM MODEL

This section introduces a split AI inference edge network and formulates the resource-constrained problem. As illustrated in Fig. 2, our analysis focuses on the structure of the edge network, comprising a central controller and interconnected edge servers via communication links. We represent the set of all edge servers as $\mathcal{N} = \{n_1, n_2, \ldots, n_N\}$, where N represents the total number of edge servers. Similarly, the set of all communication links is denoted as $\mathcal{E} = \{e_1, e_2, \ldots, e_E\}$, where E represents the total number of communication links. The notations used in this paper are summarized in Table 2.

TABLE 2. Definitions of notations.

Notation	Definition
K	Number of services
N	Number of edge servers
E	Number of communication links
c_i^k	Computation data density of service k on server i
$b_{(i,j)}^k$	Bandwidth data density of service k on link (i,j)
$d_k(t)$	Admitted input data of service k at slot t
$\mathbf{d}$	Collective admitted input data $\{d_1, d_2, \ldots, d_K\}$
$D_k(t)$	Data arrival of service k at slot t
C_i	Maximum computation resource of edge server i
$B_{(i,j)}$	Maximum bandwidth resource of link (i,j)
r_k	Unit income of service k
q_C	Unit cost of computation resource
q_B	Unit cost of bandwidth resource
p_k	Unit revenue of service k
$\mathbf{p}$	Collective revenue $\{p_1, p_2, \ldots, p_K\}$
G, h	Combination of parameters of inequality constraints
A^T	The transpose of matrix A
$diag(\cdot)$	The diagonal matrix from a vector

A. SPLIT AI INFERENCE MODEL

As stated in Section I, split AI inference partitions an AI model into multiple parts to be deployed at different network servers. Based on [2], [3], and [7], this split AI inference strategy considers the resource consumption, end-to-end delay, and private information security in splitting networks. The strategy is pre-determined. We define the indexes of different AI service models using $\mathcal{K} = \{1, 2, \ldots, K\}$, where K is the total number of AI services [4]. The split parts of a specific model are indexed by $M_k = \{M_1^k, M_2^k, \ldots, M_{m_k}^k\}$, where M_i^k $(1 \leq i \leq m_k)$ is the i-th part of AI service k and m_k denotes the number of partitions of this model.

The partitioned models are deployed to the edge servers. In this paper, we consider the case where the models are pre-deployed at the servers by adopting the existing approaches. For example, Hivemind [4] is a popular method to solve the model splitting and deployment problem efficiently, which inherits the logic of the distributed Dijkstra's algorithm. After deploying the AI models, the system operates in a chain structure. As shown in Fig. 2, each edge server processes its received intermediate data/original input by running its owned part, and sends the processed results to the next edge server (owning the subsequent model part). The framework security is based on the split AI partitioning strategy and can improve the security of private information [2]. For instance, our previous work [7] considered the

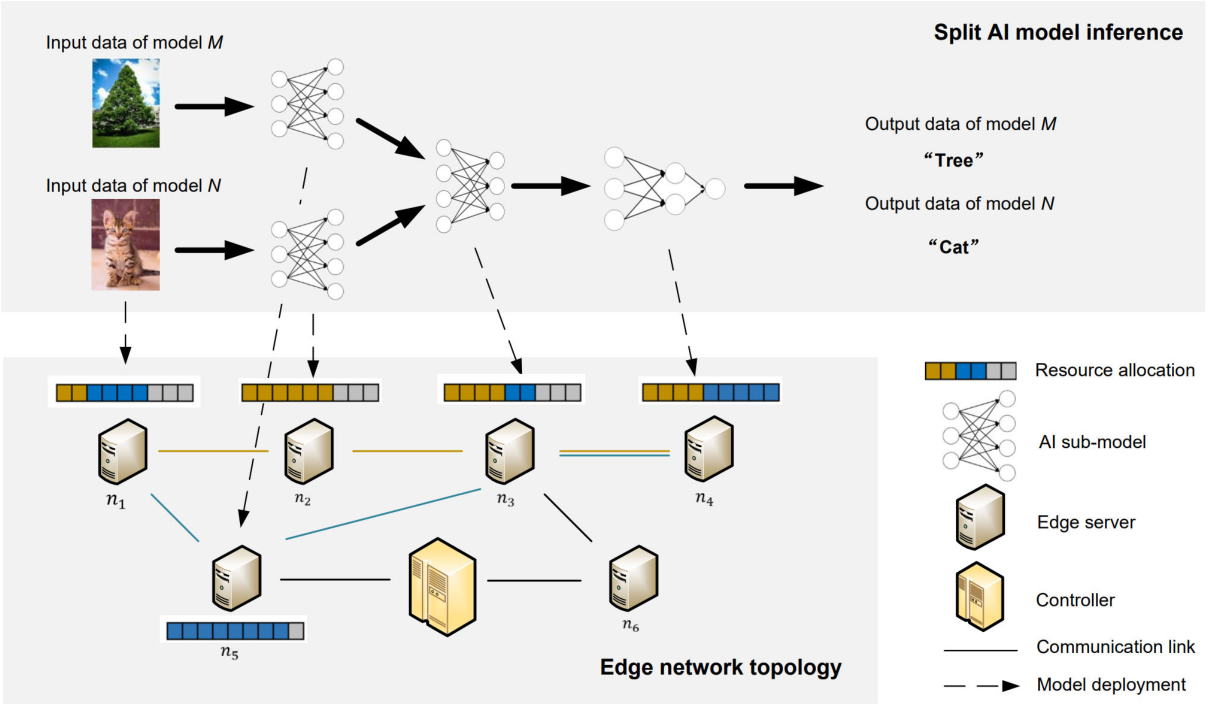

FIGURE 2. An illustrative example of the split AI inference network. The AI model is partitioned into multiple parts that are deployed at the edge servers. The data (e.g., image) can be processed along the path of the deployed model parts. The controller is responsible for allocating the resources for time-sensitive services in the presence of uncertain traffic and network heterogeneity.

security-efficiency tradeoff in split AI inference to alleviate the risk of two passive attacks for recovering the input data and private labels.

To this end, each service k can be described by $S_k(Path_k, c^k, b^k)$. In particular, $Path = \{Path_k | k \in \mathcal{K}\}$ denotes the set of ordered links between the edge servers used in service k [36]. For instance, the path of service N is denoted by $Path_N = \{(n_1, n_2), (n_2, n_3), (n_3, n_4)\}$ in Fig. 2. This indicates that the computation path of the service contains edge servers $\{n_1, n_n, n_3, n_4\}$. Here, c^k and b^k represent the data density (overhead) of service k regarding server processing and link transmission, respectively. $c_i^k \in R$ denotes the data density at edge server i, which represents the computation resource consumption for each unit of input data of service k on server i. $b_{(i,j)}^k \in R$ denotes the data density at the link connecting servers i and j, which represents the bandwidth consumption for each unit of input data of service k. Let $c^k = [c_i^k] \in R^{m_k}$ and $b^k = b_{(i,j)}^k \in R^{m_k - 1}$ collect the data density factors along the path.

Different from data transmission, the input and output data sizes may not be equal, depending on the number of neurons at the cut layer of the part. The above data density factors, c^k and b^k, are unified and defined based on the consumption per unit input data size. Moreover, the data density factors depend on the SLA (QoS metric), i.e., how many resources can guarantee on-time service provisioning for a unit size of input traffic.

B. TRAFFIC MODEL AND ADMISSION CONTROL

Let $D_k(t)$ denote the volume (e.g., in Mb) of arrived traffic at time slot t for service k. We consider the practical but challenging scenario [37], [38], where the resource allocation must be optimized without knowing the current data arrival $D_k(t)$. Particularly, $D_k(t)$ can only be observed at the end of each slot t. In other words, the framework needs to predictively allocate resources based on historical traffic arrivals in the previous slots $\{0, 1, \cdots, t-1\}$ [27].

