Engineering 54 (2025) 127–142

Contents lists available at ScienceDirect

Engineering

journal homepage: www.elsevier.com/locate/eng

Research
Theory and Key Technologies in Satellite Internet Networking—Article

Dynamic Time-Difference QoS Guarantee in Satellite–Terrestrial Integrated Networks: An Online Learning-Based Resource Scheduling Scheme

Xiaohan Qin [a], Tianqi Zhang [a], Kai Yu [a], Xin Zhang [a], Haibo Zhou [a,*], Weihua Zhuang [b], Xuemin Shen [b,*]

[a] School of Electronic Science and Engineering, Nanjing University, Nanjing 210023, China
[b] Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada

ARTICLE INFO

Article history:
Received 6 December 2024
Revised 16 July 2025
Accepted 17 September 2025
Available online 15 October 2025

Keywords:
Satellite–terrestrial integrated networks
Dynamic resource scheduling
Conformal prediction
Online convex optimization

ABSTRACT

The rapid growth of low-Earth-orbit satellites has injected new vitality into future service provisioning. However, given the inherent volatility of network traffic, ensuring differentiated quality of service in highly dynamic networks remains a significant challenge. In this paper, we propose an online learning-based resource scheduling scheme for satellite–terrestrial integrated networks (STINs) aimed at providing on-demand services with minimal resource utilization. Specifically, we focus on: ① accurately characterizing the STIN channel, ② predicting resource demand with uncertainty guarantees, and ③ implementing mixed timescale resource scheduling. For the STIN channel, we adopt the 3rd Generation Partnership Project channel and antenna models for non-terrestrial networks. We employ a one-dimensional convolution and attention-assisted long short-term memory architecture for average demand prediction, while introducing conformal prediction to mitigate uncertainties arising from burst traffic. Additionally, we develop a dual-timescale optimization framework that includes resource reservation on a larger timescale and resource adjustment on a smaller timescale. We also designed an online resource scheduling algorithm based on online convex optimization to guarantee long-term performance with limited knowledge of time-varying network information. Based on the Network Simulator 3 implementation of the STIN channel under our high-fidelity satellite Internet simulation platform, numerical results using a real-world dataset demonstrate the accuracy and efficiency of the prediction algorithms and online resource scheduling scheme.

1. Introduction

In recent years, low-Earth-orbit (LEO) satellite–terrestrial integrated networks (STINs) have gained significant attention and rapid development, emerging as a key enabling technology for sixth-generation (6G) mobile communication networks [1,2]. With their extensive coverage and dense deployment, LEO satellites offer global coverage and seamless connectivity, even in remote regions without terrestrial network infrastructure [3,4]. Furthermore, satellite networks play a crucial role in easing the workload of terrestrial networks [5,6], providing on-demand services tailored to a wide range of application scenarios.

The diverse nature of service requirements introduces considerable variation in quality of service (QoS) demands [7]. For example, delay-sensitive (DS) applications requiring real-time interaction depend on low latency and high reliability, while data-intensive services typically require greater capacity but can tolerate higher latency. Resource scheduling involves multiple stakeholders: network operators (NOs), service providers (SPs), and users. Each SP represents a specific service type, assessing its users' resource needs and requesting allocation from the NO. However, fluctuations in traffic load often lead to reduced resource efficiency and hinder QoS provisioning for SPs [8]. Additionally, due to the rapid orbital motion of LEO satellites, network resources experience highly dynamic fluctuations within the STIN [9,10]. For instance, the bandwidth of downlink feeder links can fluctuate by up to 100% within just 150 s. A critical challenge, therefore, is to develop adaptive and intelligent resource scheduling mechanisms that

* Corresponding authors.
E-mail addresses: haibozhou@nju.edu.cn (H. Zhou), sshen@uwaterloo.ca (X. Shen).

https://doi.org/10.1016/j.eng.2025.09.025

effectively allocate limited resources among SPs with varying QoS requirements, considering the fluctuations in traffic and network dynamics.

The existing research on resource scheduling primarily focuses on terrestrial mobile networks that effectively adapt to dynamic service demands through traffic prediction. Although terrestrial networks encounter stochastic service demands and dynamic resource availability, the unique characteristics of STIN significantly increase the complexity of resource scheduling. In terrestrial networks, the primary network dynamics result from user mobility, causing relatively slow (minute-level) topological changes. In contrast, the rapid movement of LEO satellites in STIN leads to much faster (second-level) topological changes, greatly intensifying fluctuations in link resources [11,12]. This increased dynamism requires real-time, accurate modeling of network dynamics to capture and reflect these rapid state transitions effectively. Regarding service demand prediction, precise forecasting is crucial in compensating for the high-latency feedback loops and limited onboard resources typical of STINs [8,13]. Additionally, as channel conditions are highly sensitive to environmental factors, maintaining a degree of redundancy is necessary to account for uncertainties during resource provisioning. In terms of on-demand matching, which faces the dual dynamics of the network environment and service requirements, resource scheduling requires both performance isolation and timely adjustment capabilities, while ensuring the long-term effectiveness of the algorithm, even in the case of incomplete or lagging information.

This study addresses three critical challenges: ① accurately characterizing dynamic satellite–terrestrial link resources, ② precisely predicting service demands while accounting for uncertainties, and ③ achieving on-demand matching between dynamic network resources and stochastic traffic fluctuations. To tackle these issues, we propose an online learning-based resource scheduling framework for STINs, designed to provide dynamic QoS guarantees with minimal resource consumption under highly dynamic network conditions and uncertain service demands. For satellite–terrestrial link modeling, we developed channel and antenna models compliant with the 3rd Generation Partnership Project (3GPP) non-terrestrial network (NTN) standards, implemented within a discrete event simulation (DES)-based STIN simulator. This accurately reflects the real transmission characteristics of STINs. For service demand prediction, we propose a one-dimensional convolution (1D-Conv) and attention-assisted long short-term memory (CA-LSTM) architecture. The 1D-Conv layer is used for local feature extraction, long short-term memory (LSTM) captures temporal dependencies, and attention aggregates weighted features, resulting in superior performance compared to single-model approaches. To enhance prediction reliability, we introduce conformal prediction (CP), which is specifically adapted to handle sequential uncertainties in service demand. CP offers the significant advantage of being distribution-free and model-free.

Building on this foundation, we designed a dual-timescale resource scheduling optimization framework to facilitate on-demand resource matching. Resource reservation occurs on a larger timescale based on service demand predictions, ensuring differentiated service isolation, whereas resource adjustments are performed on a smaller timescale in response to dynamic network resources and fluctuating traffic, thereby ensuring efficient resource utilization and QoS. The main contributions of this study are summarized as follows:

- We propose a 1D-Conv and CA-LSTM algorithm for high-accuracy multidimensional service demand prediction. Additionally, we developed an adaptive sequential conformal prediction (ASCP) algorithm tailored for time-series data. This algorithm effectively manages uncertainties caused by bursty traffic and provides statistically guaranteed prediction intervals at user-defined confidence levels.
- We designed a dual-timescale optimization framework that incorporates resource reservation at a larger timescale and resource adjustments at a smaller timescale, aiming to minimize resource occupation while maintaining QoS constraints. Using online convex optimization (OCO) theory, we propose an online resource scheduling (ORS) algorithm that guarantees long-term performance despite limited network information.
- We implemented channel and antenna models based on the 3GPP NTN standard in our high-fidelity STIN simulation platform, which simulates satellite orbital movements to capture dynamic satellite-to-ground link conditions. Numerical experiments using real-world traffic datasets validate the effectiveness of the proposed prediction algorithm. Furthermore, the prediction-assisted resource-scheduling scheme demonstrated a performance comparable to that of an optimal oracle with complete future system state information.

The remainder of this paper is organized as follows: Related studies are introduced in Section 2. In Section 3, we describe the system model and formulate a dual-timescale optimization problem. Traffic prediction and resource scheduling modules are proposed in Sections 4 and 5, respectively. Simulation results are presented in Section 6. Finally, Section 7 concludes the study.

2. Related works

Service-demand prediction is a critical component of resource scheduling. By analyzing historical data and variations in trends, future resource demands can be forecasted, providing vital information and decision-making support for resource scheduling schemes.

2.1. Service demand prediction

The existing service demand prediction methods can be broadly classified into model-based approaches [14,15] and data-driven [16–18] approaches. Model-based methods rely on a comprehensive understanding and precise modeling of the underlying rules that govern data generation. For instance, Cai et al. [14] introduced a dynamic radio access network (RAN) slicing model that incorporates three distinct service types, each with multiple service distributions, to accommodate diverse user request patterns and traffic characteristics within the same slice. Similarly, Tu et al. [15] explored the RAN slicing problem using a separate architecture for control and user planes and proposed a Lyapunov-based deep reinforcement learning (DRL) framework to optimize solutions. This method assumes that service data arrival times are independent and identically distributed (IID), following a Poisson distribution. However, implementing these model-based approaches practically is challenging, particularly when the underlying mechanisms are overly complex or uncertain. By contrast, data-driven methods primarily utilize deep-learning algorithms to uncover patterns and relationships from existing data without explicitly adhering to specific model structures. For example, Cui et al. [16] proposed a deep-learning algorithm to capture the long-term features of QoS requests in dynamic vehicular environments and train a traffic prediction network using real cellular traffic data from Milan. Li et al. [17] designed a framework to guarantee service-level agreements in network slicing by employing an LSTM-based resource demand predictor trained on data collected by Telecom Italia to facilitate cross-slice resource reservation and admission control. Additionally, Jiang et al. [18] introduced a hybrid multimodal framework that combines convolutional neural networks

(CNNs) and graph neural networks (GNNs) for single-step mobile traffic prediction. This framework incorporates a convolutional LSTM within the CNN module and includes a dedicated fusion layer designed to effectively merge the outputs from both the CNN and GNN modules.

Most previous traffic prediction methods typically produced only single-point predictions, which inadequately addressed the volatility of actual traffic demand influenced by multiple factors. As a result, prediction errors were common. Moreover, few studies have examined how these prediction errors propagate into resource scheduling decisions, often leading to suboptimal resource provisioning in real-world environments. These limitations underscore the need to incorporate error-sensing and analysis mechanisms into traffic prediction models to improve prediction robustness and scheduling efficiency. Many studies have adopted classical Bayesian learning approaches to quantify prediction uncertainty [19–21]. For example, Lin and Li [19] introduced a Bayesian deep learning framework featuring uncertainty quantification and calibration for predicting the remaining useful life (RUL). This framework integrates epistemic uncertainty (reflecting model ignorance) and aleatoric uncertainty (representing inherent observation noise) and proposes an iterative calibration method combining isotonic regression with standard deviation scaling. Similarly, He et al. [20] presented a digital twin-assisted robust adaptive resource slicing approach that merged LSTM networks with Bayesian neural networks (BNNs) to handle uncertainties in service demand predictions. Moreover, Sachdeva et al. [21] developed a robust *Q*-value estimation technique for autonomous vehicle applications by combining a BNN with skewed geometric Jensen–Shannon divergence to mitigate the uncertainties from inherent noise. However, Bayesian-based methods provide formal calibration guarantees only under the assumption that the model of the ground-truth data generation mechanism is well specified, which is easily violated in practice. Meanwhile, error-aware prediction frameworks have gained increasing attention for modeling prediction errors and dynamically correcting uncertainties [22,23]. For example, Aboeleneen et al. [22] proposed a two-phase error-aware proactive network slicing framework. The initial phase employs a deep learning-based model for accurate load forecasting, followed by a second phase that leverages a DRL agent to adjust the prediction errors based on key performance indicator requirements. Additionally, Garrido et al. [23] integrated domain knowledge related to fifth-generation (5G) network infrastructure, resource demand, and service performance into a machine learning-based predictor. However, such approaches generally require domain-specific knowledge or complex correction mechanisms, which lack rigorous statistical calibration guarantees and encounter difficulties when adapting to dynamic or diverse application scenarios.

In recent years, CP, a nonparametric method with minimal assumptions, has gained attention because of its ability to provide statistically valid calibration guarantees for predictive outcomes. Owing to its model- and distribution-free characteristics, CP can be seamlessly integrated with any predictive model, demonstrating notable flexibility and adaptability. For instance, Park et al. [24] proposed a meta-learning algorithm based on cross-validated CP, employing adaptive nonconformity scores to enhance input condition calibration. This method effectively reduced the average prediction set size while maintaining formal task-specific calibration guarantees. Similarly, Cohen et al. [25] explored the application of CP in artificial intelligence (AI)-driven communication system design. Offline CP methods have been used in demodulation and modulation classification tasks to provide theoretically sound calibrated decisions. Additionally, an online CP method was developed to achieve predefined long-term target coverage rates with minimal inflation of prediction intervals for aerial measurement scenarios. Furthermore, Piao et al. [26] combined deep learning-based RUL prediction models with uncertainty-aware conformal quantile regression to construct a credible RUL estimation framework, extending conformal quantile regression from symmetric to asymmetric variants, supported by rigorous theoretical assurances. Nevertheless, CP methods assume data exchangeability, which implies that observations should be statistically interchangeable across different time points. It is challenging to satisfy this assumption when dealing with sequential data that exhibit a clear temporal dependence. Violations of this condition may invalidate the theoretical guarantees of traditional CP methods for time-series predictions. Consequently, it is crucial to enhance and extend CP frameworks by incorporating temporal dependency structures or sequential ordering information to address the inherent nonstationarity and autocorrelation present in time-series data.

