An Efficient and Stable Knowledge Service Framework for Satellite-Ground Collaboration

Fei Teng, Hao Zhao, *Member, IEEE*, Qiyang Zhang, *Member, IEEE*,
Ranga Rao Venkatesha Prasad, *Senior Member, IEEE*, and Schahram Dustdar, *Fellow, IEEE*

***Abstract*—The rapid expansion of Low Earth Orbit (LEO) satellite constellations presents immense potential for in-orbit services. However, the large-scale and dynamic nature of LEO constellations creates unstable communication environments, where traditional methods struggle to ensure the efficiency and stability of onboard inference services. This highlights the need for an advanced knowledge service framework capable of both inference and root cause analysis of service disruptions. To address this, we propose a novel knowledge service framework that integrates data-driven and knowledge-driven models through satellite-ground collaboration. The framework leverages lightweight onboard models for real-time data processing and ground-based knowledge graphs for advanced inference and cause analysis. To further enhance stability within complex interconnected onboard systems, we propose a prediction-based algorithm for LEO satellite networks that uses joint spatio-temporal modeling to achieve accurate link prediction. Additionally, we formulate an optimization problem aimed at minimizing path distance variance and maximizing path stability across LEO topologies, and we propose a heuristic path selection strategy to ensure efficient inter-satellite routing. Extensive in-orbit deployments and simulation experiments demonstrate the feasibility and effectiveness of the proposed framework. Satellite-ground verification on the BUPT-1 satellite shows its ability to provide real-time services, while inter-satellite simulations using real constellation data indicate significant improvements in response latency and path stability. Compared with baseline methods, our proposed method significantly reduces path jitter by up to 62.6% and improves path availability by up to 17.3% across various LEO constellations.**

***Index Terms*—Knowledge service framework, LEO satellite constellation, path selection optimization, satellite-ground collaboration.**

Received 6 March 2025; revised 6 August 2025; accepted 3 September 2025. Date of publication 9 September 2025; date of current version 11 December 2025. This work was supported in part by National Natural Science Foundation of China under Grant 62272398, in part by Sichuan Science and Technology Program under Grant 2024NSFJQ0019, and in part by Postdoctoral Funding Program under Grant 2025M771581 and Grant GZC20251097. An earlier version of this paper was presented in part at the IEEE International Conference on Web Services [DOI: 10.1109/ICWS60048.2023.00085]. *(Corresponding author: Qiyang Zhang.)*

Fei Teng and Hao Zhao are with the School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu 611756, China, and also with the Engineering Research Center of Sustainable Urban Intelligent Transportation, Ministry of Education, Chengdu 611756, China (e-mail: fteng@swjtu.edu.cn; andyzhao@my.swjtu.edu.cn).

Qiyang Zhang is with Computer Science School, Peking University, Beijing 100876, China (e-mail: qiyangzhang@pku.edu.cn).

Ranga Rao Venkatesha Prasad is with the Networked Systems (NS), Delft University of Technology (TU Delft), 2629 HS Delft, The Netherlands (e-mail: R.R.VenkateshaPrasad@tudelft.nl).

Schahram Dustdar is with the head of the Distributed Systems Group at the TU Wien, 1040 Vienna, Austria, and also with ICREA, 08002, Barcelona, Spain (e-mail: dustdar@dsg.tuwien.ac.at).

Digital Object Identifier 10.1109/TSC.2025.3607909

I. Introduction

Recently, the rapid expansion of Low Earth Orbit (LEO) satellite constellations has gained significant attention. Companies such as Amazon and SpaceX [1] have committed to deploying hundreds of small cubesats in LEO to achieve global coverage. Currently, approximately 45% of LEO satellites are dedicated to Earth observation [2], operating at altitudes of around 500 km to collect high-resolution images and transmit data to ground stations. However, satellite-ground communication faces significant challenges, including limited bandwidth and frequent disruptions between satellite and ground [3][4], leading to substantial data backlogs. Moreover, the visibility window of these satellites is typically around 10 min per pass, and the link reliability can fluctuate depending on the satellites' elevation.

To address these limitations, startups like OrbitsEdge and Loft Orbital have introduced the concept of "space infrastructure as a service" by deploying satellites equipped with commercial off-the-shelf (COTS) rack servers .[1] Orbital edge computing [5], [6] has consequently emerged as a complementary paradigm to ground-based computing, enabling onboard computation and real-time data processing.

Recent research has focused on deploying lightweight deep learning models in LEO for real-time applications such as object classification, disaster prediction, and fault diagnosis [7], [8]. However, the limited computational and storage capacities of satellites and the scarcity of onboard training data pose significant challenges for purely data-driven models, often resulting in inaccurate detection of unexpected or rare faults [9]. Existing methods can identify affected services but often fail to reveal the root causes of faults. Purely data-driven models deployed onboard satellites are constrained by limited training data and computational resources, which limit their ability to diagnose unexpected or rare faults accurately. For instance, in satellite power systems, a sudden voltage drop [10] in a solar array may not indicate a fault in the array itself but could be caused by a malfunction in the power distribution unit or a communication error onboard computer. Such scenarios highlight the need for an advanced knowledge service (KS) framework capable of inferring the underlying causes of fault, especially in complex interconnected onboard systems. By integrating data-driven inference with a knowledge graph, the proposed KS framework

[1] https://datacenterfrontier.com/data-centers-above-the-clouds-colocation-goes-to-space/

can better capture system inter-dependencies and reduce misclassifications, thus providing more reliable fault analysis.

Motivated by the strengths of Knowledge Graphs (KGs) in improving data integration, inference, and decision-making across complex domains [11], we propose an efficient knowledge service framework that combines data-driven and knowledge-driven models in a satellite-ground collaborative system. This framework ensures accurate service discovery by coordinating real-time onboard models with expert knowledge-based inference at ground stations. Unlike existing satellite-ground collaborative schemes, our framework offers two key advantages: This framework constructs a comprehensive knowledge graph; The framework employs lightweight models on satellites for real-time service detection while leveraging ground-based knowledge graphs, thus addressing the challenges of limited communication resources and dynamic bandwidth variations. In-orbit deployment and verification experiments on the BUPT-1 satellite demonstrate the feasibility of the proposed framework, particularly in satellite-ground collaboration.

Meanwhile, establishing efficient and stable knowledge service paths within LEO networks is crucial to the KS framework. For example, such paths ensure seamless data transmission, support real-time decision-making, and enhance the reliability of satellite-ground collaboration in dynamic network environments. However, determining the optimal path strategy in dynamic LEO networks remains a significant challenge. Specifically, two key issues must be addressed:

(1) Accurately estimating links in LEO satellite networks is inherently challenging due to the high speeds of satellites and the dynamic nature of the network topology. Although the topology of LEO networks can be relatively straightforward to estimate, the dynamic and unpredictable communication conditions (e.g., interference, signal attenuation, and fluctuating link quality) make it difficult to obtain accurate link information in real time.