Note that the resource allocation (i.e., preserving how many computational and bandwidth resources at the edge server/inter-server links) determines the maximum value of accommodated data. Admitting more traffic would result in the violation of SLA. Let $d_k(t)$ be the size (e.g., in Mb) of admitted input data for service k at slot t. $d_k(t)$ must not exceed the total size of traffic arrivals, i.e.,

$$0 \le d_k(t) \le D_k(t) \quad \forall k \in \mathcal{K}. \tag{1}$$

Recall that the data density factors specify the minimum required computation/bandwidth resources to accommodate a unit size (e.g., per Mb) of input data within the SLA requirement. Given the admitted data size $d_k(t)$, the edge server $i \in Path_k$ and link $(i, j) \in Path_k$ need to reserve $c_i^k d_k(t)$ computational resources and $b_{(i,j)}^k d_k(t)$ bandwidth resources for service k, respectively.

Let C_i (e.g., in GHz) and $B_{(i,j)}$ (e.g., in Mbps) denote the maximum computational and bandwidth resources of edge server i and inter-server link (i, j), respectively [23]. Given the resource limitation, the data admission (i.e., resource allocation) must also satisfy

$$\sum_k c_i^k d_k(t) \leq C_i \quad \forall i \in Path_k, k \in \mathcal{K}; \tag{2a}$$

$$\sum_k b_{(i,j)}^k d_k(t) \leq B_{(i,j)} \quad \forall (i,j) \in Path_k, k \in \mathcal{K}. \tag{2b}$$

IV. OBJECTIVE-DRIVEN DIFFERENTIABLE OPTIMIZATION OF TRAFFIC PREDICTION AND RESOURCE ALLOCATION

This section introduces the formulated problem, the overall framework of the differentiable optimization framework, and outlines the process of solving the optimization problem. Fig. 3 shows the framework of the proposed, objective-driven, differentiable optimization, which resembles the structure presented in Fig. 1(b) and aims to maximize network revenue rather than minimize the prediction error. The optimization problem is embedded as the output layer following the network traffic prediction module. As shown in Fig. 3, the proposed framework integrates two parts, i.e., traffic prediction and differentiable optimization.

A. PROBLEM FORMULATION

The infrastructure provider (i.e., the owner of edge servers) is responsible for allocating the resources. Its objective is to maximize the network revenue of split AI inference services at each time slot t. The network revenue is made up of two parts, including: 1) the revenue paid by the service provider according to SLA for successfully accommodated data, and 2) the costs of allocating the resources (e.g., resource consumption for computations).

Let r_k denote the unit income for accommodating a unit size of data (e.g., in \$/MB). q_C and q_B are the costs for allocating computational and bandwidth resources (also measured in dollars). The problem of maximizing the network revenue can be formulated as

$$\begin{aligned} \max_{d_k(t)} \quad & \sum_{k \in \mathcal{K}} \{ r_k d_k(t) - q_C \sum_{i \in Path_k} c_i^k d_k(t) \\ & - q_B \sum_{(i,j) \in Path_k} b_{(i,j)}^k d_k(t) \} \\ s.t. \quad & (1), (2a), (2b), \end{aligned} \tag{3}$$

where the data admission $d_k(t)$ specifies the resource allocation, as stated in (2).

Problem (3) is a linear programming (LP) problem. However, as stated earlier, we consider a practical but challenging scenario, where the resources must be allocated predictively without the prior knowledge of the current data arrival $D_k(t)$. Additionally, we further consider that the maximum resource parameters C_i and $B_{(i,j)}$ cannot be directly observed, as the exposure of the maximum resources would increase the risks of cyberattacks. The controller can only learn some information parameters from successfully accommodated traffic at each slot.

B. OBJECTIVE-DRIVEN LSTM AND DIFFERENTIABLE OPTIMIZATION MODULES

- **Objective-driven LSTM Module.** The LSTM traffic prediction module is used to predict the arriving data for each service k at each time slot t. The LSTM module follows the same structure as the widely adopted LSTM network, whereas its training objective is to maximize network revenue, i.e., the gradients of Problem (3) generated from the differentiable optimization module. Please refer to Section IV-B-II) for details.
- **Differentiable Optimization Module.** The differentiable optimization module aims to optimize the resource allocation decisions for Problem (3). The optimal resource allocation strategy maximizes the network revenue. This module is also responsible for 1) embedding Problem (3) as a differentiable optimization layer following the LSTM module, and 2) approximating unobservable network resources and calculating the objective-driven gradients for the training of the prediction module. Please refer to Section IV-B-I) for details.

By incorporating differentiable optimization, we integrate traffic prediction and resource allocation modules. The traffic prediction module can be trained to maximize the network revenue instead of minimizing the prediction error.

1) DIFFERENTIABLE OPTIMIZATION MODULE

Problem (3) is interpreted as a layer for training the objective-driven model. This module generates the resource allocation decisions $d_k(t)$ based on the predictive traffic from the LSTM module. We solve the problem in this module to obtain the optimal resource allocation strategy. As shown in Fig. 3, the predictive traffic $D_k(t)$ of the LSTM module serves as the input of the differentiable optimization module, and the output is the optimal solution of Problem (3). The network resource parameters, i.e., C_i and $B_{(i,j)}$, need to be predicted or approximated in the module. There are two types of gradients required in the differentiable layer, including: 1) $\frac{\partial l}{\partial D_k(t)}$, which is used to train the LSTM prediction module in an objective-driven manner, where l represents the loss (e.g., the MSE between output and target) of the solution, and 2) $\frac{\partial l}{\partial C_i}$ and $\frac{\partial l}{\partial B_{(i,j)}}$, which are used to approximate the optimal network resources C_i and $B_{(i,j)}$, respectively.

To obtain these gradients for Problem (3), we combine all inequality constraints and redefine the parameters as

$$G = \begin{bmatrix} [c_i^k] \\ [b_{(i,j)}^k] \\ I \\ -I \end{bmatrix} \in R^{(2K+E+N) \times K}; \tag{4a}$$

$$h = \begin{bmatrix} C_i \\ B_{(i,j)} \\ D_k(t) \\ 0 \end{bmatrix} \in R^{(2K+E+N)}; \tag{4b}$$

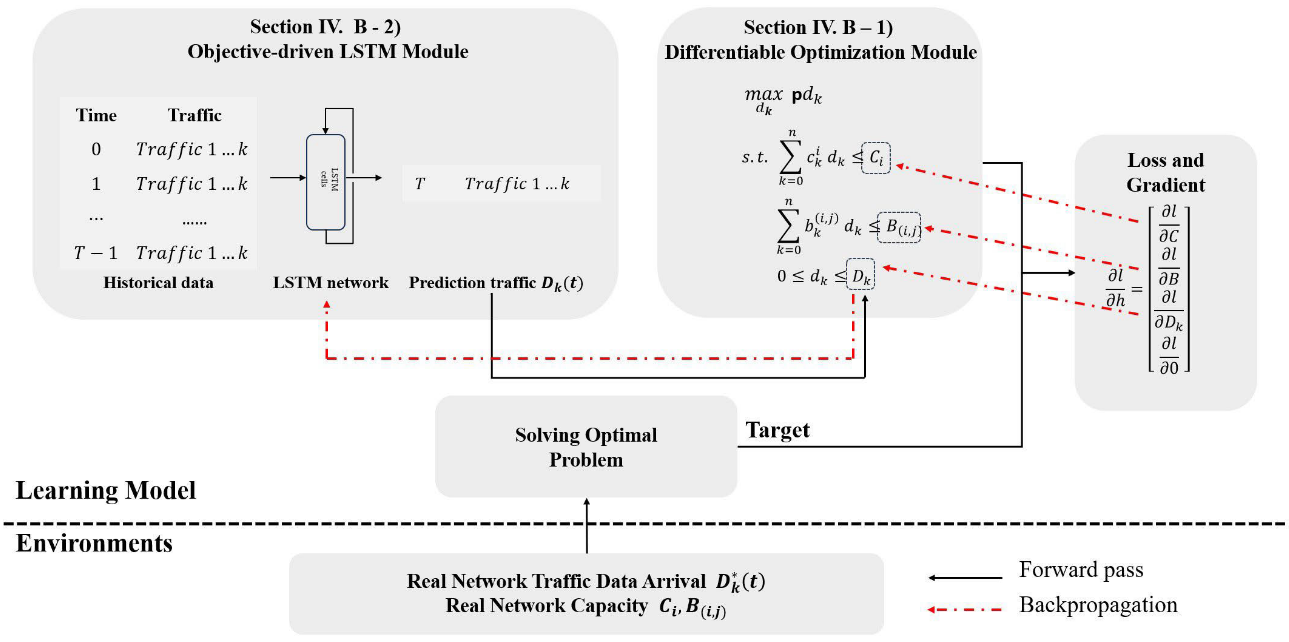
Section IV. B - 2)
Objective-driven LSTM Module
Time
Traffic
0
$Traffic\ 1 \ldots k$
1
$Traffic\ 1 \ldots k$
…
......
$T-1$
$Traffic\ 1 \ldots k$
Historical data
LSTM cells
LSTM network
T
$Traffic\ 1 \ldots k$
Prediction traffic $D_k(t)$
Section IV. B – 1)
Differentiable Optimization Module
$\max_{d_k} \mathbf{p}d_k$
$s.t. \sum_{k=0}^{n} c_k^i d_k \leq C_i$
$\sum_{k=0}^{n} b_k^{(i,j)} d_k \leq B_{(i,j)}$
$0 \leq d_k \leq D_k$
Loss and Gradient
$\frac{\partial l}{\partial h} = \begin{bmatrix} \frac{\partial l}{\partial C} \\ \frac{\partial l}{\partial B} \\ \frac{\partial l}{\partial D_k} \\ \frac{\partial l}{\partial 0} \end{bmatrix}$
Solving Optimal Problem
Target
Learning Model
Environments
Real Network Traffic Data Arrival $D_k^*(t)$
Real Network Capacity $C_i, B_{(i,j)}$
Forward pass
Backpropagation