2.2. Network resource scheduling

Leveraging service demand predictions allows resource scheduling to adapt more effectively to dynamic service requirements, thereby enhancing resource utilization and ensuring reliable service delivery in complex environments. For example, Lai et al. [27] introduced a multi-tier edge computing architecture designed to address load imbalances by utilizing inter-satellite links for collaborative offloading. This architecture integrates a multi-agent DRL optimization framework, solving the computation offloading subproblem with multi-agent deep *Q*-networks and addressing the resource allocation subproblem using a deep deterministic policy gradient model. Additionally, Jiang et al. [28] proposed an enhanced fuzzy neural network (FNN) model optimized using the African Vulture optimization algorithm, specifically tailored for access selection in STINs. Similarly, Wang et al. [29] developed a resource allocation algorithm for delay-tolerant (DT) services based on traffic prediction by employing the dueling double deep *Q*-network (D3QN) algorithm to allocate physical resource blocks (RBs) among different network slices, complemented by a heuristic approach for RB allocation among nodes within each slice. Furthermore, Hou et al. [30] proposed a slice-based RB leasing and an association adjustment scheme. In this approach, the prediction module employs LSTM networks, whereas the leasing module optimizes the RB renting and borrowing through a coalition exchange. Subsequently, the association adjustment module determines precise RB adjustment strategies using the potential game theory, ensuring effective interference isolation.

Resource scheduling typically involves balancing long-term planning with real-time responsiveness to meet varying requirements across different temporal scales and resource granularities. For instance, Huang et al. [31] aimed to optimize network computational performance and ensure QoS for tasks in STIN by designing a DRL framework that handles task scheduling on smaller timescales and resource slicing on larger timescales. Similarly, Liu et al. [32] introduced a reconfigurable satellite network slicing architecture designed to maximize resource utilization and service satisfaction. This was achieved through a two-tier DRL algorithm, where the first layer managed resource scheduling and the second layer handled user association decisions. The existing resource scheduling methods often rely on conventional optimization techniques (e.g., convex optimization and game theory) or reinforcement learning solutions (e.g., DRL). Traditional convex optimization methods typically assume complete knowledge of network state information (e.g., service demands) in advance, which is often unrealistic in rapidly evolving satellite network environments. While DRL methods excel in adapting to dynamic situations, they typically require extensive interactions with the environment to converge, leading to high computational resource

demands and increased communication overheads. This limits their practical application in resource-constrained satellite scenarios. OCO has emerged as a promising dynamic optimization framework, providing a novel approach to address these challenges. OCO algorithms optimize objective functions progressively through real-time decision updates under partial observability and dynamic conditions, without the need for full network state information. They offer advantages such as low computational overhead, strong adaptability, and theoretical reliability. For example, Wang et al. [33] proposed an OCO algorithm that utilizes periodic queuing and multistep gradient aggregation for online precoding design in large-scale multi-antenna systems. Choi et al. [34] explored joint optimization of dynamic learning and resource scaling for mobile vision applications within an OCO framework, enabling context–aware adaptation in uncertain environments. Additionally, Chouayakh and Destounis [35] developed an OCO-based online algorithm for edge resource reservation, introducing a novel metric, the degree of interruption compliance, to mitigate service disruptions. However, these studies predominantly address single-timescale optimization problems and do not fully exploit service demand predictions, leaving significant opportunities to enhance both resource utilization efficiency and QoS guarantee performance.

3. System model

In this section, we first introduce the network and communication models of the STIN, and then formulate the resource scheduling problem based on these models.

3.1. Network model

We investigated satellite-to-ground downlink transmission in STIN. Each LEO satellite provides various services to users within its target area, defined as the coverage of a single satellite beam. LEO satellites are equipped with steerable antennas that can continuously point toward a target area under elevation angle constraints. As shown in Fig. 1, we focus on two typical network services with different QoS requirements: the DS service $\mathscr{M}_{\mathrm{DS}}$ and the DT service $\mathscr{M}_{\mathrm{DT}}$. Generally, DT services aim to provide high-data-rate transmission, such as file transfer protocol (FTP), voice over Internet protocol (VoIP), and video streaming. In contrast, DS services support scenarios that require low latency and high reliability, characterized by randomness and suddenness, such as emergency communication and engineering monitoring.

We define the STIN to run in the time-slotted mode, where the time horizon is first divided into several equal time windows $\mathscr{T} = \{1, 2, ..., T\}$ (where T is the total number of the time windows) and each time window is further subdivided into K time slots, denoted as $\mathscr{K} = \{1, 2, ..., K\}$ (where K is the total number of the time slots). For simplicity, (t, k) represents the kth time slot within the tth time window, where $t = 1, 2, ..., T$ and $k = 1, 2, ..., K$. Because the duration of each time slot is relatively short (typically a few seconds), we assume that the position of the LEO satellite remains fixed within each time slot and varies over the time slots.

3.2. Communication model

Consider a time-varying satellite-to-ground downlink channel, where the channel state changes over time with satellite movement. Let $\mathscr{U} = \mathscr{M}_{\mathrm{DT}} \cup \mathscr{M}_{\mathrm{DS}}$ denote all user requests for the two services. In time (t, k), the channel state between LEO satellite $s \in \mathscr{S}$ and users $u \in \mathscr{U}$ can be modeled as follows, where $\mathscr{S}$ and $\mathscr{U}$ represent the LEO satellite set and the user set, respectively.

$$h_{s,u}(t,k) = G_s^{\mathrm{Tr}} \cdot G_{s,u}^{\mathrm{Ch}}(t,k) \cdot G_u^{\mathrm{Rv}} \tag{1}$$

where $h_{s,u}$ is the channel state, $G_{s,u}^{\mathrm{Ch}}$ represents the satellite–terrestrial channel between LEO satellite s and user u, G_s^{Tr} represents the transmitting antenna gain of the LEO satellite s, and G_u^{Rv} represents the receiving antenna gain of the user u. According to 3GPP TR 38.811 [36], both satellites and ground terminals commonly use a circular aperture antenna model. The normalized antenna gain pattern (G) is given by

$$G(\theta) = \begin{cases} 1 & \theta = 0 \\ 4\left|\frac{J_1(\gamma \cdot \ell \cdot \sin\theta)}{\gamma \cdot \ell \cdot \sin\theta}\right|^2 & 0 < |\theta| \le 90^\circ \end{cases} \tag{2}$$

where $J_1(\cdot)$ is the first order Bessel function, ℓ represents the radius of the antenna's circular aperture, and θ is the angle measured from the bore sight of the antenna's main beam. The value of γ can be calculated as $\gamma = \frac{2\pi f_s}{c_1}$, where f_s is the carrier frequency of the LEO satellite s and c_1 is the speed of light in vacuum.

The $G_{s,u}^{\mathrm{Ch}}$ channel model is also specified as Ref. [36], including path loss $G_{s,u}^{\mathrm{PL}}$, atmospheric absorption $G_{s,u}^{\mathrm{AA}}$, scintillation $G_{s,u}^{\mathrm{SS}}$, and fast fading $G_{s,u}^{\mathrm{FF}}$, which can be written as (in dB)

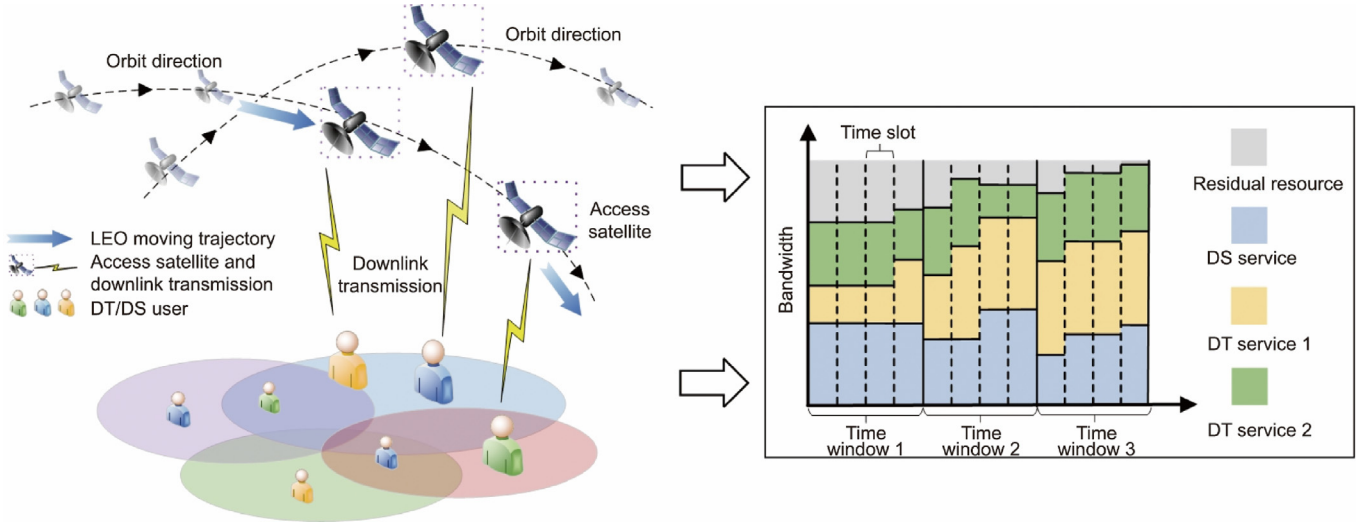

Fig. 1. Resource scheduling scenario in STIN.

$$G_{s,u}^{\mathrm{Ch}}(t,k)_{\mathrm{dB}} = G_{s,u}^{\mathrm{PL}} + G_{s,u}^{\mathrm{AA}} + G_{s,u}^{\mathrm{SS}} + G_{s,u}^{\mathrm{FF}} \tag{3}$$

Note that, on the right-hand side of Eq. (3), we omit the time indices (t,k) and the subscript dB for simplicity. The path loss $G_{s,u}^{\mathrm{PL}}$ (in dB) can be written as

$$G_{s,u}^{\mathrm{PL}} = \mathrm{FSPL}(d,f_s) + \mathrm{SF} + \mathrm{CL}(\alpha_{\mathrm{EA}},f_s) \tag{4}$$

where $\mathrm{FSPL}(d,f_s) = 32.45 + 20\lg(f_s) + 20\lg d$ is the free-space path loss and d represents the distance (in meters) between the user and the LEO satellite. $\mathrm{SF} \sim N(0,\sigma_{\mathrm{SF}}^2)$ represents the shadow fading modeled as a log-normal random variable, where N is log-normal distribution and the variance σ_{SF}^2 is related to the propagation scenarios, the frequency, the path loss condition (line of sight (LOS) or non-line of sight (NLOS)) and elevation angle (denoted as α_{EA}) [14]. For the clutter loss $\mathrm{CL}(\alpha_{\mathrm{EA}},f_s)$, its calculation is similar to shadow fading.

Atmospheric absorption $G_{s,u}^{\mathrm{AA}}$ plays a crucial role in the satellite–ground channel [37]. According to the 3GPP standard, the additional path loss caused by atmospheric gases can be calculated as follows:

$$G_{s,u}^{\mathrm{AA}} = \frac{A_{\mathrm{zenith}}(f_s)}{\sin\alpha_{\mathrm{EA}}} \tag{5}$$

where $A_{\mathrm{zenith}}(f_s)$ is zenith attenuation. Atmospheric fading is only considered at frequencies greater than 10 GHz, or at any frequency of $\alpha_{\mathrm{EA}} < 10^\circ$.

The scintillation $G_{s,u}^{\mathrm{SS}}$ is divided into two parts: ionospheric scintillation (IS) and tropospheric scintillation. The IS was based on the gigahertz scintillation model. In mid-latitude regions (between 20° and 60°) or at frequencies above 6 GHz, IS is generally negligible; otherwise, it is modeled as

$$\mathrm{IS} = \left(\frac{f_s}{4}\right)^{-1.5} \cdot \frac{P_{\mathrm{flu}}}{\sqrt{2}} \tag{6}$$

where P_{flu} is the scale factor of the ionospheric attenuation level. Unlike IS, the effect of tropospheric scintillation increases with the frequency and becomes significant at frequencies above 10 GHz.

In general, fast-fading $G_{s,u}^{\mathrm{FF}}$ is described by a frequency-selective fading model based on 3GPP TR 38.901 [38], which is widely applicable to cellular network scenarios; however, the configuration parameters of satellite networks are different.

For users $u \in \mathscr{M}_{\mathrm{DT}}$ with the DT services, the average achievable throughput provided by LEO satellite s is obtained based on Shannon's theorem and can be expressed as

$$D_{s,u} = W_{s,u}\log_2\left(1 + \frac{P_s \cdot h_{s,u}(t,k)}{W_{s,u}N_0 + \sum_{i_s \in \mathscr{I}} I_{i_s}}\right) \tag{7}$$

where $D_{s,u}$ is the average achievable throughput and P_s denotes the satellite transmission power. Constraint N_0 is the noise power density. $W_{s,u}$ is the spectrum bandwidth allocated to user u. The interference from other satellites is denoted as $\sum_{i_s \in \mathscr{I}} I_{i_s}$, where $\mathscr{I}$ represents the set of interfering satellites.