(2) Selecting the optimal path strategy among satellites is non-trivial. Different path selection strategies directly affect the performance of knowledge service (e.g., latency). The presence of numerous branching structures within LEO constellations leads to exponential growth in the number of potential path strategies. This exponential growth presents a substantial challenge in finding the optimal path strategy.

To address the above challenges, we develop a prediction-based algorithm for LEO constellations that leverages joint spatio-temporal modeling to capture the dynamic dependencies of satellite networks. Specifically, the algorithm integrates Graph Attention Network (GAT) [12]-based spatial feature extraction and Long Short-Term Memory (LSTM)-based temporal dynamics modeling to achieve more accurate link prediction. Furthermore, to determine the optimal path selection strategy, we formulate an optimization problem to minimize the path variance and reliability across LEO topologies, enabling efficient inter-satellite path selection. This problem is an integer linear programming problem, making it impossible to solve in polynomial time. To solve this problem efficiently, we propose a heuristic path selection strategy to reduce complexity and ensure path stability across various LEO satellite architectures. To evaluate the effectiveness of the proposed algorithm, we conduct extensive experiments using parameters derived from actual satellite constellations. The results demonstrate that the proposed algorithm reduces response time by accurately identifying the path selection strategy.

Our main contributions are summarized as follows:

- We present an efficient knowledge service framework designed to enhance service discovery in satellite-ground collaborative systems.
- We propose a prediction-based algorithm for satellite networks, leveraging joint spatio-temporal modeling to achieve accurate link prediction. Furthermore, we formulate the inter-satellite path selection problem, which presents high complexity. We also propose a heuristic-based path selection strategy to ensure path stability.
- We conduct extensive performance analysis using parameters derived from real-world satellite constellations, demonstrating that the proposed framework significantly reduces response latency and improves path stability compared to baseline methods.
- We deploy the proposed framework on the BUPT-1 satellite, verifying the feasibility of this framework and demonstrating its potential to improve the operation of LEO satellite constellations.

The remainder of this paper is organized as follows: Section II details the proposed framework, Section III presents the ISL optimization algorithm, Section IV and Section V discuss the experimental evaluation and satellite-ground verification, Section VI reviews related work, and Section VII discusses and concludes the paper.

II. Knowledge Service Framework

A. Overview

This section overviews the software and hardware architecture [13], as illustrated in Fig. 1. The KS framework is designed to support multi-scenario KS based on a satellite-ground collaborative system. We first present an overview of the overall system before addressing specific challenges. The framework consists of three main parts: the satellite, the LEO network, and the ground station.

The satellite architecture is divided into three layers: the platform, the onboard edge, and the application. The platform layer includes computing units of the satellite, which provide services for communication, payload, command and data handling. This setup allows for the local processing and filtering of telemetry data, especially when downlink capacity is limited. The edge layer ensures reliable onboard computing under harsh space conditions and enables efficient collaboration within the satellite network. The application layer enables advanced functions that are supported by ground-based KS.

The ground station architecture consists of three main components: the ground infrastructure, the KS layer, and ground applications. The ground infrastructure supplies all cloud computing resources necessary to support the satellite-dependent components, employing containerization technologies to run services and components in isolated environments. The KS layer delivers comprehensive KS, performing knowledge modeling

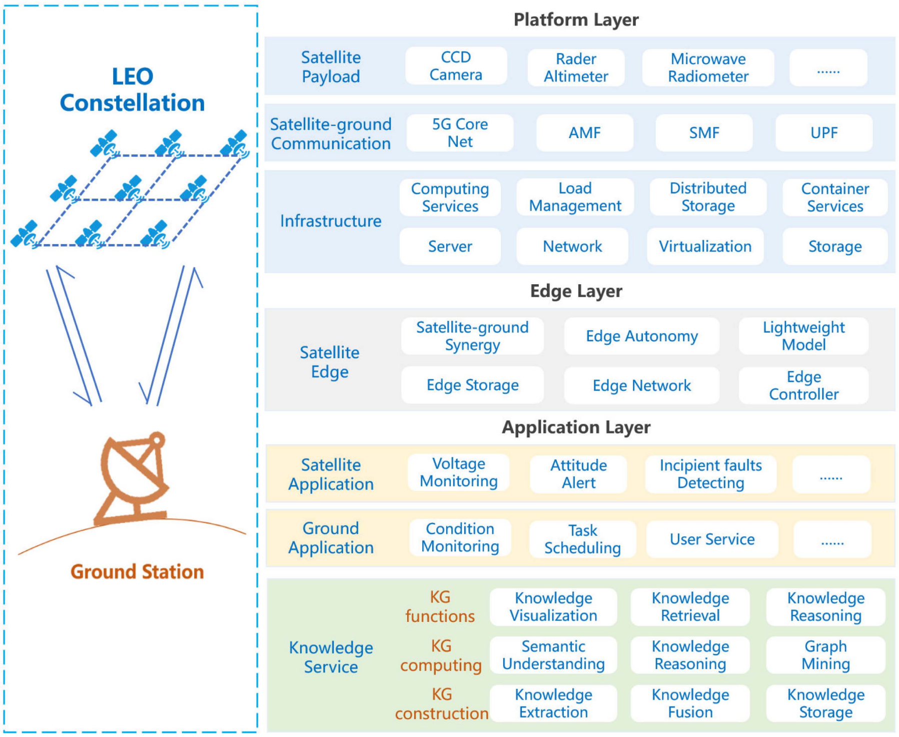


Fig. 1. Framework of the satellite-ground collaborative knowledge service.

and management tailored to various scenarios. Based on these layers, the ground application layer provides diverse intelligent functions. The ground-based KS component is the key part of the framework. It manages KGs, enabling knowledge extraction, fusion, and storage. Knowledge computing uses algorithms to discover explicit and implicit knowledge, identify patterns, and create rules that support semantic understanding and graph mining. Additionally, it offers user-friendly interfaces for visualization and information retrieval.

Satellite-ground Communication mainly uses the protocols specified by Consultative Committee for Space Data Systems (CCSDS). The CCSDS is an international organization that develops standardized protocols for space data systems, ensuring interoperability and efficient communication between satellites and ground stations. CCSDS protocols are widely adopted in space tasks due to their robustness, scalability, and ability to handle the unique challenges of space communication, such as long distances, signal delays, and limited bandwidth. The use of 5 G Non-Terrestrial Network technology allows end-to-end communication without the need for additional ground infrastructure. This optimizes the use of onboard networking, computation, storage resources and provides KS more effectively.