FIGURE 3. The framework of the proposed, objective-driven, differentiable optimization of traffic prediction and resource allocation.

$$p_k = r_k - q_C \sum_{i \in Path_k} c_i^k - q_B \sum_{(i,j) \in Path_k} b_{(i,j)}^k; \qquad (4c)$$

$$\mathbf{p} = \left[p_1, p_2, \ldots, p_K\right] \in R^K, \qquad (4d)$$

where I is the identity matrix, $[c_i^k] \in R^{N \times K}$ and $[b_{(i,j)}^k] \in R^{E \times K}$ represent the collection matrices of c_i^k and $b_{(i,j)}^k$, respectively; p_k represents the unit revenue of service k and $\mathbf{p}$ represents the collective unit revenue across all services. For brevity, let $\mathbf{d} = \{d_1(t), d_2(t), \cdots, d_K(t)\} \in R^K$ represent the collective admitted traffic of all services in time slot t. Then, Problem (3) can be reformulated as

$$\min_{\mathbf{d}} \quad -\mathbf{p}^T\mathbf{d} \qquad (5a)$$

$$s.t. \quad G\mathbf{d} \leq h. \qquad (5b)$$

Problem (5) is an LP problem and can be solved applying Lagrange duality. Let $\lambda \in R^{(2K+E+N)}$ ($\lambda \geq 0$) be the dual variable associated with the inequality constraint (5b). The Lagrangian of (5) can be written as [18]

$$L(\mathbf{d}, \lambda) = -\mathbf{p}^T\mathbf{d} + \lambda^T(G\mathbf{d} - h), \qquad (6)$$

where h is the differentiable optimization parameter in (4b) that can be affected by variables $C_i(t)$ and $B_{(i,j)}(t)$. The KKT conditions of this Lagrangian are

$$-\mathbf{p} + G^T\lambda^* = 0, \qquad (7a)$$

$$(\lambda^*)^T(G\mathbf{d}^* - h) = 0, \qquad (7b)$$

where $\mathbf{d}^*$ and λ^* are the optimal primal and dual variables, respectively. Based on KKT conditions, we can obtain the closed-form expressions for the optimal dual variable λ^* and network revenue (i.e., objective), which must satisfy

$$G^T\lambda^* = \mathbf{p}, \qquad (8a)$$

$$-\mathbf{p}^{\mathbf{T}}\mathbf{d}^* = \inf L(\mathbf{d}, \lambda) = -h^T\lambda^*. \qquad (8b)$$

This provides an effective way for the forward pass of the output layer to obtain the optimal resource allocation decisions.

We proceed to calculate the derivative of l with respect to h for efficient backpropagation, such that the resource optimization and traffic prediction modules can be directly trained based on the loss function (i.e., maximizing the network revenue). We take the derivatives of these conditions from (7) associated with $\mathbf{d}^*$, λ^*, and h, i.e.,

$$G^T d\lambda = 0, \qquad (9a)$$

$$diag(G\mathbf{d}^* - h)d\lambda + diag(\lambda^*)(G(d\mathbf{d}) - dh) = 0, \qquad (9b)$$

where $diag(\cdot)$ represents diagonal matrix. It can be rewritten in the matrix form as

$$\begin{bmatrix} 0 & G^T \\ diag(\lambda^*)G & diag(G\mathbf{d}^* - h) \end{bmatrix} \begin{bmatrix} d\mathbf{d} \\ d\lambda \end{bmatrix} = \begin{bmatrix} 0 \\ diag(\lambda^*)dh \end{bmatrix}. \qquad (10)$$

The derivatives of these conditions (10) give the gradient of $\mathbf{d}$ with respect to h. The Jacobian $\frac{\partial \mathbf{d}^*}{\partial h}$ can be computed from (10). Let $dh = I$ and then solve this equation. The result of $d\mathbf{d}$ is the Jacobian $\frac{\partial \mathbf{d}^*}{\partial h}$ for gradient calculation [18].

Next, we clarify the relationship between the loss and the unobservable parameters C_i and $B_{(i,j)}$. In the backpropagation process, the framework gets the backward pass vector $\frac{\partial l}{\partial \mathbf{d}} \in R^K$. The gradient can be obtained based on the chain rule $\frac{\partial l}{\partial h} = \frac{\partial l}{\partial \mathbf{d}} \cdot \frac{\partial \mathbf{d}}{\partial h} \in R^{(2K+E+N)}$. We multiply the inverse of the

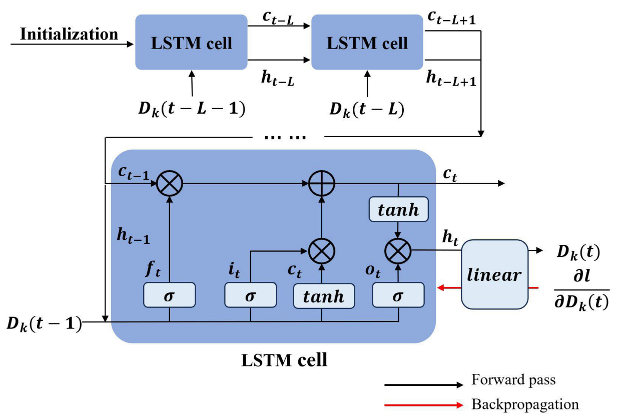

FIGURE 4. **The structure of the objective-driven LSTM network.**

```
Algorithm 1 Integrating Differentiable Optimization With Learning-Based Traffic Prediction

Initialize the network topology $\{\mathcal{N}, \mathcal{E}\}$, the attributes of services $\{S_k\}$, model partition set $\{M_k\}$, the traffic dataset $\{D_K\}$
Input the network traffic data;
for time slot $t = 1 \dots T$:
    Predict traffic flow $D_k(t)$;
    Solve Problem (3) and obtain solutions $\mathbf{d}$, $\lambda_k^*$;
    Apply this allocation strategy $d_k(t)$;
    Calculate the target in (3);
    Obtain MSE $l$ and $\frac{\partial l}{\partial h}$ by (12);
    Update model parameters;
End for.
```

left matrix on the right-hand side of (10), i.e.,

$$\begin{bmatrix} d_{\mathbf{d}} \\ d_\lambda \end{bmatrix} = \begin{bmatrix} 0 & G^T \\ diag(\lambda^*)G & diag(G\mathbf{d}^* - h) \end{bmatrix}^{-1} \begin{bmatrix} (\frac{\partial l}{\partial \mathbf{d}^*})^T \\ 0 \end{bmatrix}. \qquad (11)$$

The corresponding derivative of l with respect to h in Problem (5b) can be given by

$$\frac{\partial l}{\partial h} = -diag(\lambda^*)d\lambda = \begin{bmatrix} \frac{\partial l}{\partial C_i} \\ \frac{\partial l}{\partial B_{(i,j)}} \\ \frac{\partial l}{\partial D_k(t)} \\ \frac{\partial l}{\partial 0} \end{bmatrix}. \qquad (12)$$

The gradient (12) consists of all required vectors to update the objective-driven LSTM module (i.e., $\frac{\partial l}{\partial D_k(t)}$) the unobservable resources (i.e., $\frac{\partial l}{\partial C_i}$ and $\frac{\partial l}{\partial B_{(i,j)}}$). During the backpropagation, the computational resource C_i and bandwidth resource $B_{(i,j)}$ can be approximated by employing MSE loss and the gradients in (12).