For users $u \in \mathscr{M}_{\mathrm{DS}}$ with the DS services, since their packet sizes are much smaller than those in traditional services, the finite block-length capacity theorem [39] should be applied. The maximum data rate that can be achieved in one time slot with a decoding error probability ψ is given by

$$D_{s,u}(\psi) \approx W_{s,u}\left[\log_2\left(1 + \frac{P_s \cdot h_{s,u}(t,k)}{W_{s,u}N_0 + \sum_{i_s \in \mathscr{I}} I_{i_s}}\right) - \sqrt{\frac{V_{\mathrm{ch}}}{W_{s,u}T_K}} {f_Q}^{-1}(\psi)\right] \tag{8}$$

where ${f_Q}^{-1}(\cdot)$ is the inverse of the Q-function. T_K is the duration of each time slot. $V_{\mathrm{ch}} = \frac{1}{(\ln 2)^2}\left[1 - \left(1 + \frac{P_s \cdot h_{s,u}(t,k)}{W_{s,u}N_0 + \sum_{i_s \in \mathscr{I}} I_{i_s}}\right)^{-2}\right]$ is the channel dispersion.

3.3. Problem formulation

We divided the participants into NOs, SPs, and users. Each SP represents a type of service and applies network resources from the NO to meet the demands of its users. To ensure isolation between different services, the resources allocated to each SP must remain fixed over a specific period to prevent traffic load variations from affecting other SPs. Simultaneously, to adapt to dynamic network environments, resources must be adjusted in real-time based on traffic load conditions to mitigate the potential negative effects of environmental changes on network performance. To achieve this, resources must be reserved in advance for each SP, ensuring that each SP has exclusive access to a portion of the total resources over a given period, and avoiding frequent reconfigurations. Additionally, when user demand within an SP increases, the system should promptly allocate the remaining resources to meet the users' real-time needs.

To address this, we consider resource scheduling on two timescales, where the resource reservation for each SP is updated at the beginning of each time window, and additional resource adjustments are performed at the start of each time slot. In time window t, SP m decides its reservation plan $x_t^{s,m}$ for this window, where $x_t^{s,m}$ represents the proportion of bandwidth resources reserved by LEO satellite s to SP m, relative to all available resources. At the beginning of each time slot k, SP m determines its adjustment plan $y_k^{s,m}$ for this time slot, where $y_k^{s,m}$ also represents the proportion similar to $x_t^{s,m}$. We use $\boldsymbol{y}_t^{s,m} = (y_k^{s,m}, k \in \mathscr{K}_t)$ to denote the adjustment amounts over period t, where $\mathscr{K}_t = \{(t-1)K+1, \ldots, tK\}$. Due to the dynamic nature of satellite networks, the revenue for the same resource proportion may vary across different time slots. For example, the bandwidth of a link may change due to satellite movement. The data rate obtained by SP m's resource scheduling strategy $\{x_t^{s,m}, \boldsymbol{y}_t^{s,m}\}$ can be quantified by $x_t^{s,m}\boldsymbol{\theta}_t$ and $\boldsymbol{y}_t^{s,m}{\boldsymbol{\theta}_t}^T$, respectively, where $\boldsymbol{\theta}_t = (\theta_k, k \in \mathscr{K}_t)$ and θ_k respectively represent the performance contribution values of each unit of resource proportion within time window t and time slot k.

To facilitate the unified management of resource reservation in time windows and resource adjustment in small time slots, we introduce a pricing mechanism. Let $p_t \in \mathbb{R}_+$ denote the reservation price per unit of resource proportion within time window t, let $q_k \in \mathbb{R}_+$ represent the adjustment price per unit of resource proportion in time slot k, and let vector $\boldsymbol{q}_t = (q_k, k \in \mathscr{K}_t)$ denote the adjustment price within time windows t. Both p_t and $\boldsymbol{q}_t$ are announced by the NO and will fluctuate according to the dynamic changes of service demand and resource supply, used to measure the scarcity of resources. For example, p_t and q_k in our study are defined as the ratios of the total service demand within a time window (or time slot) to the available network resources. Furthermore, macrolevel control over resource scheduling can be achieved by setting different prices for resource reservations and adjustments. For example, when the system prioritizes service isolation, p_t can be reduced to incentivize SPs to reserve resources in advance, ensuring that critical services have access to the necessary resources when needed.

Our objective was to satisfy the on-demand QoS requirements of each SP with minimal resource occupation. The resource scheduling strategy for the entire system can be formulated as follows:

$$\min_{\{x_t^{s,m}, y_t^{s,m}\}_{t=1}^{T}} \sum_{m\in\mathscr{M}} \sum_{t=1}^{T} \sum_{k\in\mathscr{K}_t} \left(x_t^{s,m} + y_k^{s,m}\right) \tag{9a}$$

$$\text{s.t. } x_t^{s,m}\theta_k + y_k^{s,m}\theta_k \geq E_m(k),\ \forall m \in \mathscr{M}_{\mathrm{DT}}, \forall k \in \mathscr{K}_t \tag{9b}$$

$$\mathbb{P}\left[\left(x_t^{s,m}\theta_k + y_k^{s,m}\theta_k\right) \geq E_m(k)\right] \geq 1-\alpha,\ \forall m \in \mathscr{M}_{\mathrm{DS}}, \forall k \in \mathscr{K}_t \tag{9c}$$

$$\sum_{k\in\mathscr{K}_t} \left(x_t^{s,m} p_t + y_k^{s,m} q_k\right) \leq B_t, \forall m \in \mathscr{U}, \forall t \in \mathscr{T} \tag{9d}$$

$$x_t^{s,m} + y_k^{s,m} \leq D_{\max}^{s,m}, \forall m \in \mathscr{U}, \forall k \in \mathscr{K}_t \tag{9e}$$

$$x_t^{s,m}, y_k^{s,m} \in \left[0, D_{\max}^{s,m}\right], \forall m \in \mathscr{U}, \forall k \in \mathscr{K}_t \tag{9f}$$

Eq. (9b) ensures that the data rate derived from the bandwidth resources allocated to the DT during each time slot is not less than the actual demand, where $E_m(k)$ represents the actual resource demand of m in time slot k. Eq. (9c) states that the probability of the resources allocated to DS meeting the actual data rate demand must be no less than $1-\alpha$, where $\alpha \in (0,1]$ represents the level of service dissatisfaction (also referred to as the significance level in the following text) and $\mathbb{P}(\cdot)$ represents the probability. The reason for the different QoS constraints of the two SPs is that the actual demands are always unknown, making it difficult to ensure that Eq. (9b) is satisfied. Simultaneously, owing to the higher requirements of DS, it is necessary to ensure a certain level of QoS satisfaction regarding uncertainty. Eq. (9d) ensures that the cost of the resources used by an SP does not exceed its budget, where B_t represents the budget in time window t. Eqs. (9e) and (9f) confine the decision variables to a convex set that imposes upper limits on the resource occupation ratios for each SP, where $D_{\max}^{s,m}$ represents the maximum bandwidth allocation that LEO satellite s can provide to SP m.

The key to solving the problem of Eqs. (9a)–(9f) lies in meeting the actual service demands specified in Eqs. (9b) and (9c). Due to the nonstationary and time-varying nature of traffic, accurately predicting demands in advance is challenging. However, in real-world environments, traffic loads often follow specific patterns, making it feasible to predict service demand. Accordingly, the system can forecast future service demand variations based on historical traffic data and use these predictions as the basis for resource scheduling. Given the distinct prediction objectives tailored to the characteristics of each SP, we elaborate on them individually below:

For users of DT services, the service durations are typically long and the delay requirements are not strict. Therefore, the prediction objective is to minimize the average mean square error (MSE) between the predicted resource demand $\widetilde{E}_m(k)$ and the actual resource demand $E_m(k)$ over the time window:

$$\arg\min_{\widetilde{E}_m(k)} \frac{1}{K} \sum_{k\in\mathscr{K}_t} \left|\widetilde{E}_m(k) - E_m(k)\right|^2, m \in \mathscr{M}_{\mathrm{DT}} \tag{10}$$

Owing to the high reliability and latency requirements of DS users, it is essential to quantify the uncertainty of predictions. This enhances the credibility of the prediction results and mitigates the risk of service dissatisfaction. The goal is to provide prediction intervals with a user-specified level of dissatisfaction α, that is,

$$\mathbb{P}\left(E_m(k) \in \widehat{C}_m(k)\right) \geq 1-\alpha \tag{11}$$

where $\widehat{C}_m$ represents the prediction interval, that is, the range around the predicted point $\widetilde{E}_m$.

To address the complex optimization problem described in Eqs. (9a)–(9f), we propose an online learning-based STIN resource scheduling framework, as shown in Fig. 2. The proposed framework divides the original problem into three modules: link simulation, traffic prediction, and resource scheduling. Specifically, the link simulation module accurately represents the satellite-to-ground link bandwidth resources. The traffic prediction module forecasts future service demands and uncertainties, while the resource scheduling module optimizes resources on two timescales. We elaborate on each module in the following section.

4. Prediction model

It is evident that the service demands are the basis of resource scheduling in Eqs. (9a)–(9f); therefore, a traffic prediction module is required. In this section, we introduce the prediction module that includes a single-point prediction for the demand value and an uncertainty prediction for the intervals.

4.1. Single-point prediction

For single-point prediction, local features of time-series data can be effectively extracted using 1D-Conv, followed by LSTM to capture temporal dependencies. The attention mechanism then assigns different weights to these features. We combine 1D-Conv, LSTM, and the attention mechanism to build a traffic prediction model that delivers better performance than using a single model.

(1) **1D-Conv network:** 1D-Conv is the key to feature extraction, where local perception domains are obtained using sliding kernel filters [40]. Assume the input is $\boldsymbol{V}_k \in \mathbb{R}^{n\times 1}$, where n is the length of the input data. The output of the 1D-Conv is fed into the LSTM, which can be obtained as follows:

$$\boldsymbol{V}_k' = \mathrm{ReLu}[\boldsymbol{V}_k * \boldsymbol{W}_{\mathrm{cd}} + b_{\mathrm{cd}}] \tag{12}$$

where $\boldsymbol{W}_{\mathrm{cd}}$ and b_{cd} are the weight and deviation of the filters, respectively, ReLu is the activation function, $*$ indicates the convolution operations, and $\boldsymbol{V}_k'$ represents the output of the 1D-Conv network.

(2) **LSTM network:** LSTM has the ability to handle relatively long sequences and capture long-term dependencies, making it widely used in time series prediction [41]. LSTM regulates the information flow by introducing gated cell structures. The construction of a single LSTM unit consists of two states (i.e., cell state $\boldsymbol{C}_k$ and hidden state $\boldsymbol{H}_k$) and three gates (i.e., forget gate $\boldsymbol{f}_k$, input gate $\boldsymbol{i}_k$, and output gate $\boldsymbol{o}_k$), whose calculation formulas are as follows:

$$\begin{cases} \boldsymbol{f}_k = \sigma\left(\boldsymbol{W}_f^x \cdot \boldsymbol{V}_k' + \boldsymbol{W}_f^h \cdot \boldsymbol{H}_{k-1} + \boldsymbol{b}_f\right) \\ \boldsymbol{C}_k = \boldsymbol{f}_k \circ \boldsymbol{C}_{k-1} + \boldsymbol{i}_k \circ \tanh\left(\boldsymbol{W}_c^x \cdot \boldsymbol{V}_k' + \boldsymbol{W}_c^h \cdot \boldsymbol{H}_{k-1} + \boldsymbol{b}_c\right) \\ \boldsymbol{H}_k = \boldsymbol{o}_k \circ \tanh(C_k) \\ \boldsymbol{i}_k = \sigma\left(\boldsymbol{W}_i^x \cdot \boldsymbol{V}_k' + \boldsymbol{W}_i^h \cdot \boldsymbol{H}_{k-1} + \boldsymbol{b}_i\right) \\ \boldsymbol{o}_k = \sigma\left(\boldsymbol{W}_o^x \cdot \boldsymbol{V}_k' + \boldsymbol{W}_o^h \cdot \boldsymbol{H}_{k-1} + \boldsymbol{b}_o\right) \end{cases} \tag{13}$$

where the weight matrices $\boldsymbol{W}_c^x$, $\boldsymbol{W}_c^h$, $\boldsymbol{W}_f^x$, $\boldsymbol{W}_f^h$, $\boldsymbol{W}_o^x$, $\boldsymbol{W}_o^h$, $\boldsymbol{W}_i^x$, $\boldsymbol{W}_i^h$ and biases $\boldsymbol{b}_f, \boldsymbol{b}_c, \boldsymbol{b}_o, \boldsymbol{b}_i$ are all trainable parameters, and $\circ$ denotes the Hadamard product. The functions $\sigma(\cdot)$ and $\tanh(\cdot)$ represent the sigmoid function and the hyperbolic tangent function, respectively. Here, the external input interface is $\boldsymbol{V}_k'$ and the output interface is $\boldsymbol{H}_k$.