TABLE I
ENTITY TYPES IN THE KNOWLEDGE GRAPH

Entity	Example
Fault mode	Abnormal power off
Unit	Executing unit
Component	Momentum wheel set
Fault phenomenon	Fast breaking
Fault cause	Overcurrent protection
Fault location	Momentum wheel
Measure	Check circuits

TABLE II
RELATIONSHIP TYPES IN THE KNOWLEDGE GRAPH

Relation	Example
belong	Momentum-wheel set belongs to Executing unit
include	Momentum-wheel set includes Momentum wheel
occur	Abnormal power off occurs in Momentum wheel
express	Overcurrent protection expresses Fast breaking
lead	Overcurrent protection leads to Abnormal power off
apply	Check circuits applies to Overcurrent protection
locate	Fast breaking is located in Momentum wheel

B. Knowledge Graph Construction

A structured and semantically rich representation of knowledge is essential for advanced inference, decision-making, and intelligent applications. The knowledge graph is the core of the framework, providing a unified way to manage and utilize knowledge. By organizing information as connected semantic triples (subject-predicate-object relationships, e.g., “Satellite A - has a fault - Solar Array”), the knowledge graph captures complex relationships, integrates heterogeneous data sources, and supports scalable and flexible knowledge-based services. Its capability to infer relationships from limited data and manage ambiguous or incomplete information makes it a powerful tool for knowledge representation and inference in complex systems.

The schema of the knowledge graph is defined with multiple entities and relationships, as shown in Tables I and II. Entities are represented as nodes, while relationships are stored in triple format. Standardizing entity names simplifies knowledge fusion across different data sources, enabling the extraction of attributes such as event types, timestamps, and corresponding actions. The

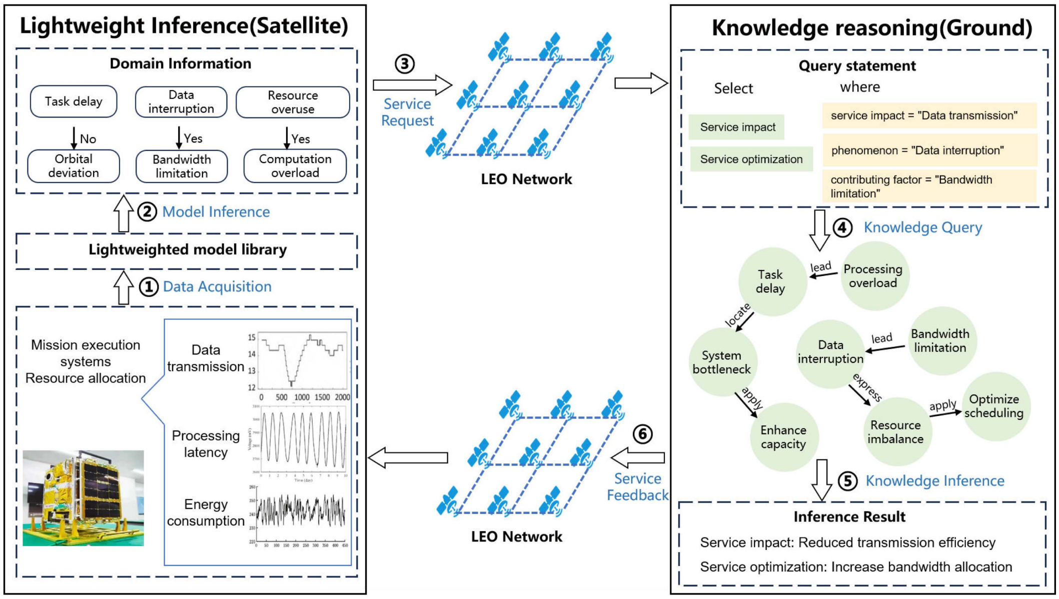


Fig. 2. Pipeline of knowledge service framework.

framework facilitates entity alignment and inference tasks by integrating diverse knowledge sources.

C. Knowledge Service Pipeline

The KS pipeline comprises six steps between satellites and the ground station, as illustrated in Fig. 2:

① *Data Acquisition:* The satellite employs onboard sensors to collect telemetry data, such as motor voltage, system voltage, and friction torque, to monitor subsystem status.

② *Model Inference:* The satellite utilizes lightweight onboard models to analyze the collected data and identify potential requests.

③ *Service Request:* The satellite initiates a service request by periodically calculating routes to establish Inter-satellite links (ISLs). It then transmits the requests and associated operational data.

④ *Knowledge Query:* Based on the transmitted data, the central controller formulates query statements to retrieve relevant knowledge from the knowledge graph.

⑤ *Knowledge Inference:* The ground station queries the knowledge graph to derive actionable insights, such as potential causes and mitigation recommendations.

⑥ *Service Feedback:* If the recommendations are adopted, the ground station transmits control instructions back to the satellite through the satellite-ground and ISLs so that the satellite can implement the suggested measures.

D. Service Response Time

Based on the KS pipeline, the target satellite initiates communication with the ground station to execute tasks. As illustrated in Fig. 3, the satellite establishes two types of communication links: ISLs and up-down links (UDLs). For instance, satellite SAT1 communicates with SAT5 via ISLs, while SAT5 communicates with the ground station through UDLs. When the ground station moves out of SAT5's coverage area, UDLs seamlessly switches to SAT3 to maintain connectivity. The total latency is computed as

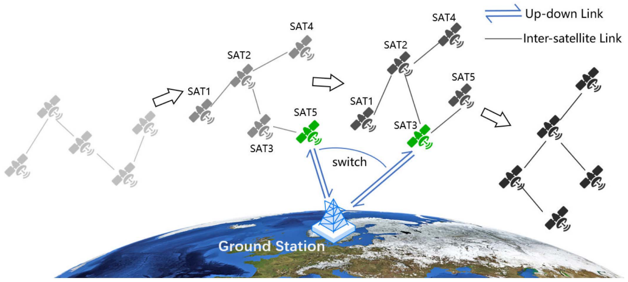


Fig. 3. Communication between satellites and ground stations under a dynamic network topology.

$$T_{total} = T_{isl1} + T_{dc} + T_{kc} + T_{ul} + T_{isl2}, \tag{1}$$

where T_{isl1} represents the inter-satellite communication time from the target satellite to the access satellite, T_{dc} denotes the downlink communication time from the access satellite to the ground station, T_{kc} signifies the knowledge computing time at the ground station, T_{ul} indicates the uplink communication time from the ground station to the access satellite, and T_{isl2} is the communication time from the access satellite back to the target satellite.

III. Efficient and Stable Inter-Satellite Link Optimization

This section formulates the ISL optimization problem under dynamic satellite topologies, introduces a spatio-temporal link prediction framework, and presents a Pareto-based evolutionary algorithm for optimal path selection. Theoretical analysis of computational complexity is also provided.

A. Problem Description

ISLs play a critical role in the overall KS process, as their performance directly determines data transmission efficiency and total service response time. Ensuring low latency and stable communication within the KS pipeline is therefore essential for optimal system performance. ISL latency times (i.e., T_{isl1} and T_{isl2}) are largely determined by the scale and topology of the satellite constellation. Although ISLs can support data transmission rates of several hundred Gbps, propagation delay remains the main factor influencing communication latency, particularly for small data packets.