2) OBJECTIVE-DRIVEN LSTM MODULE

This module predicts data arrivals. LSTM is a crucial variant of recurrent neural network (RNN) that addresses the challenges of gradient disappearance and explosion in traditional RNNs [39]. Fig. 4 shows the structure of the objective-driven LSTM module, which follows the standard LSTM network by cascading LSTM cells [40].

Each LSTM cell includes four gates: input $D_k(t)$, forget f_t, cell state c_t and output o_t gate [40]. The output gate calculates the generation of the output based on the cell state and input data. The forget gate determines the extent, to which the previous information should be disregarded in the current context. The updating of the cell state is based on the input and the previous output of the cell. The input of the LSTM module is a window of the previous traffic $\{D_k(t-L), \dots, D_k(t-1)\}$, where L represents the sequence length of LSTM cells [41]. The output of the module is the predictive traffic arrival $D_k(t)$ of all services at slot t, which also serves the admission control constraint in Problem (3).

Unlike conventional prediction-based frameworks, the LSTM network is trained based on the objective of Problem (3). It does not explicitly minimize the MSE of traffic prediction. As shown in Fig. 4, the training metric is the gradient $\frac{\partial l}{\partial D_k(t)}$ derived from Problem (3). In other words, the LSTM network is updated to optimize the overall performance.

C. IMPLEMENTATION IN MODEL-SPLIT EDGE NETWORK

The proposed algorithm is illustrated in Algorithm 1. The network topology and service attributes are generated in the initialization stage. Each service is split into several partitions and deployed on the edge servers. During the training process, the framework first predicts the arrival traffic $D_k(t)$ of this slot based on historical data, and then passes $D_k(t)$ to the differentiable optimization module. In other words, these two modules are connected by the prediction $D_k(t)$, which serves as both the output of the prediction module and the input of the resource allocation module. After receiving $D_k(t)$ to determine the admitted traffic constraint, the resource allocation module solves the LP Problem (3) to obtain the optimal primal variable $\mathbf{d}^*$ and the optimal dual variable λ^*. The objective of the resource allocation is derived from the optimal parameters. The MSE between the objective and the output serves as the loss function. In the backpropagation step, the updating gradient of the model (i.e., $\frac{\partial l}{\partial h}$) is obtained from the differentiable optimization layer to approximate network resources C_i and $B_{(i,j)}$ and update the LSTM prediction module.

Fig. 3 illustrates the interaction between the controller and the environment in Algorithm 1. The learning model is deployed at the controller. At the beginning of each slot t, the controller predicts the traffic arrival $D_k(t)$ for each service and passes it to the differentiable optimization module. The resulting resource allocation strategy $\mathbf{d}^*$ is obtained by solving the LP problem in (5). The edge servers reserve the corresponding resources for the traffic at this time slot. The actual traffic arrivals $D_k^*(t)$ are available at the end of the time slot. Accordingly, the controller calculates the loss and the backpropagation gradient using (12) to update all parameters, including the prediction module and the differentiable optimization module.

At each time slot, the complexity of the proposed framework is $\mathcal{O}(L+(K+E+N)^3)$, encompassing: 1) the forward pass with a linear complexity of $\mathcal{O}(L+K+E+N)$, where the complexity of the LSTM module is $\mathcal{O}(L)$ [40] and the complexity of the resource allocation module is $\mathcal{O}(K+E+N)$ due to the availability of the closed-form solution in (8), and 2) the backpropagation with a complexity of $\mathcal{O}(L+(K+E+N)^3)$, where the linear LSTM backpropagation incurs a complexity of $\mathcal{O}(L)$ and the resource allocation module incurs a complexity of $\mathcal{O}((K+E+N)^3)$ [18]. The complexity of the proposed framework scales only cubically with the numbers of servers N, links E, and services K, and can be reasonably implemented in real-time. Consequently, the traffic volume and network size determine the complexity and convergence speed of Algorithm 1.

Based on the time complexity analysis, the complexity grows cubically with the numbers of services, servers, and links, which affects the scalability of the proposed framework. Because the numbers of servers, links, and services affect the dimensions of the corresponding parameters in the proposed LP Problem (3), the change of network topology or services would result in a new LP problem in the optimization module. Nevertheless, the approximate values of the unobservable resources (i.e., C_i and $B_{(i,j)}$) are still valuable after this change. Moreover, a new network service can also result in the reconstruction of the LSTM prediction module due to the fact that LSTM (and other conventional prediction networks) can only accept fixed input dimensions [42], [43], [44].

V. EXPERIMENTAL RESULTS

In this section, we conduct extensive experiments on different network traffic datasets to validate the superiority of the proposed objective-driven framework. In the following, we present the experimental settings and analyze the results.

A. EXPERIMENTAL SETTING AND BENCHMARKS

We have developed a Python-based platform to evaluate the proposed framework. The experimental setup is summarized in Table 3. The topology and the historical traffic arrivals of our experiments are extracted from the TOTEM Project dataset [45]. The dataset provides a network topology consisting of 23 nodes and 74 links, and records network traffic data on a time slot basis. We employ three different DNN-based services: AlexNet, VGG, and FACE with the optimal partitioning points according to [3], and conduct extensive experiments, where the density parameters represent the resource consumption for each unit of traffic data, as specified in [2]. The optimal partitioning points help balance computational and bandwidth resource consumption, and the framework energy consumption of the splitting network.

TABLE 3. **The experimental parameters.**

Parameters	Value
AI service number	3
Edge server number	23
Communication link number	74
LSTM mid-layer	40
LSTM data length	50
Time slot	700
Computational resource distribution	$N(100, 20^2)$
Bandwidth resource distribution	$N(80, 20^2)$

The computational resource C_i and bandwidth resource $B_{(i,j)}$ of each edge server and link follow a normal distribution [46], [47], [48] to capture the network heterogeneity. Compared with uniform distributions [49], [50], normal distributions make it easier to tackle the heterogeneity of resources in a network (i.e., the standard deviations of the distributions). Moreover, the normally distributed data points are closer to their mean, thereby reducing excessive errors caused by unobservable resources in the benchmarks.

To deploy models at the servers, we utilize the Greedy on Used Server (GUS) algorithm [51]. This algorithm aims to minimize the number of occupied servers by prioritizing the placement of models on already used servers, thus avoiding the allocation of additional spare servers. We apply this algorithm to the edge intelligence framework, deploying each partition on servers. This allows resource starvation scenarios to be simulated in the experiments.

For comparison purposes, we consider the following benchmarks.