(3) **Attention mechanism:** Conceptualized as weighting, attention aims to mimic the cognitive way the human brain quickly identifies and concentrates on important information within complex environments [42]. This mechanism evaluates the importance of the input and aggregates the most relevant elements to generate the final output, while effectively disregarding irrelevant details. In

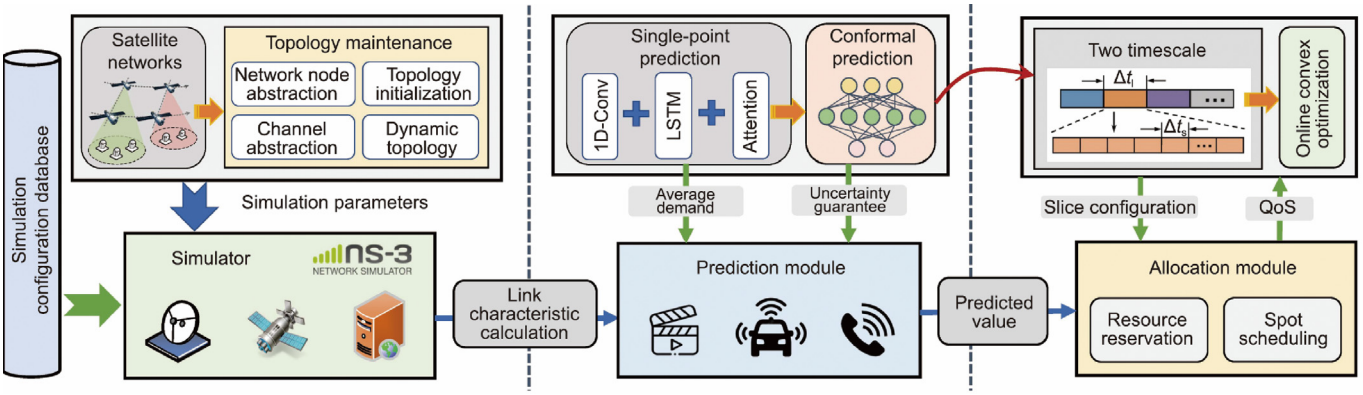

Fig. 2. Overview of the online learning-based STIN resource scheduling framework. Δt_l: the size of the unit time window; Δt_s: the size of the unit time slot.

this study, we apply an attention mechanism to enhance the weighting of peaks in traffic prediction.

The hidden state $\boldsymbol{H}_k$ is used as the LSTM output, also as the input of the attention mechanism. The output $\widetilde{\boldsymbol{E}}_k$ is calculated as follows:

$$\widetilde{\boldsymbol{E}}_k = \sum_k \beta_k \boldsymbol{H}_k \tag{14}$$

where β_k is the attention weight, which is represented by the softmax function.

$$\beta_k = \text{softmax}(s_k) = \frac{\exp(s_k)}{\sum_{k=1}^{K} \exp(s_k)} \tag{15}$$

where the score s_k represents $s_k = \tanh(W_{\text{att}}\boldsymbol{H}_k)$ and W_{att} is the weighting.

The single-point prediction model CA-LSTM is shown in Fig. 3. In this synergistic architecture, the 1D-Conv layer first refines the local features to enhance the input quality for LSTM processing. The LSTM layer captures temporal dependencies but may be influenced by irrelevant historical context. To address this, the attention layer automatically filters out noncritical time steps, highlighting key information. Through the joint optimization of these three modules, the CA-LSTM model adapts to diverse data patterns, with each module compensating for the limitations of the others, forming a comprehensive and robust prediction model.

4.2. Conformal prediction

CP is a general scheme for quantifying uncertainty, which has attracted considerable attention in recent years. It provides a statistically valid prediction interval with the desired significance-level requirement for arbitrary machine learning models. Compared with traditional probabilistic predictors, CP offers the following significant advantages:

- **Model-free:** CP is compatible with various predictive models and can be applied as a post-processing step to trained models without the need for hyperparameter tuning in the original models.
- **Distribution-free:** CP does not rely on any assumptions about the probability distribution of the data, but only on the relatively weaker condition in which the data are exchangeable.

Suppose a sequence of observations (A_k, E_k), where E_k is the service demand and A_k denotes the features, generally historical traffic. Let the first K_{tr} data be the training data $\{(A_k, E_k)\}_{k=1}^{K_{\text{tr}}}$. The goal is to construct prediction intervals sequentially from time $K_{\text{tr}} + 1$

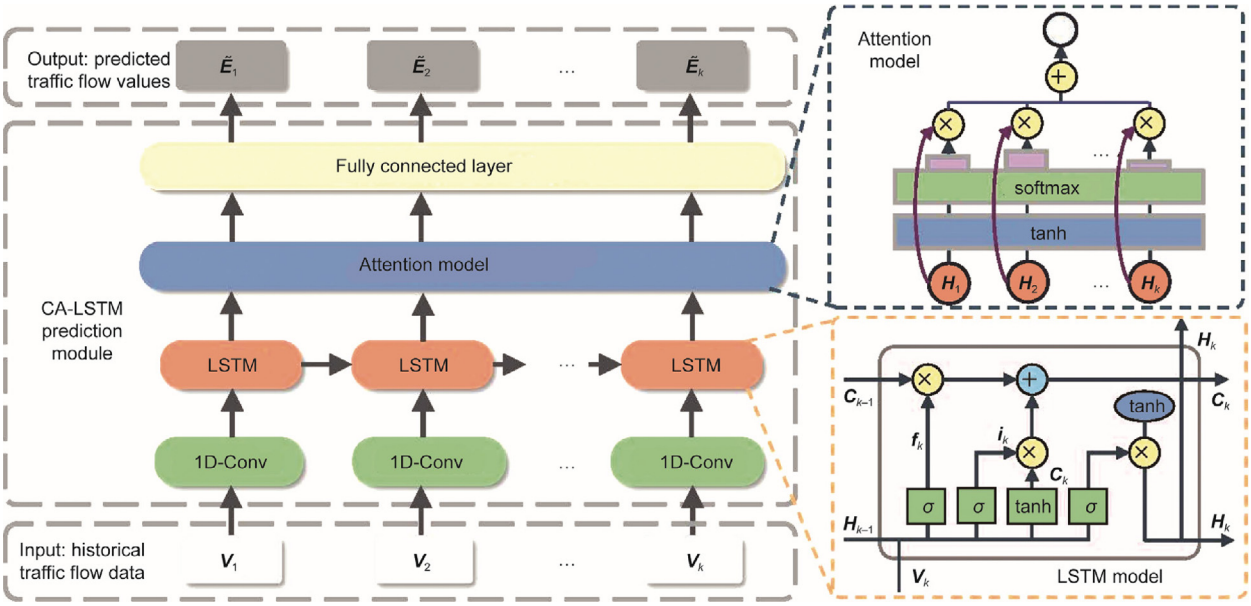

Fig. 3. Structure of the single-point prediction module.

that encompass the true values with a user-specified level of satisfaction $(1-\alpha)$.

$$\mathbb{P}\left(E_k \in \widehat{C}_{k-1}(A_k)\right) \geq 1-\alpha \tag{16}$$

The prediction intervals $\widehat{C}_{k-1}(A_k)$ are around single-point predictions $\widehat{f}(A_k)$ described previously and the subscript $k-1$ indicates that the interval can be built using up to $k-1$ previous observations. In extreme cases, selecting the entire range of real numbers always yields true values. To avoid vacuous prediction intervals, the width of the prediction interval $|\widehat{C}_{k-1}(A_k)|$ should be as narrow as possible without significant violations to reflect the difficulty of accurate predictions. A wider interval indicates a greater uncertainty or difficulty in achieving precise predictions.

CP uses a nonconformity score to estimate the prediction intervals, where a commonly used nonconformity score is the absolute residual $\widehat{\epsilon}_k$ between the true value E_k and the corresponding prediction $\widehat{E}_k = \widehat{f}(A_k)$.

$$\widehat{\epsilon}_k = \left|E_k - \widehat{E}_k\right| = \left|E_k - \widehat{f}(A_k)\right| \tag{17}$$

By calculating the non-conformity score for each data point, we obtained a distribution of scores that reflects the range of nonconformities observed in the training set. This distribution serves as a reference for evaluating the non-conformity of new query data, and the prediction interval can then be computed as

$$\left|\widehat{C}_{k-1}(A_k)\right| = \widehat{f}(A_k) \pm \left(\lceil (1-\alpha)(K_{\mathrm{tr}}+1)\rceil - \mathrm{th\ of}\ R_{\mathrm{asc}}\left\{\widehat{\epsilon}_j\right\}_{j=1}^{K_{\mathrm{tr}}}\right) \tag{18}$$

where $\lceil\cdot\rceil$ denotes the ceiling function, $R_{\mathrm{asc}}\{\cdot\}$ arranges the elements in ascending order, and j is the sequence number of the training data. Equivalently, there is

$$\left|\widehat{C}_{k-1}(A_k)\right| = \widehat{f}(A_k) \pm Q_{1-\alpha}\left(\frac{1}{K_{\mathrm{tr}}+1}\delta_{+\infty} + \sum_{j=1}^{K_{\mathrm{tr}}}\frac{1}{K_{\mathrm{tr}}+1}\delta_{\widehat{\epsilon}_j}\right) \tag{19}$$

where $Q_{1-\alpha}(\cdot)$ denotes the $(1-\alpha)$-quantile and $\delta_{+\infty}$ denotes the probabilistic mass.

Nevertheless, the desired coverage guarantee of the traditional CP methods relies on the critical assumption of data exchangeability. The exchangeable sequence $(A_1, A_2, \ldots, A_q)$ means that for any permutation of the observations ρ, the joint probability distribution remains unchanged, that is,

$$\mathbb{P}(A_1, A_2, \ldots, A_q) = \mathbb{P}(A_{\rho(1)}, A_{\rho(2)}, \ldots, A_{\rho(q)}) \tag{20}$$

where q is the length of the sequence.

In real-world applications, ordered data such as time series tend to exhibit significantly strong correlations, making it challenging to satisfy the exchangeability assumption. Referring to Ref. [43], we developed a novel CP algorithm for sequential data with theoretical guarantees, and applied it to uncertainty prediction in service demands. The main idea of sequential CP is to leverage the "feedback" structure of prediction residuals to achieve the desired coverage. As shown in Fig. 4, past residuals were used to perform a quantile regression for future residuals by exploiting the temporal correlation across the prediction residuals. To this end, we propose an ASCP algorithm.

The ASCP algorithm consists of two phases: training and prediction. In the training phase, the ASCP first fits a set of bootstrap estimators based on the training data. Using the leave-one-out (LOO) method, the prediction results of these bootstrap estimators on the training data are aggregated to generate LOO-predicted values and LOO residuals, which are used for subsequent predictions. In the prediction phase, for each data point, the center of the prediction interval is calculated by aggregating the LOO-predicted values. A quantile estimator is then trained using historical LOO residual sets to construct the prediction intervals. Once the actual response variable was observed in the new prediction data, the residual set was immediately updated by removing the old values and adding new ones to ensure the adaptiveness of the PIs. The core steps are as follows.

(1) **Bootstrap estimator:** The bootstrap estimator is a statistical estimation technique based on the bootstrap method, which is a resampling technique that generates multiple new sample sets by performing random sampling with replacement from the original dataset. These sample sets are then used to estimate the distribution of statistics. The bootstrap method does not rely on assumptions about the data distribution, making it capable of providing reliable estimates, even for small sample sizes. The bootstrap estimator $\widehat{f}_b$ can be expressed as

$$\widehat{f}_b = \mathscr{SP}\left(A_{i_b}, E_{i_b}\right),\ i_b \in S_b \tag{21}$$

where $S_b \subset [1, 2, \ldots, K_{\mathrm{tr}}]$ represents the index set obtained from the bth bootstrap resampling with replacement. $\mathscr{SP}(\cdot)$ denotes arbitrary single point prediction model, (A_{i_b}, E_{i_b}) is the i_bth training data.

(2) **LOO ensemble prediction:** The basic idea of the LOO method involves sequentially treating each data point in the dataset as a test sample, while using all remaining samples as the training set. This process was repeated until every individual sample served as a test sample exactly once, thereby constructing multiple models. The predictions from these models are then aggregated to maximize the utilization of the limited training data while enhancing both the robustness and accuracy of the predictions.

$$\widehat{f}_{-k}(A_k) = \phi\left(\left\{\widehat{f}_b(A_k) : k \notin S_b\right\}_{b=1}^{B}\right) \tag{22}$$

where $\phi(\cdot)$ represents an aggregation function for scalar values, where we specifically choose the mean function. $\widehat{f}_{-k}(A_k)$ denotes the LOO ensemble prediction model. B is the number of bootstrap estimators.

(3) **Conditional quantile estimator:** The latest feedback from the prediction residuals obtained by the LOO model is used to construct the sequential prediction interval, where the estimate of the conditional quantile estimator is adopted. The center of the prediction interval is calculated by aggregating the LOO-predicted values, and the prediction interval $\widehat{C}_{k-1}$ can be expressed as

$$\widehat{C}_{k-1}(A_k) = \left[\widehat{f}_{-k}(A_k) + \widehat{Q}_k\left(\widehat{\beta}\right), \widehat{f}_{-k}(A_k) + \widehat{Q}_k\left(1-\alpha+\widehat{\beta}\right)\right] \tag{23}$$

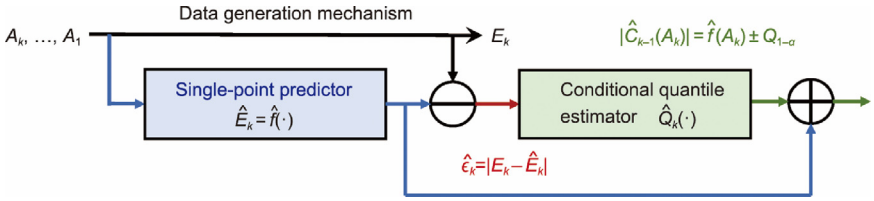

Fig. 4. Structure of the CP.

where $\widehat{\beta}=\operatorname{argmin}_{\beta\in[0,\alpha]}\left(\widehat{Q}_k(1-\alpha+\beta)-\widehat{Q}_k(\beta)\right)$ is to minimizes interval width. Here, we used quantile random forest (QRF) algorithm $\mathscr{Q}_{\mathrm{qr}}$ to train the quantile estimator $\widehat{Q}_k$.