Due to the rapid movement of large-scale LEO constellations, all satellites operate within a highly dynamic network topology. Consequently, the neighboring satellites that maintain ISLs change over time as link visibility varies. As depicted in Fig. 3, this dynamic environment highlights the critical importance of selecting appropriate neighboring satellites to establish reliable communication links. Optimizing ISL configurations is thus essential to ensure continuous communication paths and to maintain the integrity of the KS framework. By strategically selecting the most suitable neighboring satellites for link establishment, the system ensures timely knowledge inference and feedback, thereby improving overall network performance.

B. Spatio-Temporal Prediction Mechanism

To calculate the inter-satellite propagation delay, we first identify an access satellite, which is defined as the satellite with the longest communication time to the ground station. The communication time between a LEO satellite and the ground station is given by

$$T_{dc} = \frac{L_{dc}}{v_s}, \tag{2}$$

where v_s is the orbital speed of the LEO satellite, and L_{dc} is the arc length over which the ground station can communicate with the satellite, calculated as

$$L_{dc} = 2 \cdot (R_e + h) \cdot \alpha, \tag{3}$$

where α is the geocentric angle corresponding to the satellite's coverage area, h is the orbital altitude above the ground station, R_e denotes the radius of the Earth.

In this work, the ISL topology adopts the +Grid configuration [14], in which each satellite connects to two neighboring satellites in the same orbital plane and two in adjacent orbital planes. This configuration allows each satellite to maintain point-to-point communication with up to four other satellites simultaneously. The dynamic satellite network is modeled as a spatio-temporal graph $\mathcal{G}_t = \{\mathcal{V}_t, \mathcal{E}_t\}$, where $\mathcal{V}_t$ represents satellite nodes and $\mathcal{E}_t$ denotes links between satellites at time t. Although connections between LEO satellites typically change every few minutes, the overall constellation topology can be assumed static within a single time interval.

Our primary objective is to predict the future state of the graph $\mathcal{G}_{t+1}, \mathcal{G}_{t+2}, \ldots, \mathcal{G}_{t+T}$ based on historical ISL information within the planning horizon T, specifically determining whether a link e_{ij} exists between satellite nodes i and j at each future time step. Since the orbits and trajectories of LEO constellations with ISLs are generally predefined and regular to ensure stability and reliability, these spatio-temporal characteristics can be exploited to enhance the forecasting capability of the proposed algorithm.

TABLE III
LIST OF MAIN NOTATIONS

Parameter	Description
T	The planning horizon
P_{size}	Population size of candidate paths
G	Number of algorithm generations
$L_{\mathcal{P}}$	Average number of hops in a candidate path
λ_1, λ_2	The coefficients of the optimization
δ	Confidence threshold for link prediction
H_t	The feature matrices at time t
$\mathcal{P}$	The candidate path
$d_{t+k}(\mathcal{P})$	The total length of the candidate path $\mathcal{P}$
F	The overall optimization
$f_1(\mathcal{P})$	The path length variation of the candidate path $\mathcal{P}$
$f_2(\mathcal{P})$	The path stability of the candidate path $\mathcal{P}$

1) Spatio-Temporal Mixing: To address the challenges posed by frequent topology changes, we propose a Spatio-Temporal Mixing (STM) mechanism. STM mechanism integrates multiple types of spatio-temporal features (such as orbital parameters and historical link states) to improve the accuracy of predicting satellite service requirements. Specifically designed for spatio-temporal data, this mechanism effectively captures and forecasts the dynamic evolution of network topologies. In the context of link prediction, STM mechanism evaluates the likelihood that a valid link exists between any two satellites. A link is considered valid if its predicted probability exceeds a predefined confidence threshold, ensuring the system remains remains robust to frequent topology changes and maintains reliable communication as satellites move.

The accuracy of link prediction depends on both spatial and temporal factors. Spatial factors include orbital parameters such as altitude, inclination, and relative position to Earth and the Sun. Temporal factors, such as the rotation of Earth and user requirement variations, can cause periodic changes in link availability. To fully capture these dynamics, our STM mechanism jointly models spatial and temporal influences. Some main notations and related descriptions are summarized in Table III.

The STM Network integrates GAT-based spatial feature extraction with LSTM-based temporal dynamics modeling. As illustrated in Fig. 4, the workflow comprises the following key steps:

① *Spatial Feature Extraction:* Satellite node embeddings $\mathbf{H}'_t$ are updated using GAT to capture spatial dependencies within the constellation.

② *Temporal Dependency Modeling:* The spatially-enhanced embeddings $\mathbf{H}'_t$ are processed by an LSTM network to model sequential dynamics, capturing temporal correlations over time.

③ *Link Prediction:* A decoder estimates the probability of each link $e_{ij} \in \mathcal{E}_{t+k}$ ($k \in \{1, 2, \ldots, T\}$) based on the embedding structures processed $p(e_{ij}^{(t+k)}) = \sigma(\text{MLP}(\mathbf{H}''_t(i) \parallel \mathbf{H}''_t(j)))$, where Multilayer Perceptron (MLP) is a fully connected feedforward neural network that transforms input features through multiple nonlinear layers, and $\parallel$ represents the concatenation operator.

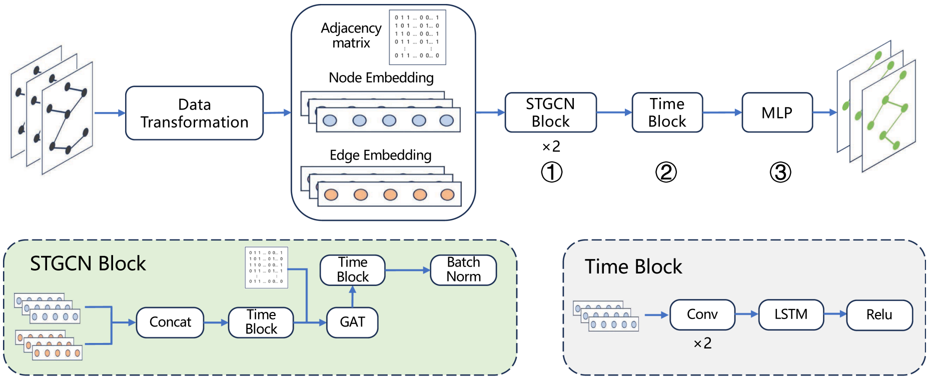


Fig. 4. Architecture of the STM Mechanism.

④ *Link Availability Determination:* The link e_{ij} between satellite i and j at time $t+k$ is considered available if the predicted probability satisfies the threshold condition, i.e., $a_{ij}^{(t+k)} = 1$ if $p(e_{ij}^{(t+k)}) \geq \delta$, and $a_{ij}^{(t+k)} = 0$ otherwise.

2) Spatial Feature Modeling: Graph Attention Networks: In LEO constellations, accurately modeling spatial dependencies is crucial for reliable link prediction. Standard Graph Convolution Network (GCNs) aggregate information from neighboring nodes to learn local structural patterns [15]. The feature matrix after GCN processing is given by

$$\mathbf{H}_t' = \sigma\left(\tilde{A}\mathbf{H}_t W\right), \tag{4}$$

where $\mathbf{H}_t$ represents the raw feature matrix (e.g., 3D position, semi-major axis, orbital inclination) for all satellites at time t; $\tilde{A}$ is the binary adjacency matrix encoding connectivity (1 for connected, 0 otherwise); W is a learnable weight matrix that transforms the input features into a new feature space; and $\sigma(\cdot)$ is a non-linear activation function.