- **Non-splitting framework:** This framework uses the conventional non-splitting framework, which does not split AI inference models. This benchmark deploys each model to one server. Therefore, the efficiency of each service depends on this server. The non-splitting framework consists of a traffic prediction and an optimization module, which are the same as the predictive optimization framework.
- **Predictive optimization framework:** This framework consists of a traffic prediction layer and a simple optimization layer, which has the same structure as in [27], [29], and [30]. The prediction layer predicts arriving data in the same way as the proposed framework. The optimization layer is not a part of the learning model. We consider two traffic prediction modules: LSTM and Bi-LSTM. The Bi-LSTM collects the forward and backward information, and consists of two LSTM layers. The exact values of computational and bandwidth resources are unobservable. We use the mean of these normal distributions instead. Compared to the proposed framework, the predictive optimization framework has lower time complexity, and fails to approximate these unobservable parameters, which can cause inefficiency.
- **Offline optimization framework:** The optimal framework has the same structure as the predictive

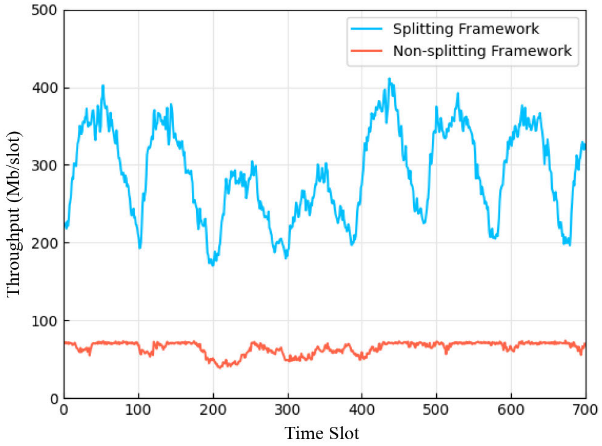

FIGURE 5. **The average throughput achieved by the splitting and non-splitting frameworks.**

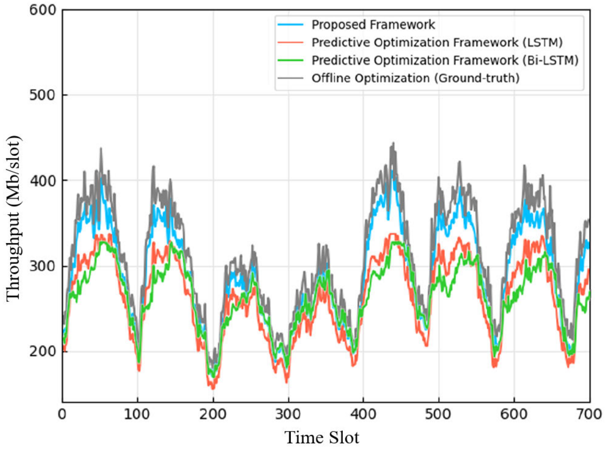

FIGURE 6. **The admitted network traffic volume (i.e., network throughput) achieved by different frameworks.**

optimization framework, with access to all of its available optimal parameters. This framework represents the best-case solution to the network resource allocation problem and provides a benchmark for other frameworks. The traffic variations and resource parameters are known to the offline optimization framework, so that it serves as the upper bound for both the proposed framework and prediction-based frameworks. Compared to the proposed framework, the offline optimization framework outputs the optimal resource allocation in the network (i.e., the target of the network), which can be hard to generate in practice.

B. VALIDATION OF SPLITTING MODEL

We consider the difference between splitting and non-splitting frameworks. As shown in Fig. 5, we compare the average network traffic for over 700 slots on the TOTEM dataset. It is noticed that the splitting framework improves the average throughput substantially, compared to the non-splitting framework, i.e., 289.50 Mb/slot versus 68.81 Mb/slot. The non-splitting framework is limited by the local resources, while the splitting framework develops a cooperative computing network to achieve higher efficiency.

C. ADMITTED NETWORK TRAFFIC VOLUME

Recall in Problem (3) that the network revenue is proportional to the admitted traffic volume $\sum_{k=1}^{K} d_k(t)$. We compare the total traffic flows $\sum_{k=1}^{K} d_k(t)$ achieved by different approaches in Fig. 6. The throughput would be identical to the network revenue when taking $\mathbf{p} = [1] \in R^k$.

After 20 training epochs, all frameworks exhibit periodicity. On average, the proposed framework achieves a total traffic flow of 289.50 Mb/slot over 700 slots, while the predictive framework based on LSTM and Bi-LSTM achieves 261.48 Mb/slot and 256.61 Mb/slot, respectively. The proposed framework outperforms the predictive frameworks, providing an improvement of 9.12% and 11.03% in network throughput.

It is noted in Fig. 6 that the throughput gain of the proposed approach becomes more pronounced at the traffic peak. The approach can achieve up to approximately 10% higher throughput than the conventional prediction-based separate frameworks. This validates the effectiveness of our integrated framework since the traffic prediction module is trained to maximize the network revenue (not minimizing the traffic prediction error in the conventional design).

To validate the discrepancy between the traffic prediction accuracy and resource optimization objective (maximizing throughput), we also compare the predicted and ground-truth network traffic achieved by different frameworks in Fig. 7. Here, the offline curve indicates the ground-truth traffic, while the other three lines represent the predicted traffic in the proposed framework (aimed to maximize throughput) and the predictive frameworks (aimed to minimize prediction errors). The prediction of the proposed objective-driven approach may deviate from the ground-truth traffic, achieving a prediction accuracy of 90.92%. By contrast, the LSTM and Bi-LSTM frameworks achieve a higher prediction accuracy of 98.27% and 97.98%, but degrade throughput.

Based on Figs. 6 and 7, we can conclude that the accuracy of traffic prediction does not fully capture the objective of maximizing network throughput. As a result, the proposed objective-driven integrated approach can achieve approximately considerably throughput (on average) than the conventional prediction-based separate design. Moreover, the maximal throughput gain happens at the traffic peak (where the predicted traffic deviates most from the ground truth).

D. RESULTS ON NETWORK HETEROGENEITY

1) AVERAGE RESOURCE HETEROGENEITY

We consider the heterogeneity of the network by assuming that the computational resources $C_i, \forall i \in Path$ and bandwidth resources $B_{(i,j)}, \forall (i,j) \in Path$ follow normal distributions

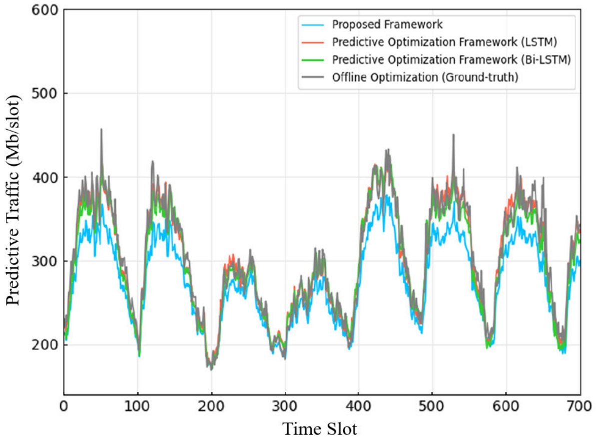

FIGURE 7. The predicted and ground-truth network traffic achieved by different frameworks. The offline provides the ground-truth traffic.

(the few servers/links with negative values are discarded before the deployment of the AI parts). The distributions of computational and bandwidth resources can be described as

$$C_i \sim \mathcal{N}(\mu_C, \sigma_C^2), \quad B_{(i,j)} \sim \mathcal{N}(\mu_B, \sigma_B^2), \tag{13}$$

where μ_C and μ_B represent the means of the resources distributions, and σ_C and σ_B represent the standard deviation of them.

The average resource $M \in R$ can be described as

$$M = \frac{\mu_C + \mu_B}{2}. \tag{14}$$

Fig. 8 illustrates the average throughput of the considered frameworks under different levels of resource availability. The average resource M represents the computational and bandwidth resources of the network. When the average resource is relatively small, the throughput of all frameworks is limited by scarce resources. The proposed framework achieves slightly higher efficiency. As the average resource increases, all frameworks achieve higher throughput. The proposed framework surpasses the benchmarks rapidly. When the average resource exceeds the requirements, the throughput tends to stabilize and the proposed framework shows its improvement. When the average resource is 120 Mb/slot, the proposed framework achieves 292.15 Mb/slot, while the predictive frameworks based on LSTM and Bi-LSTM achieve 264.62 Mb/slot and 257.36 Mb/slot, respectively. This improvement is attributed to the effective approximation of unobservable resources in the differentiable optimization module of the proposed objective-driven structure.

2) RESOURCE DEVIATION HETEROGENEITY

Let $\Sigma \in R$ represent the average of the two standard deviations, σ_C and σ_B; i.e.,

$$\Sigma = \frac{\sigma_C + \sigma_B}{2}. \tag{15}$$

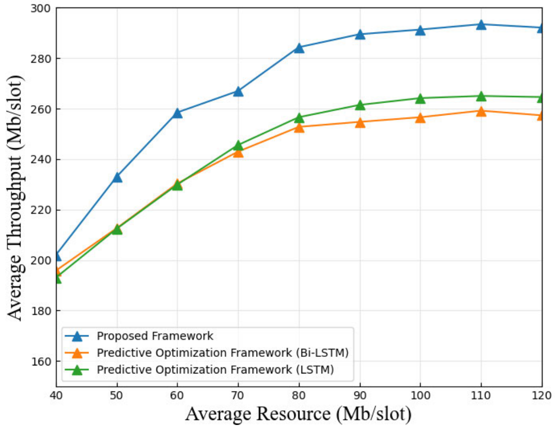

FIGURE 8. The average throughput of different frameworks under different average resource heterogeneity.