(4) **Residual sliding update:** During prediction process, when a new true value is observed, the current residual is computed and added to the residual set, and the oldest residual is removed. Simultaneously, the quantile estimator was refitted with an updated residual set to recalculate the prediction intervals, ensuring that the constructed intervals had adaptive widths. Specifically, we use the past $w\geq 1$ residuals to predict the conditional quantile of future residuals. Let $\widetilde{K}_{\mathrm{tr}}=K_{\mathrm{tr}}-w$. Here $\mathscr{E}_{k'}^{w}=\{\widehat{\epsilon}^{t}{}_{k'+w-1},\ldots,\widehat{\epsilon}_{k'}\}$ consists of w residuals to predict the conditional quantile of $\widetilde{\mathscr{E}}_{k'}^{w}=\widehat{\epsilon}_{k'+w}$ for $k'=1,\ldots,\widetilde{K}_{\mathrm{tr}}$. Therefore, $\left(\mathscr{E}_{k'}^{w},\widetilde{\mathscr{E}}_{k'}^{w}\right)$ is used to train QRF, while continuously updating the residual set to reflect the most recent prediction errors.

The overall description of the ASCP is shown in Algorithm 1. In summary, the ASCP provides an efficient, accurate, and adaptive method for constructing prediction intervals by combining bootstrap methods with LOO prediction, which can dynamically adjust the PIs on the test data to ensure accuracy and reliability. According to Ref. [43], it can be proven that ASCP can maintain marginal coverage when the data satisfies exchangeability, and for time series data, ASCP can provide asymptotic guarantee of prediction interval coverage.

Algorithm 1. ASCP.

Require: Training data $\{(A_k,E_k)\}_{k=1}^{K_{\mathrm{tr}}}$, arbitrary single-point prediction model $\mathscr{SP}(\cdot)$, significance level α, quantile random forest algorithm $\mathscr{Q}_{\mathrm{qr}}$, number of bootstrap estimators B

Output: Prediction intervals $\widehat{C}_{k-1}(A_k)$, $k>K_{\mathrm{tr}}$

1. **For** $b=1$ to B **do**
2. Based on the bootstrap method, perform random sampling with replacement
   From the indices of the original training dataset, that is, $\{S_b:S_b\subset[1,\ 2,\ldots,\ K_{\mathrm{tr}}]\}$
3. Train the corresponding bootstrap estimators $\widehat{f}_b$ on data $\{(A_{i_b},E_{i_b}),i_b\in S_b\}$, that is,
   $\widehat{f}_b=\mathscr{SP}(A_{i_b},E_{i_b}),\ i_b\in S_b$
4. **End for**
5. Initialize residual set $\widehat{\boldsymbol{\varepsilon}}=\{\ \}$ as an ordered set
6. **For** $k=1$ **to** K_{tr} **do**
7. Aggregate the prediction results of bootstrap estimators on the training data using the LOO method, that is,
   $\widehat{f}_{-k}(A_k)=\operatorname{mean}\left(\left\{\widehat{f}_b(A_k):k\notin S_b\right\}_{b=1}^{B}\right)$
8. Compute the residuals obtained from the LOO models
   $\widehat{\epsilon}_k=\left|E_k-\widehat{f}_{-k}(A_k)\right|$
9. Update the residual set to $\widehat{\boldsymbol{\varepsilon}}=\widehat{\boldsymbol{\varepsilon}}\cup\{\widehat{\epsilon}_k\}$
10. **End for**
11. **For** $t>K_{\mathrm{tr}}$ **do**
12. Calculate the center of the prediction interval by aggregating the LOO predicted values, that is,
    $\widehat{f}_{-t}(A_t)=\operatorname{mean}\left(\left\{\widehat{f}_{-i_{\mathrm{tr}}}(A_t),i_{\mathrm{tr}}=1,\ldots,\ K_{\mathrm{tr}}\right\}\right)$
13. Train the quantile estimator based on the most recent residual set $\widehat{\boldsymbol{\varepsilon}}$, that is,$\widehat{Q}_t\leftarrow\mathscr{Q}_{\mathrm{qr}}(\widehat{\boldsymbol{\varepsilon}})$
14. Compute $\widehat{\beta}=\operatorname{argmin}_{\beta\in[0,\alpha]}\left(\widehat{Q}_k(1-\alpha+\beta)-\widehat{Q}_k(\beta)\right)$
15. Compute the prediction intervals
    $\widehat{C}_{k-1}(A_k)=\left[\widehat{f}_{-k}(A_k)+\widehat{Q}_k\left(\widehat{\beta}\right),\widehat{f}_{-k}(A_k)+\widehat{Q}_k\left(1-\alpha+\widehat{\beta}\right)\right]$
16. Upon observing the actual response variable in the test data, advance the sliding window by one index to update the residuals, that is, $\widehat{\boldsymbol{\varepsilon}}=(\widehat{\boldsymbol{\varepsilon}}-\{\widehat{\epsilon}_1\})\cup\{\widehat{\epsilon}_t\}$
17. **End for**

5. Resource scheduling module

Based on the predictive values and intervals of service demands discussed in the previous section, this section introduces a resource scheduling module that incorporates resource reservation on a large timescale and resource adjustment on a small timescale. The problem described in Eqs. (9a)–(9f) is a convex optimization problem; however, it cannot be solved directly due to the following challenges: Ch1: The actual resource demand E_m of each service is unknown and is potentially generated by a nonstationary stochastic process. Even with predictions, the true value cannot be precisely determined. Ch2: Adjustment prices $\boldsymbol{q}_t$ and reservation prices p_t are time-varying and unpredictable, as they depend on the service demands of all SPs, which are private to each other. Considering the long-term optimization objective in Eq. (9a), we cannot initially solve the problem for the subsequent time windows because the future evolution of the system parameters is unknown. Therefore, it is imperative to design an online learning algorithm that can adapt to the dynamics of networks and SPs. Because SPs are relatively independent, we consider a certain SP as an example to analyze its resource scheduling strategy. To simplify the description, we simplified the variable notation by omitting the superscripts/subscripts related to the satellite and SP information. First, we analyze the resource scheduling strategy on a large timescale and define the resource occupancy, QoS satisfaction, and cost functions of the SP during one time window as

$$f_t(x_t,\boldsymbol{y}_t)=\sum_{k\in\mathscr{K}_t}(x_t+y_k) \tag{24}$$

$$g_{t,1}(x_t,\boldsymbol{y}_t)=\sum_{k\in\mathscr{K}_t}E_m(k)-\boldsymbol{y}_t{}^{\mathrm{T}}\boldsymbol{\theta}_t-x_t\theta_t \tag{25}$$

$$g_{t,2}(x_t,\boldsymbol{y}_t)=x_tp_t+\boldsymbol{y}_t{}^{\mathrm{T}}\boldsymbol{q}_t-B_t \tag{26}$$

where f_t, $g_{t,1}$, and $g_{t,2}$ are the resource occupancy, QoS satisfaction, and cost functions of the SP during the time window t, respectively. Let vector $\boldsymbol{z}_t=(x_t,\boldsymbol{y}_t)\in\mathscr{Z}\triangleq\Gamma^K$ denote the resource scheduling strategy of the SP during the time window t, where $\mathscr{Z}$ is the resource scheduling strategy set and $\Gamma=[0,\ D_{\max}^{s,m}]$ represents the range of values for the resource scheduling strategy. By introducing dual variables $\boldsymbol{\lambda}=\{\lambda_1,\ \lambda_2\}$, the Lagrangian function $\mathscr{L}_t$ can be defined as

$$\mathscr{L}_t(\boldsymbol{z},\boldsymbol{\lambda})=\sum_{i'=1}^{t}\nabla f_t(\boldsymbol{z}_{i'})(\boldsymbol{z}-\boldsymbol{z}_t)+\sum_{i=1,2}\lambda_ig_{t,i}(\boldsymbol{z})+\frac{\|\boldsymbol{z}-\boldsymbol{z}_t\|^2}{2v} \tag{27}$$

where $\sum_{i'=1}^{t}\nabla f_t(\boldsymbol{z}_{i'})$ denotes the cumulative gradient of resource occupancy function, $\boldsymbol{z}_i$ is the resource scheduling strategy vector during the time window i', and $\boldsymbol{z}$ is the resource scheduling strategy vector variable. Because of the high randomness inherent in online learning, the cumulative gradient can help avoid misjudgments caused by excessive local fluctuations in certain sample dimensions. λ_i are Lagrangian multipliers of constraint. The last term is a

regularization, represented by a Euclidean regulator with a nonnegative argument v. Note that $f_t(\boldsymbol{z}_t)$ is omitted here because it has no impact on $\boldsymbol{z}$ and $\boldsymbol{\lambda}$.

The information about the current time window is revealed to the SP after a decision is made. At the beginning of time window t, each SP determines its reservation policy as follows:

$$\boldsymbol{z}_t = \arg\min_{\boldsymbol{z}} \mathscr{L}_{t-1}(\boldsymbol{z}, \boldsymbol{\lambda}_t) \tag{28}$$

As mentioned above, SP makes this decision without having access to f_t, $g_{t,1}$, or $g_{t,2}$, which can be inferred from the time index $t-1$ in the Lagrangian function. After $\boldsymbol{z}_t$ is fixed, demand and price are revealed, at which point the SP measures its resource occupancy $f_t(\boldsymbol{z}_t)$, QoS satisfaction $g_{t,1}(\boldsymbol{z}_t)$, and cost $g_{t,1}(\boldsymbol{z}_t)$. At the end of this window, SP can access the current Lagrangian value and update its dual variables by performing dual gradient rise with step size μ, that is,

$$\boldsymbol{\lambda}_{t+1} = \boldsymbol{\lambda}_t + \mu \nabla_{\boldsymbol{\lambda}} \mathscr{L}_t(\boldsymbol{z}_t, \boldsymbol{\lambda}) \tag{29}$$

These dual variables assist the SP in resource scheduling decision-making in the subsequent time window. Furthermore, we discuss the resource adjustment strategy for each time slot. In each time slot k, the resource occupancy, QoS satisfaction, and cost functions are expressed as follows:

$$f_k(y_k) = x_t + y_k \tag{30}$$

$$g_{k,1}(y_k) = E_m(k) - y_k\theta_k - x_t\theta_k \tag{31}$$

$$g_{k,2}(y_k) = (x_t p_t - B_t)/K + y_k q_k \tag{32}$$

where x_t is treated as a parameter fixed at the beginning of t. Similarly, the Lagrangian function of time slot k is

$$\mathscr{L}_k\left(y, \widehat{\boldsymbol{\lambda}}\right) = \sum_{j'=1}^{k} \nabla f_k(y_{j'})(y - y_k) + \sum_{i''=1,2} \widehat{\lambda}_{i''} g_{k,i''}(y) + \frac{(y-y_k)^2}{2\widehat{v}} \tag{33}$$

where $\widehat{\lambda}_{i''}$ and $\widehat{v} > 0$ are the dual variables and regulator argument, respectively. y is the resource adjustment plan variable and $y_{j'}$ is the adjustment plan in time slot j'.

At the beginning of each time slot k, the SP updates its resource adjustment strategy y_k through the following equation:

$$y_k = \arg\min_{y \in \Gamma} \mathscr{L}_{k-1}\left(y, \widehat{\boldsymbol{\lambda}}_k\right) \tag{34}$$

After each time slot, the dual variables are updated based on the known network information.

$$\widehat{\boldsymbol{\lambda}}_{k+1} = \widehat{\boldsymbol{\lambda}}_k + \widehat{\mu} \nabla_{\boldsymbol{\lambda}} \mathscr{L}_k\left(y_k, \widehat{\boldsymbol{\lambda}}\right) \tag{35}$$

ORS scheme for the two timescales is summarized in Algorithm 2. In an ORS, each SP iteratively makes decisions to minimize cumulative resource occupancy. At the beginning of each time window t, the SP determines its reservation strategy by solving Eq. (27) without knowing the current actual demand and price. Once x_t is fixed, at the beginning of each time slot k, the SP updates its resource adjustment strategy y_k using Eq. (33). At the end of each time slot, the SP obtains the demand and price parameters to update its dual variables according to Eq. (34). Similarly, after committing to decision $(x_t, \boldsymbol{y}_t)$, the resource occupancy, QoS satisfaction, and cost functions are revealed, and the SP updates its dual variables based on Eq. (35).

Algorithm 2. ORS.