While GCNs local topological structure effectively, their fixed aggregation weights limit their ability to adapt to the dynamic conditions typical of LEO networks. To overcome this, we adopt a GAT-based approach, which dynamically adjusts the aggregation weights through an attention mechanism, allowing flexible assignment of different weights to the ISLs. This mechanism effectively accommodates dynamic variations in LEO satellite network. Specifically, each attention coefficient α_{ij} quantifies the relevance of neighboring satellite v_j to v_i. By GAT, we redesign the feature matrix of satellite as

$$\mathbf{H}_t'(i) = \sigma\left(\sum_{j\in\mathcal{N}(v_i)} \alpha_{ij}\mathbf{W}_t(j)\right), \tag{5}$$

$$\alpha_{ij} = \text{softmax}_j\left(\text{LeakyReLU}\left(\mathbf{a}^T\left[\mathbf{W}_t(i) \,\|\, \mathbf{W}_t(j)\right]\right)\right), \tag{6}$$

where $\mathbf{a}$ is a learnable vector and $\|$ denotes concatenation.

This adaptive attention ensures that more relevant neighbors exert greater influence during feature aggregation, while less relevant nodes are suppressed. These weights are computed in real time based on the relevance of neighboring nodes, ensuring that significant neighbors have a greater influence while irrelevant ones are suppressed.[2]

3) Temporal Feature Modeling: LSTM-Based Temporal Encoding: To capture the temporal evolution of ISLs and periodic orbital patterns, we employ an LSTM network [16], which captures complex temporal dependencies and long-term trends in time-series data, which are essential for predicting future demand based on historical usage. The LSTM processes sequential feature matrices and learns the temporal dependencies, which is denoted by:

$$\mathbf{H}_t'' = \text{LSTM}\left(\mathbf{H}_{t-1}'', \mathbf{H}_t'\right), \tag{7}$$

where $\mathbf{H}_t'$ is the spatial feature at time t, and $\mathbf{H}_{t-1}''$ is the temporal feature from the previous time step $t-1$.

The LSTM network processes the sequential input $\{\mathbf{H}_1', \mathbf{H}_2', \ldots, \mathbf{H}_k', \ldots, \mathbf{H}_T'\}$ over time, enabling the model to learn the temporal dependencies that are crucial to predict link quality. By iteratively encoding historical states and current observations, the LSTM network generates temporal embeddings that capture both short-term dynamics (e.g., link connection and disconnection) and long-term patterns (e.g., orbital periodicity). These temporal embeddings $\mathbf{H}_t''$ are then combined with spatial features extracted by the GAT to form a unified representation, thus significantly improving the accuracy of link prediction. This enables the proposed optimizer to construct future-valid paths by leveraging probabilistic link forecasts within the planning horizon T. By integrating LSTM-based temporal modeling with GAT-based spatial modeling, the framework effectively captures dynamic changes and static dependencies in LEO satellite constellations.

C. Optimal Path Selection Policy

In LEO constellations, the establishment of stable paths is essential for constructing a highly reliable KS framework. The dynamic nature of satellite networks, driven by frequent topology changes due to satellite movement and environmental

[2] In this paper, we assume all satellites are identical; however, the analysis is valid even if the satellites are heterogeneous. In such cases, weights can be appropriately selected based on factors such as power, communication bandwidth, etc., to define the importance of neighboring nodes.

variations, necessitates adaptive routing strategies. Traditional static routing methods often perform suboptimally under such conditions. By leveraging the predicted availability of links across the planning horizon, the algorithm identifies candidate communication paths to construct optimal routes, ensuring that routing adapts to the continuously changing network topology. We formulate a multi-objective optimization problem to simultaneously minimize path distance variation and maximize path stability, and propose an NSGA-II based evolutionary algorithm to efficiently solve it.

1) Problem Formulation: To minimize the variation of the path distance and maximize the stability of the path within the planning horizon T, we formulate a multi-objective optimization problem to identify the optimal set of links $\mathcal{P}^*$. The objective function for path length variation is defined as follows:

$$f_1(\mathcal{P}) = \frac{1}{T}\sum_{k=1}^{T}\left(d_{t+k}(\mathcal{P}) - \bar{d}(\mathcal{P})\right)^2, \tag{8}$$

where $d_{t+k}(\mathcal{P})$ represents the total length of the candidate path $\mathcal{P}$ in topology snapshot at time $t+k$, and $\bar{d}(\mathcal{P})$ is the mean path length across all time steps within the planning horizon, given by

$$\bar{d}(\mathcal{P}) = \frac{1}{T}\sum_{k=1}^{T} d_{t+k}(\mathcal{P}). \tag{9}$$

The second objective ensures that the links that form the path remain active across all topologies, thereby minimizing the likelihood of disconnections. The objective function for path stability is given by:

$$f_2(\mathcal{P}) = -\frac{1}{|\mathcal{P}|}\frac{1}{T}\sum_{e_{ij}\in\mathcal{P}}\sum_{k=1}^{T} a_{ij}^{(t+k)}, \tag{10}$$

where $\mathcal{P}$ is the set of links that form the path, and $a_{ij}^{(t+k)}$ is the binary availability variable for link e_{ij} at time $t+k$ determined by the STM mechanism (1 if available based on $p(e_{ij}^{(t+k)}) \geq \delta$, 0 otherwise).

The overall optimization combines these two objectives, weighted by coefficients λ_1 and λ_2, which can be expressed as

$$\min_{\mathcal{P}} F = \lambda_1 f_1(\mathcal{P}) + \lambda_2 f_2(\mathcal{P}). \tag{11}$$

The goal is to find the optimal path $\mathcal{P}^*$ that minimizes this combined objective, subject to the following two constraints:

1) *Connectivity Constraint:* The selected links must maintain a valid path between the source and destination satellite nodes across all time steps in the planning horizon.
2) *Path Length Constraint:* The total path length $L_{\mathcal{P}}$ must lie within an acceptable range, i.e., $L_{\mathcal{P}} \in [L_{\min}, L_{\max}]$, where $L_{\min}$ and $L_{\max}$ define the allowed length bounds respectively.

2) Analysis and Solution: The optimization problem is formulated as a 0-1 integer linear programming (ILP) problem, which is NP-Hard due to the exponential growth of potential path combinations in LEO constellations. To address this complexity, we adopt the NSGA-II algorithm [17], a well-established heuristic for multi-objective optimization. We adapt this approach by embedding the weighted sum of objectives into a nonlinear fitness function for evaluation, while selection remains governed by non-dominated ranking and crowding distance.