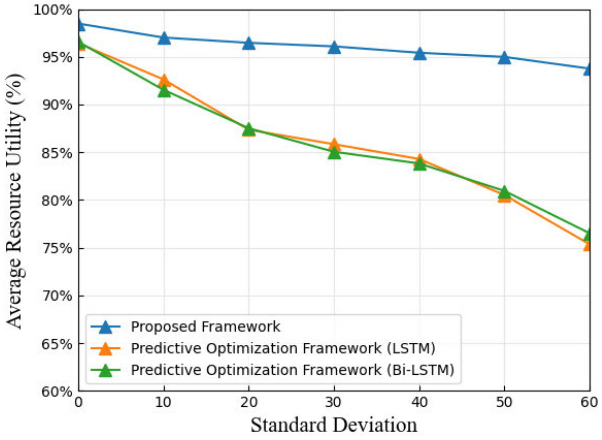

FIGURE 9. The utility (%) of the framework (i.e., the percentage of admitted traffic to the offline optimum) under different resource heterogeneity.

The resource distribution becomes more complex as Σ increases. In the proposed framework, these unobservable resources are iteratively matched through the proposed differentiable optimization module during the backpropagation process. A higher Σ indicates a greater deviation of resources from their mean values. As Σ increases, the task of parameter matching by the methods becomes more challenging.

Fig. 9 illustrates the resource utility outcomes under various standard deviation scenarios. The utility is defined as the percentage of admitted service traffic to the maximum service traffic. The utility helps verify the feasibility of the framework. When Σ is relatively small, the throughput of the proposed framework slightly surpasses that of the benchmarks. As Σ increases, the superiority of the proposed framework is increasingly noticed. This improvement is achieved by effectively approximating unobservable resources, which results in the improved accuracy of resource allocation.

TABLE 4. The summary of resource utility (%), i.e., the percentage of admitted traffic to the offline optimum, on network heterogeneity.

Average standard deviation Σ		0	10	20	30	40	50	60
TOTEM dataset	LSTM framework	96.42	92.63	87.35	85.84	84.30	80.52	75.36
	Bi-LSTM framework	96.58	91.55	87.52	85.05	83.83	80.95	76.51
	Proposed framework	98.49	97.02	96.47	96.09	95.43	94.98	93.76
	The improvement	+2.07	+4.39	+9.12	+10.25	+11.13	+14.46	+18.40
	in the proposed framework	+1.91	+5.47	+8.95	+11.04	+11.60	+14.03	+17.25
Kaggle dataset	LSTM framework	92.13	85.46	79.58	73.42	70.25	66.88	62.45
	Bi-LSTM framework	91.85	85.10	78.03	73.18	70.62	67.14	62.50
	Proposed framework	94.92	93.95	91.65	90.21	88.73	86.16	86.71
	The improvement	+2.79	+8.49	+12.07	+16.79	+18.48	+19.28	+24.26
	in the proposed framework	+3.07	+8.85	+13.62	+17.03	+18.11	+19.02	+24.21

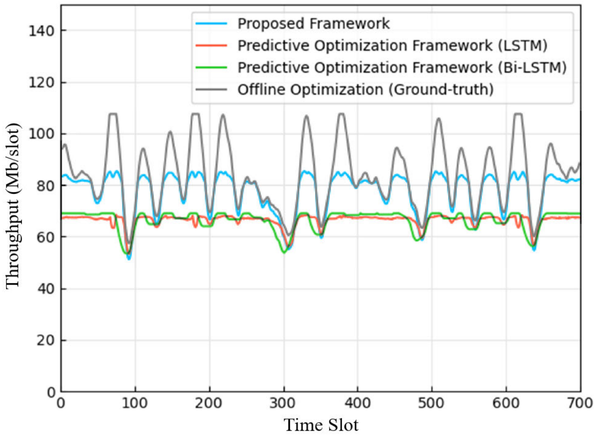

FIGURE 10. The admitted network traffic volume under Kaggle dataset.

3) DATASET HETEROGENEITY FROM KAGGLE

In addition to the TOTEM Project dataset, we have also conducted experiments using an alternative dataset obtained from Kaggle [52]. This Kaggle dataset exhibits a higher level of complexity than the TOTEM Project dataset, incorporating intricate patterns and periodic fluctuations in the traffic data from various servers. This increased complexity poses challenges for prediction-based frameworks, particularly in accurately forecasting traffic for individual time slots.

As shown in Fig. 10, there can be a notable limitation of the conventional predictive optimization framework in handling intricate datasets, leading to high network traffic errors. By contrast, the proposed framework emerges as a viable solution. On average, our proposed framework achieves a traffic rate of 78.60 Mb/slot, while the conventional predictive optimization frameworks based on LSTM and Bi-LSTM achieve 66.92 Mb/slot and 68.25 Mb/slot, respectively. Our framework outperforms the predictive frameworks by about 12% over 700 time slots. This improvement is attributed to the incorporation of the differentiable optimization layer in the proposed framework. This layer mitigates the adverse impact of the LSTM's sub-optimal performance by dynamically updating the parameters to appropriate values, compensating for the limitations of LSTM in handling complex and periodically fluctuating datasets.

4) HETEROGENEITY RESULT SUMMARY

Table 4 presents the network resource utility (i.e., the percentage of admitted traffic to the offline optimum) under different parameter distributions and datasets. The table shows a joint impact of resource heterogeneity and dataset complexity. 1) As M or Σ increases, the differentiable layer in the proposed framework can approximate the network resource parameters via gradients, resulting in a throughput gain or a resource utility gain. 2) A complicated dataset (e.g., Kaggle) would reduce the prediction accuracy of the LSTM model, where the proposed objective-driven framework can show its performance gain due to its robustness and compensate for the prediction errors. The gain can be as significant as up to 38.85% (86.71% versus 62.45%) at the large values of Σ on a dataset with higher complexity.

In summary, the accuracy of traffic prediction (as in the conventional prediction-based separate frameworks) cannot fully capture the objective of maximizing network revenue (i.e., the throughput in the experiments). By training the traffic prediction module to maximize network revenue (instead of minimizing prediction error), the proposed approach can significantly increase the revenue (by up to 38.85%), especially when the network resource heterogeneity (i.e., M and Σ) and traffic uncertainty (complexity of traffic dataset) are prominent.

Moreover, the proposed framework can scale to a large number of edge servers and inter-server links, although the adopted experimental topology originates from the TOTEM dataset. The reason is that the numbers of servers and links only affect the dimensions of the proposed LP problem and does not compromise the stability of the framework. The proposed framework is effective in the face of complex traffic datasets, significant network heterogeneity (i.e., network resources), and different network sizes.

VI. CONCLUSION

This paper has proposed a new, objective-driven, differentiable optimization framework that integrates traffic prediction and resource allocation for split AI inference. To facilitate objective-driven training, the optimization problem (aimed to maximize network revenue while adhering to service and network constraints) has been embedded as the output layer after the traffic prediction module. We have conducted extensive experiments on different traffic datasets, verifying the superiority of the proposed objective-driven framework to existing alternatives in resource utility. The proposed approach is effective in the face of complex traffic datasets and significant network heterogeneity (i.e., different network resources) in which situation the LSTM suffers from the degradation of network throughput. Our approach can achieve up to 38.85% higher network revenue than its alternatives.

The proposed framework integrates traffic prediction and resource allocation, which can increase the efficiency of networks solving LP optimization problems. In the future, we will consider more general optimization problems, which can require more resources and non-trivial extension of the proposed framework. Moreover, an online network dataset will be considered to adapt to more network scenarios.

REFERENCES

[1] *Future Technology Trends of Terrestrial International Mobile Telecommunications Systems Towards 2030 and Beyond*, document ITU-R M.2516-0, 2022.