Initialize: Resource scheduling strategy $(x_0, \boldsymbol{y}_0)$, dual variable number $\widehat{\boldsymbol{\lambda}}_1 = \{0, 0\}$, $\boldsymbol{\lambda}_1 = \{0, 0\}$, learning rates v, $\widehat{v}$, μ, and $\widehat{\mu}$
Output: Resource scheduling strategy $\boldsymbol{z}_t = (x_t, \boldsymbol{y}_t)$

For $m \in \mathscr{M}_{\mathrm{DS}} \cup \mathscr{M}_{\mathrm{DT}}$ **do**
  For $t = 1$ **to** T **do**
    Determine the reservation strategy for the current time window x_t by solve
    $\boldsymbol{z}_t = \arg\min_{\boldsymbol{z}} \mathscr{L}_{t-1}(\boldsymbol{z}, \boldsymbol{\lambda}_t)$
    For $k = 1$ **to** K **do**
    Determine the adjustment strategy for the current time slot y_k by solve
    $y_k = \arg\min_{y} \mathscr{L}_{k-1}\left(y, \widehat{\boldsymbol{\lambda}}_k\right)$
    Replace the kth element of vector $\boldsymbol{y}_t$ by y_k
    Observe the k-slot adjustment price q_k and actual resource demand $E_m(k)$
    Calculate the resource occupancy function $f_k(y_k)$, QoS satisfaction function $g_{k,1}(y_k)$, and cost function $g_{k,2}(y_k)$ according to Eq. (30), Eq. (31), and Eq. (32), respectively
    Update $\widehat{\boldsymbol{\lambda}}_{k+1}$ by $\widehat{\boldsymbol{\lambda}}_{k+1} = \widehat{\boldsymbol{\lambda}}_k + \widehat{\mu} \nabla_{\boldsymbol{\lambda}} \mathscr{L}_k\left(y_k, \widehat{\boldsymbol{\lambda}}\right)$
    End for
    Observe the t-window reservation price p_t and actual service demand
    Calculate the resource occupancy function $f_t(x_t, \boldsymbol{y}_t)$, QoS satisfaction function $g_{t,1}(x_t, \boldsymbol{y}_t)$, and cost function $g_{t,2}(x_t, \boldsymbol{y}_t)$ as Eq. (24), Eq. (25), and Eq. (26), respectively
    Update $\boldsymbol{\lambda}_{t+1}$ by $\boldsymbol{\lambda}_{t+1} = \boldsymbol{\lambda}_t + \mu \nabla_{\boldsymbol{\lambda}} \mathscr{L}_t(\boldsymbol{z}_t, \boldsymbol{\lambda})$
  End for
End for

The ORS algorithm operates online, where resource scheduling strategies are learned online from previous experiences rather than being computed using a pre-given model. SPs can adaptively provide better strategies using more observations. Because the QoS satisfaction function can only be obtained retrospectively, it is unlikely that the actual cumulative resource occupancy will be minimized. Instead, an appropriate performance metric is the difference between the cumulative resource occupancy of the SP during the ORS and the optimal solution in hindsight, which is referred to as regret. Specifically, when full information is available at the start of each window, including the reservation price p_t, resource demand E_m, and adjustment price q_k, the optimal decision $\boldsymbol{z}_t^* = (x_t^*, \boldsymbol{y}_t^*)$ within t can be calculated using the CVX toolbox.

$$\begin{aligned} &\min_{\{\boldsymbol{z}_t\} \in \Gamma^{K+1}} f_t(\boldsymbol{z}_t) \\ &\text{s.t. } g_{t,1}(\boldsymbol{z}_t) \le 0, \;\; g_{t,2}(\boldsymbol{z}_t) \le 0 \end{aligned} \tag{36}$$

Formally, we define the dynamic regret as

$$R_T = \sum_{t=1}^{T} \left(f_t(\boldsymbol{z}_t) - f_t(\boldsymbol{z}_t^*)\right) \tag{37}$$

where R_T is the dynamic regret and quantifies how well our policy $(x_t, \boldsymbol{y}_t)$ fairs gains $(x_t^*, \boldsymbol{y}_t^*)$. Another commonly known metric for algorithm performance measurement is the constraint fit metric V_T, which is defined as

$$V_T = \left[\sum_{t=1}^{T} g_t(\boldsymbol{z}_t)\right]_+ \tag{38}$$

which quantifies the extent to which the constraints are violated, on average. Here, we are more concerned with the QoS constraints. If the regret grows sublinearly over time, the algorithm can be considered low-regret. This implies that the average low-regret algorithm performs as well as the optimal solution in hindsight as T goes to infinity. It is also necessary to satisfy the condition that the constraint violation metrics are sublinear as a function of T, that is,

$$\lim_{T\to\infty} \frac{R_T}{T} = 0, \ \lim_{T\to\infty} \frac{V_T}{T} = 0 \tag{39}$$

6. Numerical simulations

In this section, we conduct simulations to evaluate the performance of the proposed resource scheduling framework, including channel implementation, service demand prediction, and the resource scheduling scheme. We considered the constellation configuration of StarLink phase I as a typical satellite network case [44], including 72 orbits with 22 satellites in each orbit, where the satellite altitude was 550 km and the inclination angle was 53°. The default values of the basic parameters used in the simulations are listed in Table 1.

6.1. High-fidelity STIN simulation and link resource characterization

Compared to terrestrial networks, the dynamics of STINs have distinctive characteristics such as global coverage, high-speed mobility, periodicity, and predictability, all of which are primarily driven by satellite motion, while user mobility is typically negligible. The strictly deterministic nature of satellite orbital motion, governed by orbital mechanics, allows for precise predictions of future network topologies using ephemeris data, enabling accurate characterization of network dynamics. Leveraging the high predictability of satellite trajectories, we have developed a dedicated high-fidelity simulation platform for STINs. This platform supports dynamic topology construction and maintenance, protocol stack development and configuration, and provides quantitative network performance statistics and evaluations [44]. In our simulation platform, we propose a dual-stage topology maintenance method specifically designed for the high dynamics of LEO satellites, including topology initialization and updating. In the topology initialization stage, the initial satellite coordinates are generated based on the given constellation configuration parameters. In the topology update stage, the current satellite coordinates are calculated based on the orbit prediction model [45]. Under this framework, discrete motion with appropriate time resolution is used to simulate the continuous motion of nodes based on DES theory.

Table 1
Simulation parameters.

Module	Parameters	Value
Channel module	Satellite antenna gain (circular aperture)	40 dB
	Minimum elevation angle	30°
	Transmission power	50 dBm
	Noise power density	−174 dBm·Hz^{-1}
	Frequency	20 GHz
	Scenario	Suburban
Prediction module	Number of convolution kernels	3
	Number of hidden layers	2
	Hidden units of each layer	32
	Number of trees in QRF	10
	Max depth for fitting QRF	2
	Number of bootstrap models	20
	Learning rate	0.001
	Number of epochs	600
	Time step	12
	Batch size	64
Scheduling module	Time window	60 s
	Time slot	5 s
	Step size of the primal (slot) update	0.01
	Step size of the dual (slot) update	0.05

In the STIN simulation, the physical layer module is responsible for simulating real-world signal propagation environments, including mobility models, antenna patterns, and propagation scenarios for the transmitting or receiving node. This enables a realistic characterization of various environmental factors in STINs. Building on the communication model described in Section 3.2, NTN physical layer modules based on 3GPP TR 38.811 and 38.901 were further developed within the simulation platform. These modules include antenna, channel, and statistical components, allowing for accurate characterization of ground-to-satellite link (GSL) resources.

- **Antenna module:** This module provides an interface for antenna gain calculation, which computes the gain based on the azimuth and elevation angles of the transceivers. It supports multiple functions, such as frequency setting, maximum antenna gain setting, azimuth/elevation angle setting, and conversion calculation.
- **Channel module:** Extended from Network Simulator 3 (NS-3)'s propagation module, the channel model described in Section 3.2 is implemented. It supports various propagation scenarios (dense urban, urban, remote, and rural areas), calculating the received power based on the transmission power and positions of the transceivers.
- **Statistical module:** This module supports the quantitative statistical analysis of various network performances. This study focuses on the calculation of the link signal-to-noise ratio (SNR) to reflect dynamic changes in bandwidth resources, which provides critical data support for subsequent resource-scheduling algorithms.

To demonstrate this phenomenon intuitively, we simulated the variation in the SNR of the GSLs over a period using our high-fidelity and high-efficiency STIN simulation platform. All satellites were initialized based on two-line element (TLE) data. After generating the initial constellation's TLE, it was used as input for the mobility model to maintain the nodes' mobility. The simplified general perturbation (SGP4) orbital prediction model was employed for the satellite mobility model, accounting for both long-term and periodic variations caused by the Earth's non-spherical shape, the gravitational effects of the Sun and Moon, gravitational resonance effects, and orbital decay. Based on this, we developed coordinate systems and geocentric mobility models to accurately simulate satellite movement.

The simulation was based on the constellation parameters of 1584 LEO satellites from StarLink phase I, with four satellite Earth stations (SESs) located in Tokyo, Japan (35°41′N, 139°41′E), São Paulo, Brazil (23°32′S, 46°38′W), Shanghai, China (31°13′N, 121°27′E), and Mumbai, India (19°04′N, 72°52′E). The simulation employed a minimum-distance handover strategy, where a ground station selected the nearest LEO to establish a GSL until the angle between the LEO and ground station fell below the minimum elevation threshold. At that point, the ground station switched to the next nearest LEO. As shown in Fig. 5, the SNR of all four GSLs exhibited significant periodic fluctuations. These fluctuations were primarily caused by the rapid movement of LEO satellites, which continuously altered the signal propagation conditions of GSLs. For example, SES 2 sequentially established GSLs with LEOs 273,

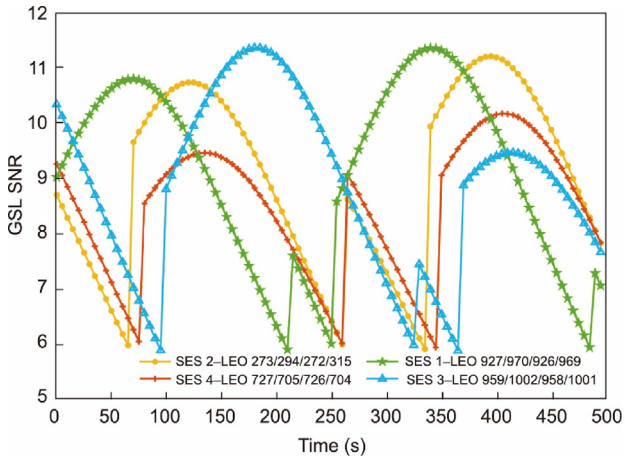

Fig. 5. The SNR diagram of GSLs.

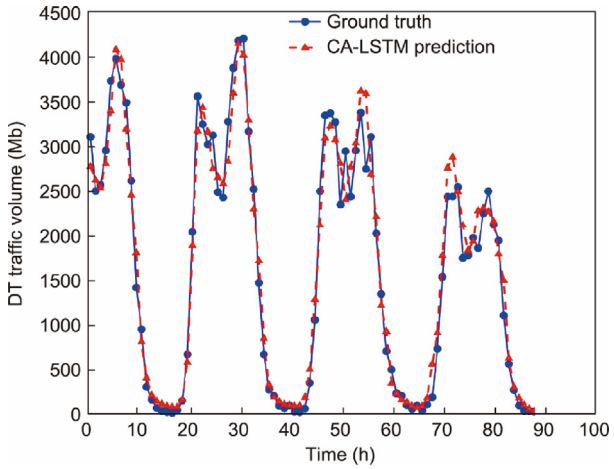

Fig. 6. Traffic prediction results of DT service on partial testing set.

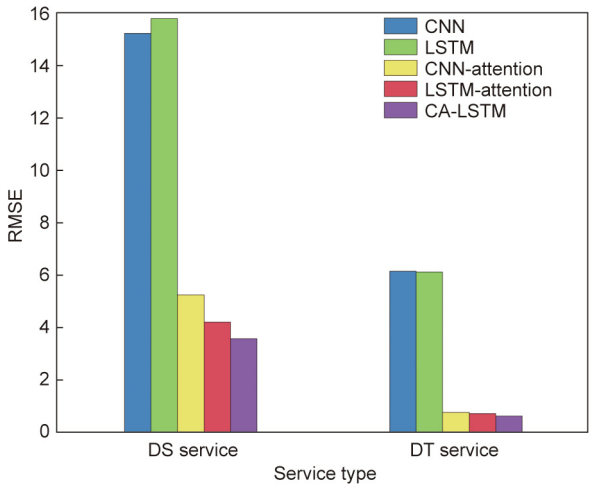

Fig. 7. Comparison of RMSE for traffic prediction among different predictors.

294, 272, and 315. The SNR of the GSL reached a peak at a certain time and then gradually decreased. When the angle between the current LEO satellite and SES reached the minimum elevation threshold, SES switched to the nearest satellite, causing an abrupt change in channel quality, and the SNR immediately rebounded. This pattern repeated cyclically. Additionally, the fluctuation range of the SNR for the same GSL was significant. For instance, the SNR of SES 3–LEO 1002 drops from a peak of approximately 11.5 dB to a minimum of approximately 5.9 dB within 150 s, representing a fluctuation of nearly 5.6 dB. Within the time span of 500 s shown in Fig. 1, the SNR for the same SES (i.e., users in the same region) changed frequently and often exhibited significant fluctuations within tens of seconds. This second-level variation was far faster than the minute-level changes observed in terrestrial networks. Moreover, within a short period, the SNR of the GSL can fluctuate by a factor of approximately two, leading to substantial variations in the maximum data rate provided by the GSL. This highlights that compared to terrestrial networks, satellite networks exhibit more pronounced and complex dynamics. These highly dynamic resource fluctuations impose stricter requirements on resource scheduling and QoS assurance in STIN.

6.2. Prediction effectiveness

We evaluated the proposed prediction approaches using a real-world traffic dataset [17] that recorded the traffic changes of three services: short message service (SMS), calls, and the Internet. We classify these services into DT and DS services and use traffic data within a certain period to represent the service resource demand for prediction and analysis. We divided the entire dataset with real values into two parts, where 80% was used as the training dataset and the remaining 20% was used as the test dataset. Fig. 6 shows the DT traffic prediction results for the partial testing set using the CA-LSTM structure. As observed, the actual traffic consistently exhibits a periodic pattern, making service demand prediction feasible. The predicted values from CA-LSTM were very close to the ground truth, demonstrating high accuracy and the model's ability to track changes in service resource demand effectively.