Algorithm 1: Optimal Path Selection Algorithm.

```
Input: Set of temporal topology graphs $\{\mathcal{G}^{(t)}\}_{t=1}^{T}$;
       source satellite $s$; destination satellite $d$;
       maximum generations $G$; population size
       $P_{size}$; weights $\lambda_1, \lambda_2$
Output: Optimal path $\mathcal{P}^*$
Initialize $\mathcal{P}_g$ with $P_{size}$ individuals
for $gen = 1$ to $G$
    foreach $\mathcal{P}_i \in \mathcal{P}_g$ do
        Compute $f_1(\mathcal{P}_i)$, $f_2(\mathcal{P}_i)$
        Compute the fitness $F$ for reporting
    $\mathcal{R}_g \leftarrow \mathcal{P}_g$
    Perform non-dominated sorting on $\mathcal{R}_g$ to form
    fronts $\mathcal{F}_1, \mathcal{F}_2, \ldots$
    Compute crowding distance for each individual in
    each front
    Perform binary tournament selection on $\mathcal{R}_g$ using
    Pareto sorting to build mating pool
    Apply crossover & mutation $\rightarrow$ offspring $\mathcal{Q}_g$
    $\mathcal{R}_g \leftarrow \mathcal{R}_g \cup \mathcal{Q}_g$
    Perform non-dominated sorting and crowding
    distance on $\mathcal{R}_g$
    $\mathcal{P}_g \leftarrow$ select the candidate $\mathcal{R}_g$ by increasing front
    index, breaking ties by crowding distance
return $\mathcal{P}^*$ from $\mathcal{P}_g$
```

In this work, the fitness of each candidate path is evaluated based on two conflicting objectives: the average fluctuation of path length and the path stability. Unlike scalar-based ranking methods that sort solely by the fitness, NSGA-II adopts a Pareto-based selection strategy that better handles the trade-off. Each individual solution is evaluated in the objective space and assigned a *non-dominated rank* through Pareto sorting. Solutions in the first front are not dominated by any other solutions and represent the best current trade-offs. Subsequent fronts contain solutions dominated only by those in earlier fronts. This ranking ensures that selection prioritizes globally superior trade-offs rather than local optima in the weighted sum. Within each front, we compute the *crowding distance* of each solution to encourage diversity. This metric measures the density of solutions around a given point in the objective space, with larger values indicating more isolated and thus more diverse candidates. Selection is performed via tournament selection that considers both rank and crowding distance: individuals with lower non-dominated rank are always preferred, and ties are broken in favor of higher crowding distance. This rank-based approach replaces traditional fitness minimization and ensures both convergence and diversity.

As illustrated in Algorithm 1, the proposed algorithm starts by generating an initial population of P_{size} random paths connecting the source satellite s to the destination satellite d. In each generation, candidate paths are evaluated using both objective functions. Non-dominated sorting and crowding distance guide the selection process. Crossover and mutation operations generate offspring, and the population is updated by selecting the best individuals from the combined parent-offspring pool. This evolutionary process continues for G generations. Finally, the algorithm returns one or more optimal paths from the first front as the solution set $\mathcal{P}^*$, providing a robust trade-off between path stability and the variation in distance.

D. Complexity of Proposed Algorithm

The complexity of the proposed algorithm is primarily determined by the population size P_s, the number of generations G, and the average path length $L_{\mathcal{P}}$. In each generation, non-dominated sorting requires $O(P_s^2)$ time, while crowding distance calculation incurs $O(P_s \log P_s)$. Genetic operations, including tournament selection, crossover, and mutation, are applied to P_s individuals, with per-individual cost proportional to the path length, yielding a total of $O(P_s \cdot L_{\mathcal{P}})$. Hence, the overall time complexity across all generations is $O(G \cdot (P_s^2 + P_s \log P_s + PL_{\mathcal{P}}))$. Assuming that $L_{\mathcal{P}} = O(1)$, which holds in practice as the number of hops between LEO satellites remains bounded by geometric constraints and communication radius limitations, the dominant term is $O(G \cdot P_s^2)$. To maintain computational tractability for large-scale constellations, we set $P_s = 50$ and $G = 100$, resulting in 5,000 candidate path evaluations per run. Empirical results show that the GPU-accelerated implementation converges within 3–6 seconds for constellations of up to 1,500 satellites.

IV. Evaluation

This section first describes the simulation setup and then evaluates the performance of the STM mechanism and the optimal path selection policy.

A. Simulation Setup

We used two-line element (TLE) data for several open multiple satellite constellations[3] to provide accurate positional and orbital information. The TLE data were processed using the Systems Tool Kit (STK) for detailed link simulations. Simulations were conducted on a high performance workstation equipped with an NVIDIA GeForce RTX 3090 GPU and an Intel Xeon W-2245 CPU. We evaluated our approach on three representative LEO constellations: Globalstar, Iridium, and OneWeb. The simulation parameters and their values are summarized in Table IV.

B. Baselines

1) To evaluate the performance of STM mechanism for link prediction, we compared it with the following approaches:
   a) *Temporal Convolution Only (TCO):* The approach includes only the temporal convolution module to capture the dynamics of sequential data. This comparison removes the GAT module of STM mechanism, retaining solely the temporal convolution component.
   b) *Graph Convolution Network (GCN):* This variant replaces the GAT with a standard GCN for spatial modeling while retaining the temporal convolution module. This comparison evaluates the relative performance of GAT and standard GCN as spatial feature extraction methods.

TABLE IV
Parameter Settings

Key Parameter	**Value**
Number of Nodes	24-1584
Number of Edges	Nodes × 4
Number of Input Time Steps	3000
Number of Output Time Steps	300
Node Embedding Dimension	64
Edge Embedding Dimension	64
Temporal Convolution Output Channels	64
Spatial Convolution Output Channels	16
TimeBlock Kernel Size	3
Extra Projection Layer	Linear(12, 64)
Edge Projection Layer	Linear(256, 64)
Fully Connected Layer	Linear(3000 * 64, 300)

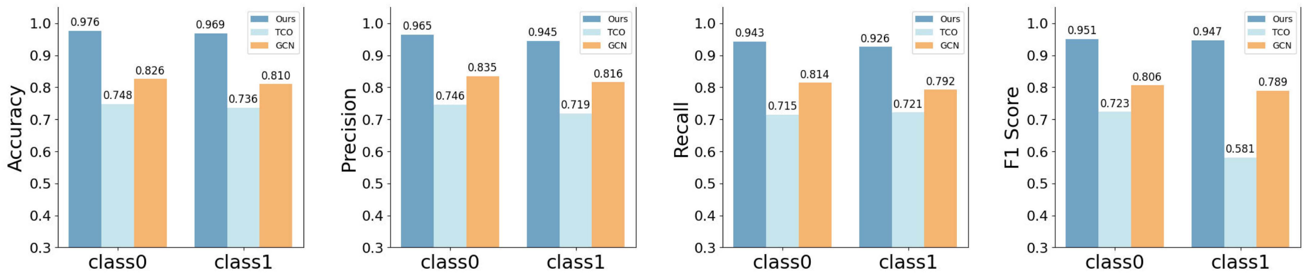
2) To evaluate the performance of the proposed optimal path selection algorithm, we compared it with the following approaches:
   a) *Static Dijkstra Path (SDP) [18]:* This algorithm computes the shortest path by traversing each topological graph. However, it does not adapt to dynamic network changes, which can lead to suboptimal paths in highly dynamic environments.
   b) *Dynamic Bellman-Ford Path (DBFP) [19]:* This algorithm accommodates time-varying network topologies by dynamically recalculating shortest paths as edge weights change. It incrementally updates path information as the network evolves. However, DBFP focuses exclusively on shortest path calculation and does not explicitly consider multi-objective trade-offs, such as the optimization of the distance jitter and the path stability.