[2] *Study on Traffic Characteristics and Performance Requirements for AI/ML Model Transfer in 5GS*, 3rd Generation Partnership Project, Sophia Antipolis, France, 2021.

[3] Y. Kang et al., “Neurosurgeon: Collaborative intelligence between the cloud and mobile edge,” *ACM SIGPLAN Notices*, vol. 52, no. 4, pp. 615–629, May 2017.

[4] S. Wang, X. Zhang, H. Uchiyama, and H. Matsuda, “HiveMind: Towards cellular native machine learning model splitting,” *IEEE J. Sel. Areas Commun.*, vol. 40, no. 2, pp. 626–640, Feb. 2022.

[5] Y. Shi et al., “Machine learning for large-scale optimization in 6G wireless networks,” *IEEE Commun. Surveys Tuts.*, vol. 25, no. 4, pp. 2088–2132, 4th Quart., 2023.

[6] X. Lyu, C. Ren, W. Ni, H. Tian, R. P. Liu, and E. Dutkiewicz, “Optimal online data partitioning for geo-distributed machine learning in edge of wireless networks,” *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2393–2406, Oct. 2019.

[7] X. Lyu, J. Liu, C. Ren, and G. Nan, “Security-communication-computation tradeoff of split decisions for edge intelligence,” *IEEE Wireless Commun.*, vol. 31, no. 3, pp. 257–263, Mar. 2023.

[8] G.-L. Pereteanu, A. Alansary, and J. Passerat-Palmbach, “Split HE: Fast secure inference combining split learning and homomorphic encryption,” 2022, *arXiv:2202.13351*.

[9] Q. Lan, Q. Zeng, P. Popovski, D. Gündüz, and K. Huang, “Progressive feature transmission for split inference at the wireless edge,” 2021, *arXiv:2112.07244*.

[10] A. E. Eshratifar, M. S. Abrishami, and M. Pedram, “JointDNN: An efficient training and inference engine for intelligent mobile cloud computing services,” *IEEE Trans. Mobile Comput.*, vol. 20, no. 2, pp. 565–576, Feb. 2021.

[11] T. Feltin, L. Marché, J.-A. Cordero-Fuertes, F. Brockners, and T. H. Clausen, “DNN partitioning for inference throughput acceleration at the edge,” *IEEE Access*, vol. 11, pp. 52236–52249, 2023.

[12] D. Luo, T. Yu, Y. Wu, H. Wu, T. Wang, and W. Zhang, “SPLIT: QoS-aware DNN inference on shared GPU via evenly-sized model splitting,” in *Proc. 52nd Int. Conf. Parallel Process.*, vol. 1, Aug. 2023, pp. 605–614.

[13] K.-J. Fu, Y.-T. Yang, and H.-Y. Wei, “Split computing video analytics performance enhancement with auction-based resource management,” *IEEE Access*, vol. 10, pp. 106495–106505, 2022.

[14] W. Wu et al., “Split learning over wireless networks: Parallel design and resource management,” *IEEE J. Sel. Areas Commun.*, vol. 41, no. 4, pp. 1051–1066, Apr. 2023.

[15] Y. Pointurier, N. Benzaoui, W. Lautenschlaeger, and L. Dembeck, “End-to-end time-sensitive optical networking: Challenges and solutions,” *J. Lightw. Technol.*, vol. 37, no. 7, pp. 1732–1741, Apr. 15, 2019.

[16] Z. Jia, Q. Liu, Y. He, Q. Wu, R. P. Liu, and Y. Sun, “Efficient end-to-end failure probing matrix construction in data center networks,” *J. Commun. Netw.*, vol. 25, no. 4, pp. 532–543, Aug. 2023.

[17] R. Wang, Y. Zhang, L. Peng, G. Fortino, and P.-H. Ho, “Time-varying-aware network traffic prediction via deep learning in IIoT,” *IEEE Trans. Ind. Informat.*, vol. 18, no. 11, pp. 8129–8137, Nov. 2022.

[18] B. Amos and J. Z. Kolter, “OptNet: Differentiable optimization as a layer in neural networks,” in *Proc. Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 136–145.

[19] A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, and J. Z. Kolter, “Differentiable convex optimization layers,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–13. [Online]. Available: https://proceedings.neurips.cc/paper/2019/hash/9ce3c52fc54362e22053399d3181c638-Abstract.html

[20] Q. Tang et al., “Collective deep reinforcement learning for intelligence sharing in the Internet of Intelligence-empowered edge computing,” *IEEE Trans. Mobile Comput.*, vol. 22, no. 11, pp. 6327–6342, Nov. 2023.

[21] D. Zeng, L. Gu, S. Pan, J. Cai, and S. Guo, “Resource management at the network edge: A deep reinforcement learning approach,” *IEEE Netw.*, vol. 33, no. 3, pp. 26–33, May 2019.

[22] D. Wang, B. Song, P. Lin, F. R. Yu, X. Du, and M. Guizani, “Resource management for edge intelligence (EI)-assisted IoV using quantum-inspired reinforcement learning,” *IEEE Internet Things J.*, vol. 9, no. 14, pp. 12588–12600, Jul. 2022.

[23] Y. Tu, H. Chen, L. Yan, and X. Zhou, “Task offloading based on LSTM prediction and deep reinforcement learning for efficient edge computing in IoT,” *Future Internet*, vol. 14, no. 2, p. 30, Jan. 2022.

[24] X. Xiong, K. Zheng, L. Lei, and L. Hou, “Resource allocation based on deep reinforcement learning in IoT edge computing,” *IEEE J. Sel. Areas Commun.*, vol. 38, no. 6, pp. 1133–1146, Jun. 2020.

[25] Y. Zhu and S. Wang, “Joint traffic prediction and base station sleeping for energy saving in cellular networks,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2021, pp. 1–6.

[26] J. Li, J. Wen, and M. Sheng, “Intelligent power control algorithm in heterogeneous wireless cellular networks,” *IEEE Trans. Veh. Technol.*, vol. 70, no. 9, pp. 8823–8837, Sep. 2021.

[27] S. Manzoor, A. N. Mian, and S. Mazhar, “An LSTM-based cell association scheme for proactive bandwidth management in 5G fog radio access networks,” *Int. J. Commun. Syst.*, vol. 34, no. 15, p. e4943, Oct. 2021.

[28] M. Chen, Y. Miao, H. Gharavi, L. Hu, and I. Humar, “Intelligent traffic adaptive resource allocation for edge computing-based 5G networks,” *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 2, pp. 499–508, Jun. 2020.

[29] U. Fattore, M. Liebsch, B. Brik, and A. Ksentini, “AutoMEC: LSTM-based user mobility prediction for service management in distributed MEC resources,” in *Proc. 23rd Int. ACM Conf. Modelling Anal. Simulation Wireless Mobile Syst. (MSWiM)*, 2020, pp. 155–159.

[30] Y. Cui, X. Huang, D. Wu, and H. Zheng, “Machine learning-based resource allocation strategy for network slicing in vehicular networks,” *Wireless Commun. Mobile Comput.*, vol. 2020, pp. 1–10, Nov. 2020.

[31] F. Kavehmadavani, V.-D. Nguyen, T. X. Vu, and S. Chatzinotas, “Intelligent traffic steering in beyond 5G open RAN based on LSTM traffic prediction,” *IEEE Trans. Wireless Commun.*, vol. 22, no. 11, pp. 7727–7742, Nov. 2023.

[32] J. Kotary, F. Fioretto, P. Van Hentenryck, and B. Wilder, “End-to-end constrained optimization learning: A survey,” 2021, *arXiv:2103.16378*.

[33] K. Wang, B. Wilder, A. Perrault, and M. Tambe, “Automatically learning compact quality-aware surrogates for optimization problems,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 9586–9596.

[34] L. Kong, J. Cui, Y. Zhuang, R. Feng, B. A. Prakash, and C. Zhang, “End-to-end stochastic optimization with energy-based model,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 11341–11354.

[35] A. Zanfir and C. Sminchisescu, "Deep learning of graph matching," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2684–2693.

[36] P. Jin, X. Fei, Q. Zhang, F. Liu, and B. Li, "Latency-aware VNF chain deployment with efficient resource reuse at network edge," in *Proc. IEEE Conf. Comput. Commun.*, Oct. 2020, pp. 267–276.