We compared the proposed CA-LSTM predictor with four other predictors: CNN, LSTM [41], CNN-attention, and LSTM-attention [46]. The root mean square error (RMSE) was used to evaluate the performance of the predictors. Fig. 7 presents a comparison of the RMSE for various predictors in the DT and DS traffic predictions. Compared to a single prediction module or algorithms combining two modules, our proposed approach can achieve the smallest RMSE, regardless of the service type. For single-point prediction, local features are first extracted using 1D-Conv, followed by LSTM to capture temporal correlations in sequences. The attention layer is then applied to improve the weighting of peak values in traffic flow prediction, resulting in superior performance compared to the other benchmarks.

For DS services with high latency and reliability requirements, single-point prediction alone is insufficient; therefore, CP is used to provide additional reliability. Next, we evaluated the performance of CP in uncertainty prediction. The traditional split CP (SCP) [47] and SCP with normalized nonconformity (SCPN) [47] were selected as comparison algorithms. We utilized the proposed CA-LSTM as a single-point predictor. CP is a general framework for uncertainty quantification, and can be applied to an arbitrary user-chosen predictive algorithm.

Fig. 8 illustrates the single-point prediction, ground truth, and prediction intervals for DS services, with a significance level set to $\alpha = 0.1$. The prediction intervals from all three methods encompass the ground truth with a probability greater than the user-defined threshold. However, it is evident that SCP lacks adaptability, as the width of its prediction interval remains constant across all data points, as shown in Eq. (19). In contrast, ASCP and SCPN offer more adaptive and flexible prediction intervals, better capturing varying levels of prediction difficulty or uncertainty for different data points. For cases with less accurate predictions—especially those far from the

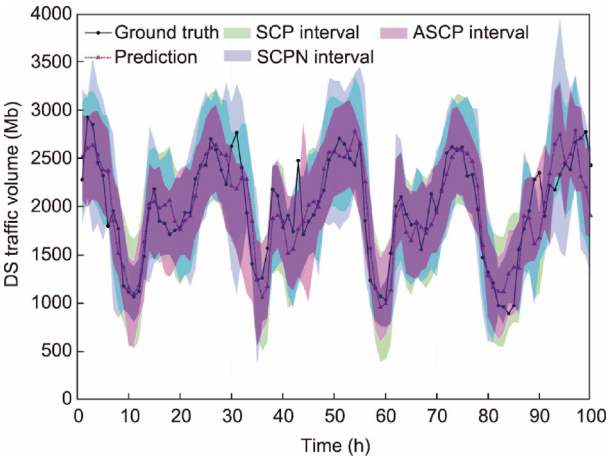

Fig. 8. Traffic prediction results of DS service on partial testing set.

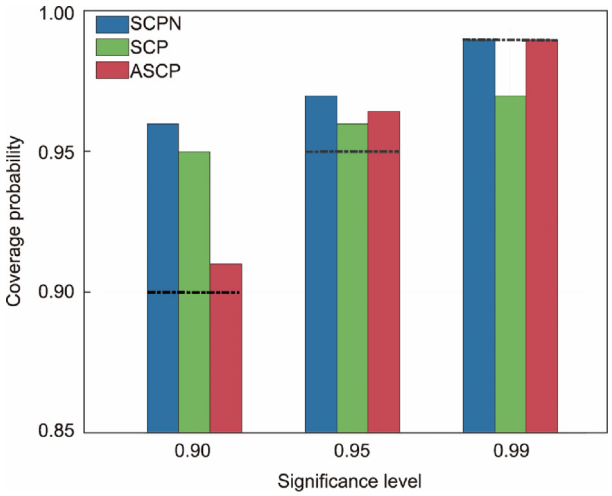

Fig. 10. Comparisons of coverage probability among different CP methods.

true values—the prediction intervals tend to be wider, reflecting weaker confidence in the predictions. Furthermore, the proposed algorithm achieves significantly narrower interval widths, providing stronger confidence in the prediction results.

The performance of CP is evaluated in terms of coverage and inefficiency. Coverage refers to the probability that the true values fall within the prediction interval, while inefficiency relates to the average width of the prediction intervals. The objective is to achieve a narrower prediction interval while maintaining the desired coverage level. Figs. 9 and 10 present comparisons of prediction interval width and coverage probability across different CP methods. The horizontal black dashed line in Fig. 10 indicates the user-specified confidence levels. As the desired confidence level increases, the prediction interval width also expands to ensure that the true value is included with a higher probability. It is evident that the proposed ASCP method consistently guarantees the coverage properties of prediction intervals across different significance levels. Furthermore, ASCP outperformed the other methods, providing significantly narrower interval widths without sacrificing coverage.

6.3. Resource scheduling effectiveness

Next, we evaluated the performance of the ORS scheme. In this study, price varies with time, set by NO as the ratio of the total resource demand of all SPs to the available network resources over the considered time period. In addition, the advance reservation price needs to be multiplied by the number of slots K and then a discount factor c, that is, $p_t = c\frac{\sum_m \sum_{k\in\mathcal{K}_t} E_m(k)}{\sum_m D_{\max}^{s,m}}$ and $q_k = \frac{\sum_m E_m(k)}{\sum_m D_{\max}^{s,m}}$. NO can encourage the SPs to reserve more resources in advance by offering higher discounts, which default to $c = 1$.

Figs. 11 and 12 show the simulation results for the time-averaged regret and QoS constraint violation, respectively. The online gradient descent (OGD) algorithm [48], a classical low-regret algorithm, was used for comparison. The two algorithms achieved sublinear growth in regret for the two types of services, demonstrating the overall effectiveness of the proposed algorithm. The proposed algorithm exhibited smaller regret values, indicating that it was closer to the global optimal solution provided in hindsight. After the calculation, the time-averaged regret values of the ORS for DT and DS services differ from the optimal solution by 0.6% and 1.3%, respectively. As shown in Fig. 12, the optimal solution obtained by convex optimization always satisfies the QoS constraints; therefore, its time-averaged constraint fit metrics remain at zero. Moreover, the constraint fit metrics for both algorithms approached zero over time, demonstrating the ORS's superior ability to satisfy the QoS constraints.

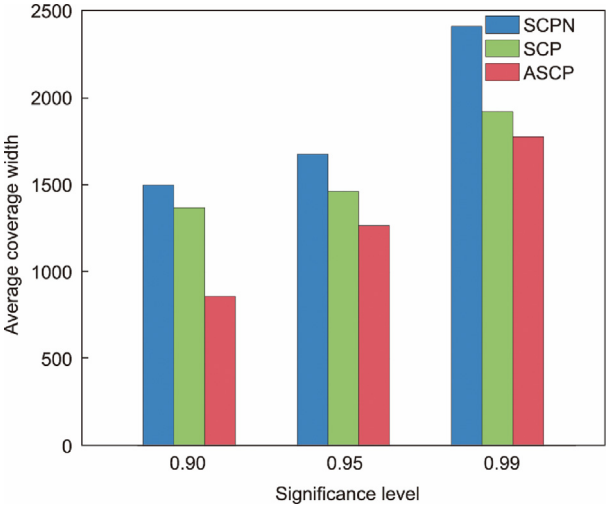

Fig. 9. Comparisons of prediction interval width among different CP methods.

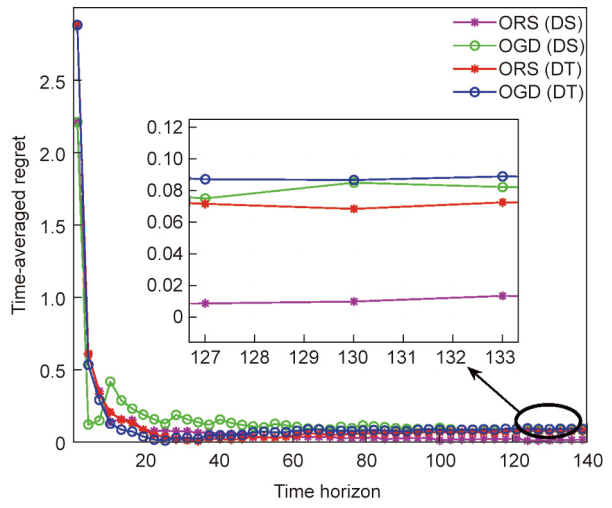

Fig. 11. Comparison of dynamic regret. OGD: online gradient descent.

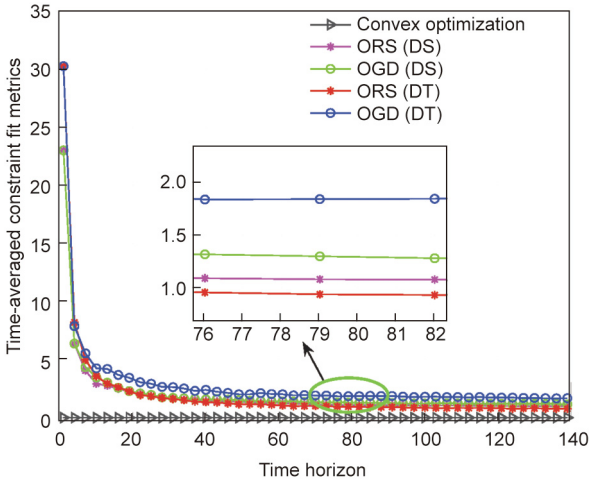

Fig. 12. Comparison of constraint fit metrics.

Next, we analyzed the role of the prediction module in the resource scheduling module using the DS service as an example. Fig. 13 illustrates the relationship between resource bounds and QoS constraint violations. The resource bound refers to the maximum proportion of resources that can be allocated to the SP. When the resource bound is relatively low, all algorithms struggle to guarantee user QoS. As the resource bound increases, the number of QoS violations decreases. Among the three algorithms, the CP-assisted ORS allocated more resources to accommodate uncertainty, consistently achieving the lowest constraint violations and demonstrating the strongest ability to meet QoS requirements. Furthermore, incorporating the prediction module improved resource scheduling optimization, leading to better QoS fulfillment compared to an ORS without prediction. From Fig. 14, it is clear that the higher the budget, the lower the degree of QoS violation. Similarly, the CP-assisted ORS exhibited the best performance in terms of QoS guarantee, whereas the prediction-assisted ORS outperformed the ORS without prediction.

Fig. 15 shows the relationship between the resource occupancy ratio and the resource price. Here, we consider the change in adjustment price q_k as an example. As the adjustment price increases, the reservation amount gradually increases and the adjustment amount decreases. It can be observed that an ORS with

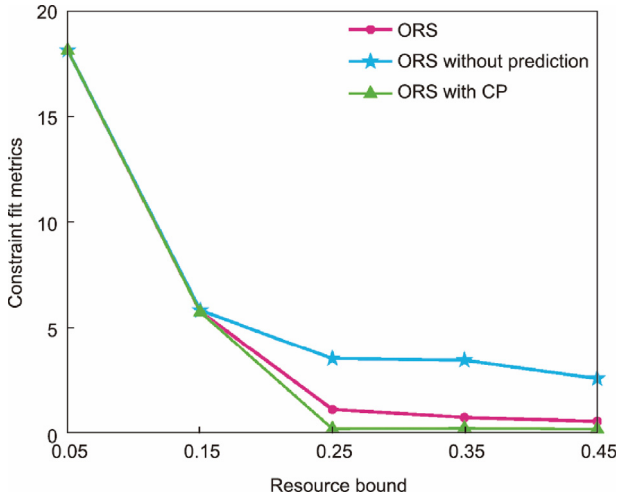

Fig. 13. Constraint fit metrics versus resource bound.

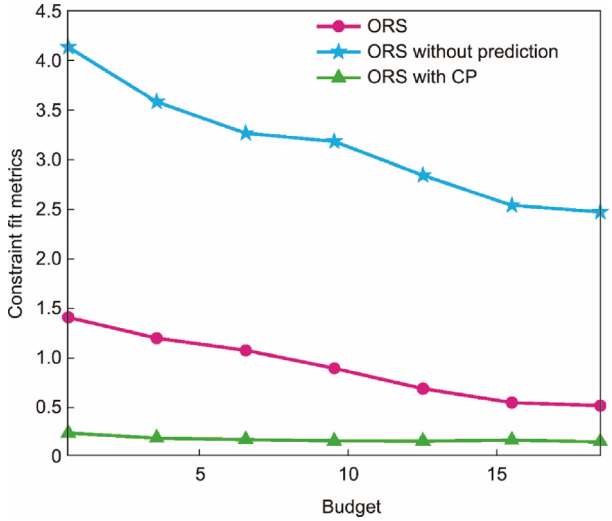

Fig. 14. Constraint fit metrics versus budget.

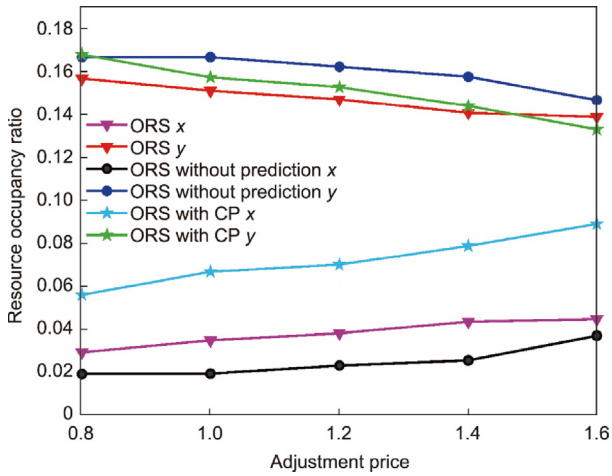

Fig. 15. Resource occupancy ratio versus price. *x*: resource reservations over a large timescale; *y*: resource adjustments over a small timescale.