C. Metrics

This subsection presents the metrics used to evaluate the performance in dynamic satellite networks, focusing on two main aspects: the link prediction accuracy of the proposed STM mechanism and the performance of the optimal path selection algorithm.

- *Link Prediction Accuracy:* Accuracy, precision, recall, and F1-Score are used to evaluate the performance of the link prediction model. Collectively, these metrics evaluate the performance to correctly predict the presence or absence of ISLs [20].
- *Path Selection Metrics:*
  – *Average Path Jitter (APJ):* This metric quantifies the fluctuation in latency along the path across different topologies. It evaluates path stability by calculating the standard deviation of the communication delay (in milliseconds). Lower jitter indicates more stable communication under dynamic conditions, demonstrating the path's ability to maintain consistent performance despite network changes.

[3] https://celestrak.org/NORAD/elements/



(a) Accuracy (b) Precision (c) Recall (d) F1 Score

Fig. 5. Link prediction for the Iridium constellation. Class 0 indicates predicted disconnected links, while Class 1 indicates predicted connected links. (a) Accuracy (b) Precision (c) Recall (d) F1 Score

TABLE V
Comparison of Baseline Methods for Different Satellite Constellations

Satellite Constellations	Methods	Average Propagation Delay (ms)	APJ	APA
Iridium	SDP	117.6	113.8	83.1
	DBFP	109.8	98.7	84.6
	Ours	**114.7**	**42.5**	**97.5**
Globalstar	SDP	123.9	143.6	85.3
	DBFP	121.3	129.3	87.4
	Ours	**124.6**	**57.8**	**96.3**
OneWeb	SDP	132.9	87.9	85.7
	DBFP	126.7	65.7	91.6
	Ours	**129.5**	**39.7**	**98.1**

– *Average Path Availability (APA):* This metric evaluates the reliability of the path connectivity by measuring how consistently the path remains operational during the observation period. It is calculated as the ratio of the time the path is available to the total observation time. Higher path availability indicates more reliable, continuous connectivity.

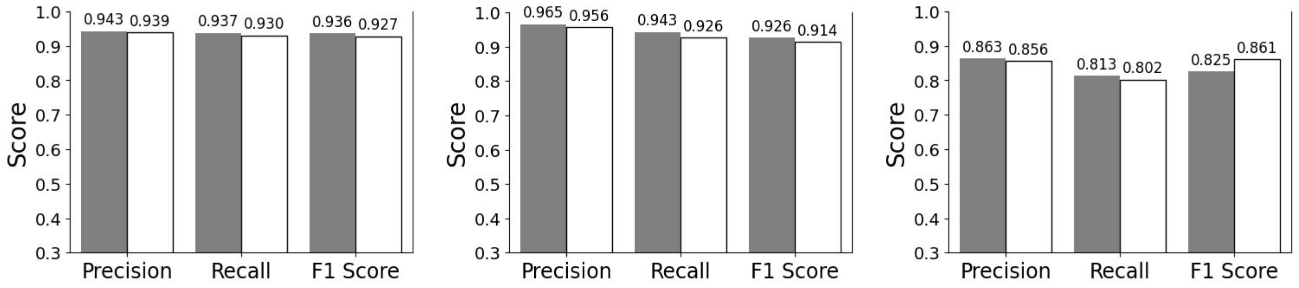
D. Evaluational Results

The simulation results demonstrate the advantages of the STM mechanism in predicting the connectivity of satellites and the proposed heuristic algorithm in optimizing KS paths in large-scale satellite constellations.

Our algorithm enhances path stability and reduces path length variation: The proposed heuristic solution significantly improves in performance, leading to more stable and efficient network performance. As illustrated in Table V, by effectively minimizing path length variation and demonstrating superior resilience to satellite constellations, the heuristic solution outperforms traditional algorithms such as DBFP. In this work, the algorithm employs a balanced weighting combination of $\lambda_1 = \lambda_2 = 0.5$ to optimize both path stability and latency. In the Iridium constellation, our method achieves an APJ of 42.5 ms and an APA of 97.5%. Compared to the SDP method, our approach reduces the APJ by 62.6% and increases the APA by 17.3%. Similarly, compared to the DBFP method, our method improves these two metrics by 56.9% and 15.2%, respectively. These results demonstrate that our algorithm significantly enhances both path stability and availability, making it highly effective for dynamic satellite networks. This enhanced stability ensures robust and reliable communication channels, even under significant changes in the satellite constellation.

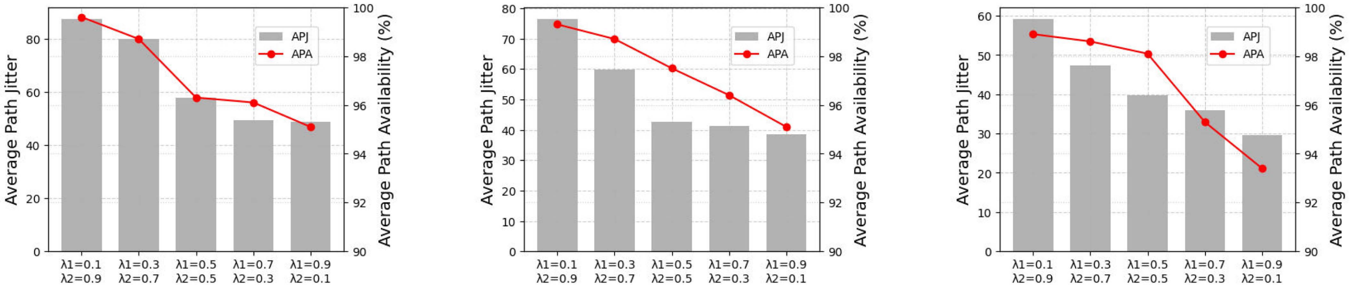
Our algorithm enhances link prediction accuracy: As illustrated in Fig. 5, our STM mechanism outperforms baseline methods by effectively capturing spatio-temporal correlations of satellite interlinks. While prediction accuracy slightly decreases with larger constellations due to increased complexity, it remains robust. The model excels at predicting disconnected links, often caused by predictable events like loss of line-of-sight, whereas predicting connected links is more challenging due to the dynamic nature. For example, in the Iridium constellation, the STM mechanism achieves an accuracy of 94.7%, with precision and recall values of 93.8% and 94.2%, respectively. Fig. 6 illustrates the prediction accuracy of different satellite constellations. The slight decrease in performance for OneWeb highlights the challenges posed by larger and more complex networks. However, the model's ability to maintain robust performance across diverse constellations underscores its effectiveness in different configuration.