[37] X. Lyu, C. Ren, W. Ni, H. Tian, Q. Cui, and R. P. Liu, "Online learning of optimal proactive schedule based on outdated knowledge for energy harvesting powered Internet-of-Things," *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 1248–1262, Feb. 2021.

[38] X. Lyu, C. Ren, W. Ni, H. Tian, R. P. Liu, and X. Tao, "Distributed online learning of cooperative caching in edge cloud," *IEEE Trans. Mobile Comput.*, vol. 20, no. 8, pp. 2550–2562, Aug. 2021.

[39] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural Comput.*, vol. 31, no. 7, pp. 1235–1270, Jul. 2019.

[40] C. Olah. (2015). *Understanding LSTM Networks*. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs

[41] H. D. Nguyen, K. P. Tran, S. Thomassey, and M. Hamad, "Forecasting and anomaly detection approaches using LSTM and LSTM autoencoder techniques with the applications in supply chain management," *Int. J. Inf. Manage.*, vol. 57, Apr. 2021, Art. no. 102282.

[42] Y. Xie, R. Liang, Z. Liang, C. Huang, C. Zou, and B. Schuller, "Speech emotion classification using attention-based LSTM," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 27, no. 11, pp. 1675–1685, Nov. 2019.

[43] Q. Kang, E. J. Chen, Z.-C. Li, H.-B. Luo, and Y. Liu, "Attention-based LSTM predictive model for the attitude and position of shield machine in tunneling," *Underground Space*, vol. 13, pp. 335–350, Dec. 2023.

[44] Y. Pan, T. Yao, H. Li, and T. Mei, "Video captioning with transferred semantic attributes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6504–6512.

[45] S. Balon and G. Monfort. *Totem Project*. Accessed: Jun. 14, 2023. [Online]. Available: https://totem.run.montefiore.uliege.be/datatools.html

[46] R. Andrade, A. Lisser, N. Maculan, and G. Plateau, "Telecommunication network capacity design for uncertain demand," *Comput. Optim. Appl.*, vol. 29, no. 2, pp. 127–146, Nov. 2004.

[47] S. He, L. Guo, M. Ghanem, and Y. Guo, "Improving resource utilisation in the cloud environment using multivariate probabilistic models," in *Proc. IEEE 5th Int. Conf. Cloud Comput.*, Jun. 2012, pp. 574–581.

[48] E. M. Heien, D. Kondo, and D. P. Anderson, "Correlated resource models of internet end hosts," in *Proc. 31st Int. Conf. Distrib. Comput. Syst.*, Jun. 2011, pp. 278–287.

[49] X. Tian, P. Xu, H. Gu, and H. Meng, "Energy-efficient DNN partitioning and offloading for task completion rate maximization in multiuser edge intelligence," *Wireless Commun. Mobile Comput.*, vol. 2023, pp. 1–14, Feb. 2023.

[50] Y. Z. Yaping Zheng, "Research on cloud-edge joint task inference algorithm in edge intelligence," *J. Comput.*, vol. 32, no. 4, pp. 211–224, Aug. 2021.

[51] Q. Zhang, F. Liu, and C. Zeng, "Online adaptive interference-aware VNF deployment and migration for 5G network slice," *IEEE/ACM Trans. Netw.*, vol. 29, no. 5, pp. 2115–2128, Oct. 2021.

[52] C. Crawford. *Computer Network Traffic*. Accessed: Jul. 6, 2023. [Online]. Available: https://www.kaggle.com/datasets

XINCHEN LYU received the B.E. degree from Beijing University of Posts and Telecommunications (BUPT) in 2014 and the dual Ph.D. degree from BUPT and the University of Technology Sydney in 2019. He is currently an Associate Professor with the National Engineering Research Center for Mobile Network Technologies, BUPT. His research interests include the resource management and security of edge intelligence and its applications in future wireless networks.

YUEWEI LI received the B.E. degree from Beijing University of Posts and Telecommunications (BUPT) in 2022, where he is currently pursuing the master's degree with the School of Cyberspace Security. His research interests include the edge intelligence and convex optimization.

YING HE (Senior Member, IEEE) received the B.E. degree in telecommunications engineering from Beijing University of Posts and Telecommunications, China, in 2009, and the Ph.D. degree in telecommunications engineering from the University of Technology Sydney, Australia, in 2017. She is currently a Senior Lecturer with the School of Electrical and Data Engineering, University of Technology Sydney. Her research interests are physical layer algorithms in wireless communication, with machine learning, vehicular communication, spectrum sharing, and satellite communication.

CHENSHAN REN received the B.E. degree from Zhengzhou University, Henan, China, in 2013, and the dual Ph.D. degree from Beijing University of Posts and Telecommunications and the University of Technology Sydney in 2019. She is currently a Lecturer with the Minzu University of China. Her research interests include fog computing, software-defined networking, and radio resource management.

WEI NI (Fellow, IEEE) received the B.E. and Ph.D. degrees in communication science and engineering from Fudan University, Shanghai, China, in 2000 and 2005, respectively. He is an Adjunct Professor with the University of Technology Sydney and an Honorary Professor with Macquarie University. He was a Post-Doctoral Fellow with Shanghai Jiao Tong University, from 2005 to 2008; the Deputy Project Manager with Bell Labs, Alcatel/Alcatel-Lucent, from 2005 to 2008; and a Senior Researcher with Nokia, from 2008 to 2009. His research interests include machine learning, online learning, stochastic optimization, and their applications to the security, integrity, and efficiency of network systems.

REN PING LIU (Senior Member, IEEE) received the B.E. degree in telecommunication engineering and the M.E. degree in computer engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 1985 and 1988, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Newcastle, Callaghan, NSW, Australia, in 1996.

He is a Professor and the Head of Discipline of Network and Cybersecurity, University of Technology Sydney, Ultimo, NSW. As a Research Leader, a Certified Network Professional, and a Full Stack Web Developer, he has delivered networking and cybersecurity solutions to government agencies and industry customers. He has supervised over 30 Ph.D. students and has over 200 research publications. His research interests include wireless networking, 5G, the IoT, vehicular networks, 6G, cybersecurity, and blockchain.

Prof. Liu was the winner of NSW iAwards 2020 for leading the Blockchain enabled Fish provenance And Quality Tracking (BeFAQT) Project. He received the Australian Engineering Innovation Award 2012 and the CSIRO Chairman’s Medal for his contribution in the Wireless Backhaul Project. He was the Founding Chair of IEEE NSW VTS Chapter.

PENGCHENG ZHU (Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from Shandong University, Jinan, China, in 2001 and 2004, respectively, and the Ph.D. degree in communication and information science from Southeast University, Nanjing, China, in 2009. He is currently a Professor with the National Mobile Communications Research Laboratory, Southeast University. His research interests include wireless communications and mobile networks, including 5G/6G mobile communication systems, massive MIMO, ultra-reliable and low latency communications (URLLC), and mmWave communications.

QIMEI CUI (Senior Member, IEEE) received the B.E. and M.S. degrees in electronic engineering from Hunan University, Changsha, China, in 2000 and 2003, respectively, and the Ph.D. degree in information and communications engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 2006. She has been a Full Professor with the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, since 2014. She was a Visiting Professor with the Department of Electronic Engineering, University of Notre Dame, Notre Dame, IN, USA, in 2016. Her research interests include B5G/6G wireless communications, mobile computing, and the IoT. She won the Best Paper Award at IEEE ISCIT 2012, IEEE WCNC 2014, and WCSP 2019; the Honorable Mention Demo Award at ACM MobiCom 2009; and the Young Scientist Award at URSI GASS 2014. She serves as the Technical Program Chair for APCC 2018, the Track Chair for IEEE/CIC ICCC 2018, and the Workshop Chair for WPMC 2016. She also serves as a Technical Program Committee Member for several international conferences, such as IEEE ICC, IEEE WCNC, IEEE PIMRC, IEEE ICCC, WCSP 2013, and IEEE ISCIT 2012. She serves as an Editor for *Science China Information Science*; and a Guest Editor for *EURASIP Journal on Wireless Communications and Networking*, *International Journal of Distributed Sensor Networks*, and *Journal of Computer Networks and Communications*.