CP consistently occupies more resources to counteract uncertainty. However, as shown in Figs. 13 and 14, this additional resource occupation is beneficial for QoS guarantees. Clearly, the NO must strike a balance between QoS provision and resource utilization. Moreover, because the ORS has some foresight into future conditions, it tends to prioritize reservation strategies over the ORS without prediction.

7. Conclusions

To address the QoS guarantee challenges in STINs, particularly in highly dynamic network environments and stochastic traffic fluctuations, we investigated an online resource-scheduling scheme assisted by service demand prediction. First, to characterize dynamic network resources, we implemented 3GPP NTN standard-based channel and antenna models on NS-3, ensuring accurate capture of the actual transmission characteristics of STINs. Next, for service demand prediction and uncertainty quantification, we propose a CA-LSTM architecture that combines 1D-Conv, LSTM, and attention mechanisms, enabling precise multidimensional traffic demand forecasting. To effectively manage uncertainties from burst traffic, we developed an adaptive CP algorithm, ASCP, for time-series data. This algorithm provides

statistically guaranteed prediction intervals, significantly enhancing prediction reliability. Building on this foundation, we introduced a dual-timescale resource scheduling framework for on-demand resource matching, which includes large-timescale resource reservation and small-timescale resource adjustment. We also introduced an ORS algorithm based on OCO, delivering long-term performance guarantees with limited network information. Experimental results on real-world datasets validated the accuracy and efficiency of our prediction algorithms: CA-LSTM achieved the smallest RMSE, and ASCP outperformed competitors with narrower interval widths and no loss of coverage. Furthermore, using our high-fidelity STIN simulation platform, the experimental results show that the proposed scheme achieves near-optimal performance, comparable to the optimal solution with global information. This approach significantly reduces the resource occupancy rate while ensuring QoS, thus achieving the co-optimization of resource utilization efficiency and service quality.

CRediT authorship contribution statement

Xiaohan Qin: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization. **Tianqi Zhang:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Resources, Methodology. **Kai Yu:** Methodology, Investigation, Data curation, Conceptualization. **Xin Zhang:** Writing – review & editing, Software, Investigation, Conceptualization. **Haibo Zhou:** Writing – review & editing, Supervision, Project administration, Funding acquisition. **Weihua Zhuang:** Writing – review & editing, Supervision, Resources, Project administration, Methodology, Funding acquisition, Formal analysis, Conceptualization. **Xuemin Shen:** Writing – review & editing, Supervision, Project administration, Methodology, Investigation, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by the Major Program of the National Natural Science Foundation of China (62495021 and 62495020).

References

[1] Zhou D, Sheng M, Li J, Han Z. Aerospace integrated networks innovation for empowering 6G: a survey and future challenges. IEEE Commun Surv Tutor 2023;25(2):975–1019.
[2] Xiao Z, Yang J, Mao T, Xu C, Zhang R, Han Z, et al. LEO satellite access network (LEO-SAN) towards 6G: challenges and approaches. IEEE Wirel Commun 2024;31(2):89–96.
[3] Pan G, Ye J, An J, Alouini MS. Latency versus reliability in LEO mega-constellations: terrestrial, aerial, or space relay? IEEE Trans Mobile Comput 2023;22(9):5330–45.
[4] Ma T, Qian B, Qin X, Liu X, Zhou H, Zhao L. Satellite–terrestrial integrated 6G: an ultra-dense LEO networking management architecture. IEEE Wirel Commun 2024;31(1):62–9.
[5] Cao X, Yang B, Shen Y, Yuen C, Zhang Y, Han Z, et al. Edge-assisted multi-layer offloading optimization of LEO satellite–terrestrial integrated networks. IEEE J Sel Areas Commun 2023;41(2):381–98.
[6] Qin X, Ma T, Tang Z, Zhang X, Zhou H, Zhao L. Service-aware resource orchestration in ultra-dense LEO satellite–terrestrial integrated 6G: a service function chain approach. IEEE Trans Wirel Commun 2023;22(9):6003–17.
[7] Xu Q, Su Z, Lu R, Yu S. Ubiquitous transmission service: hierarchical wireless data rate provisioning in space–air–ocean integrated networks. IEEE Trans Wirel Commun 2022;21(9):7821–36.
[8] Kawamoto Y, Takahashi M, Verma S, Kato N, Tsuji H, Miura A. Traffic prediction-based dynamic resource control strategy in HAPS-mounted MEC-assisted satellite communication systems. IEEE Internet Things J 2024;11(8):13824–36.
[9] Yuan S, Sun Y, Peng M. Joint network function placement and routing optimization in dynamic software-defined satellite–terrestrial integrated networks. IEEE Trans Wirel Commun 2024;23(5):5172–86.
[10] Li D, Liu X, Yin Z, Cheng N, Liu J. CWGAN-based channel modeling of convolutional autoencoder-aided SCMA for satellite–terrestrial communication. IEEE Internet Things J 2024;11(22):36775–85.
[11] Zhang X, Wang Y, Qin X, Zhang Z, Zhou H, Shen X. Link-level performance analysis of DVB standards in ultra-dense LEO satellite–terrestrial networks. In: Proceedings of IEEE 99th Vehicular Technology Conference; 2024 Jun 24–27; Singapore. New York City: IEEE; 2024.
[12] Yu J, Liu X, Gao Y, Shen X. 3D on and off-grid dynamic channel tracking for multiple UAVs and satellite communications. IEEE Trans Wirel Commun 2022;21(6):3587–604.
[13] Chen Z, Zhang J, Min G, Ning Z, Li J. Traffic-aware lightweight hierarchical offloading towards adaptive slicing-enabled SAGIN. IEEE J Sel Areas Commun 2024;42(12):3536–50.
[14] Cai Y, Cheng P, Chen Z, Ding M, Vucetic B, Li Y. Deep reinforcement learning for online resource allocation in network slicing. IEEE Trans Mobile Comput 2024;23(6):7099–116.
[15] Tu H, Zhao L, Zhang Y, Zheng G, Feng C, Song S, et al. Deep reinforcement learning for optimization of RAN slicing relying on control- and user-plane separation. IEEE Internet Things J 2024;11(5):8485–98.
[16] Cui Y, Huang X, He P, Wu D, Wang R. QoS guaranteed network slicing orchestration for Internet of Vehicles. IEEE Internet Things J 2022;9(16):15215–27.
[17] Li Q, Wang Y, Sun G, Luo L, Yu H. Joint demand forecasting and network slice pricing for profit maximization in network slicing. IEEE Trans Netw Sci Eng 2024;11(2):1496–509.
[18] Jiang W, Zhang Y, Han H, Huang Z, Li Q, Mu J. Mobile traffic prediction in consumer applications: a multimodal deep learning approach. IEEE Trans Consum Electron 2024;70(1):3425–35.
[19] Lin YH, Li GH. A Bayesian deep learning framework for RUL prediction incorporating uncertainty quantification and calibration. IEEE Trans Industr Inform 2022;18(10):7274–84.
[20] He M, Wu H, Zhou C, Hu S, Tang Z, Zhuang W. Digital twin-assisted robust and adaptive resource slicing in LEO satellite networks. In: Proceedings of IEEE Global Communications Conference; 2024 Dec 8–12; Cape Town, South Africa. New York City: IEEE; 2024. p. 3261–6.
[21] Sachdeva R, Gakhar R, Awasthi S, Singh K, Pandey A, Parihar AS. Uncertainty and noise aware decision making for autonomous vehicles: a Bayesian approach. IEEE Trans Veh Technol 2025;74(1):378–89.
[22] Aboeleneen AE, Abdellatif AA, Erbad AM, Salem AM. ECP: error-aware, cost-effective and proactive network slicing framework. IEEE Open J Commun Soc 2024;5:2567–84.
[23] Garrido LA, Dalgkitsis A, Ramantas K, Ksentini A, Verikoukis C. Resource demand prediction for network slices in 5G using ML enhanced with network models. IEEE Trans Veh Technol 2024;73(8):11848–61.
[24] Park S, Cohen KM, Simeone O. Few-shot calibration of set predictors via meta-learned cross-validation-based conformal prediction. IEEE Trans Pattern Anal Mach Intell 2024;46(1):280–91.
[25] Cohen KM, Park S, Simeone O, Shitz SS. Calibrating AI models for wireless communications via conformal prediction. IEEE Trans Mach Learn Commun Netw 2023;1:296–312.
[26] Piao S, Huang R, Tsung F. CRULP: reliable RUL estimation inspired by conformal prediction. IEEE Trans Instrum Meas 2024;74:3505411.
[27] Lai J, Liu H, Xu G, Jiang W, Wang X, Jiang D. Joint computation offloading and resource allocation for LEO satellite networks using hierarchical multi-agent reinforcement learning. IEEE Trans Cogn Commun Netw 2025;11(4):2554–67.
[28] Jiang W, Zhan Y, Fang X. Fuzzy neural network based access selection in satellite–terrestrial integrated networks. J Netw Comput Appl 2025;236:104108.
[29] Wang H, Bai Y, Xie X. Deep reinforcement learning based resource allocation in delay-tolerance-aware 5G industrial IoT systems. IEEE Trans Commun 2024;72(1):209–21.
[30] Hou Y, Zhang K, Liu X, Chuai G, Gao W, Chen X. An inter-slice RB leasing and association adjustment scheme in O-RAN. IEEE Trans Netw Serv Manag 2024;21(1):402–17.
[31] Huang T, Fang Z, Tang Q, Xie R, Chen T, Yu FR. Dual-timescales optimization of task scheduling and resource slicing in satellite–terrestrial edge computing networks. IEEE Trans Mobile Comput 2024;23(12):14111–26.
[32] Liu Y, Ma T, Qin X, Zhou H, Shen XS. Reconfigurable RAN slicing for ultra-dense LEO satellite networks via DRL. IEEE Trans Cogn Commun Netw 2025;11(1):566–80.
[33] Wang J, Dong M, Liang B, Boudreau G. Periodic updates for constrained OCO with application to large-scale multi-antenna systems. IEEE Trans Mobile Comput 2023;22(11):6705–22.
[34] Choi P, Ham D, Kim Y, Kwak J. VisionScaling: dynamic deep learning model and resource scaling in mobile vision applications. IEEE Internet Things J 2024;11(9):15523–39.
[35] Chouayakh A, Destounis A. Towards no regret with no service outages in online resource allocation for edge computing. In: Proceedings of IEEE

International Conference on Communications; 2022 May 16–20; Seoul, Republic of Korea. New York City: IEEE; 2023. p. 4378–83.
[36] TR 38.811: Study on new radio (NR) to support non-terrestrial networks. Report. Valbonne: 3rd Generation Partnership Project; 2020 Oct.
[37] Series P. Attenuation by atmospheric gases and related effects. Geneva: International Telecommunication Union Radiocommunication Sector; 2019.
[38] TR 38.901: Study on channel model for frequencies from 0.5 to 100 GHz. Report. Valbonne: 3rd Generation Partnership Project; 2024 Apr.
[39] Xue J, Yu K, Zhang T, Zhou H, Zhao L, Shen X. Cooperative deep reinforcement learning enabled power allocation for packet duplication URLLC in multi-connectivity vehicular networks. IEEE Trans Mobile Comput 2024;23(8):8143–57.
[40] Li Z, Liu F, Yang W, Peng S, Zhou J. A survey of convolutional neural networks: analysis, applications, and prospects. IEEE Trans Neural Netw Learn Syst 2022;33(12):6999–7019.
[41] Zhang Y, Xiong R, He H, Pecht MG. Long short-term memory recurrent neural network for remaining useful life prediction of lithium-ion batteries. IEEE Trans Veh Technol 2018;67(7):5695–705.
[42] Du W, Wang Y, Qiao Y. Recurrent spatial–temporal attention network for action recognition in videos. IEEE Trans Image Process 2018;27(3):1347–60.
[43] Xu C, Xie Y. Sequential predictive conformal inference for time series. In: Proceedings of the 40th International Conference on Machine Learning, 2023 Jul 23–29; Honolulu, HI, UAS. New York City: ML Research Press; 2023. p. 38707–27.
[44] Liu X, Ma T, Tang Z, Qin X, Zhou H, Shen XS. UltraStar: a lightweight simulator of ultra-dense LEO satellite constellation networking for 6G. IEEE/CAA J Autom Sinica 2023;10(3):632–45.
[45] Kassing S, Bhattacherjee D, Águas AB, Saethre JE, Singla A. Exploring the "Internet from space" with Hypatia. In: Proceedings of the ACM Internet Measurement Conference; 2020 Oct 27–29; online. New York City: Association for Computing Machinery; 2020. p. 214–29.
[46] Chen M, Miao Y, Gharavi H, Hu L, Humar I. Intelligent traffic adaptive resource allocation for edge computing-based 5G networks. IEEE Trans Cogn Commun Netw 2020;6(2):499–508.
[47] Xu W, Zio E. A general data-driven framework for remaining useful life estimation with uncertainty quantification using split conformal prediction. In: Proceedings of the 7th International Conference on System Reliability and Safety; 2023 Nov 22–24; Bologna, Italy. New York City: IEEE; 2023. p. 67–74.
[48] Monteil JB, Iosifidis G, DaSilva LA. Learning-based reservation of virtualized network resources. IEEE Trans Netw Serv Manag 2022;19(3):2001–16.