Additionally, as illustrated in Fig. 7, we investigate the impact of different λ combinations on APJ and APA. For example,



(a) Globalstar (b) Iridium (c) OneWeb

Fig. 6. Verification results for different constellations. The indicators for connection and disconnection prediction for each constellation are shown. The left bars represent disconnection predictions, while the right bars represent connection predictions. (a) Globalstar (b) Iridium (c) OneWeb



(a) Globalstar (b) Iridium (c) OneWeb

Fig. 7. Impact of varying λ_1 and λ_2 on APJ and APA for the optimal path selection algorithm in various satellite constellations. (a) Globalstar (b) Iridium (c) OneWeb

TABLE VI
PERFORMANCE UNDER DIFFERENT SATELLITE CONSTELLATIONS

	Constellation configuration			Communication parameters			
	Total satellites	Altitude (km)	Inclination angle (°)	Average ISL propagation delay (ms)	Average coverage waiting time (min)	APJ	APA (%)
Tiansuan	24	480	97.4	122.7	65	51.73	95.4
Iridium	66	780	86.4	114.7	20.9	57.8	96.3
Globalstar	85	1400	52	124.6	20.9	42.5	97.5
Telesat	351	1015	98.98	104.4	4.44	37.16	99.2
OneWeb	581	1200	87	129.5	13.6	39.7	98.1
Kuiper	1156	630	51.9	129.8	1.24	26.14	98.7
Starlink	1584	550	53	132.2	0.9	32.16	99.3

higher λ_1 values (e.g., $\lambda_1 = 0.7$, $\lambda_2 = 0.3$) prioritize low jitter with slightly reduced availability, while higher λ_2 values (e.g., $\lambda_1 = 0.3$, $\lambda_2 = 0.7$) achieve higher availability with a slight increase in jitter. This demonstrates the algorithm's flexibility in adapting to diverse task requirements.

Our algorithm maintains superior performance across various constellation configurations: Our STM mechanism, combined with the path selection algorithm, effectively optimizes ISL paths across multiple constellations. As illustrated in Table VI, our approach improves critical metrics such as ISL propagation delay, coverage waiting time, path jitter and availability. For instance, the Starlink constellation, comprising 1,584 satellites, achieves a high path availability and a low path jitter. The STM mechanism significantly enhances communication performance, ensuring efficient and reliable inter-satellite connectivity across diverse satellite configurations.

V. Satellite-Ground Verification

To evaluate the feasibility of our KS framework, we implemented a fault diagnosis system on the BUPT-1 satellite, part of the TianSuan constellation.[4] The system utilizes a local cloud platform and utilizes X-band communication for satellite-ground connectivity. During the satellite-ground

[4] www.tiansuan.org.cn

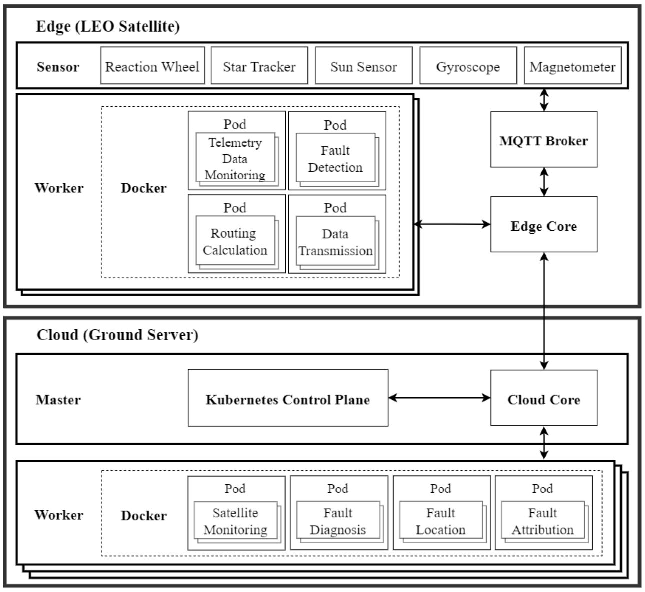
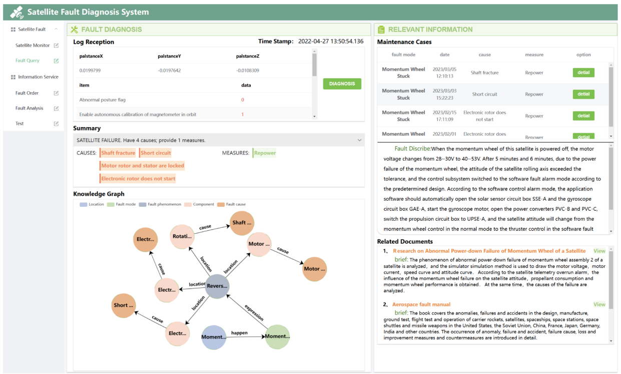


Fig. 8. System architecture based on KubeEdge for cloud–edge collaboration.



Log Reception Time Stamp: 2022-04-27 13:50:54.136

palstanceX	palstanceY	palstanceZ
0.0199799	-0.0197642	-0.0108309

item	data
Abnormal posture flag	0
Enable autonomous calibration of magnetometer in orbit	1

Summary

SATELLITE FAILURE. Have 4 causes; provide 1 measures.

CAUSES: Shaft fracture Short circuit Motor rotor and stator are locked Electronic rotor does not start MEASURES: Repower

Maintenance Cases

fault mode	date	cause	measure	option
Momentum Wheel Stuck	2023/03/05 12:10:13	Shaft fracture	Repower	detial
Momentum Wheel Stuck	2023/03/03 15:22:23	Short circuit	Repower	detial
Momentum Wheel Stuck	2023/02/15 17:11:09	Electronic rotor does not start	Repower	detial
Momentum Wheel	2023/02/01	Electronic rotor does	Repower	detial

Fault Discribe:When the momentum wheel of this satellite is powered off, the motor voltage changes from 28~30V to 40~53V. After 5 minutes and 6 minutes, due to the power failure of the momentum wheel, the attitude of the satellite rolling axis exceeded the tolerance, and the control subsystem switched to the software fault alarm mode according to the predetermined design. According to the software control alarm mode, the application software should automatically open the solar sensor circuit box SSE-A and the gyroscope circuit box GAE-A, start the gyroscope motor, open the power converters PVC-B and PVC-C, switch the propulsion circuit box to UPSE-A, and the satellite attitude will change from the momentum wheel control in the normal mode to the thruster control in the software fault

Related Documents

1、 Research on Abnormal Power-down Failure of Momentum Wheel of a Satellite View

brief: The phenomenon of abnormal power-down failure of momentum wheel assembly 2 of a satellite is analyzed, and the simulator simulation method is used to draw the motor voltage, motor current, speed curve and attitude curve. According to the satellite telemetry overrun alarm, the influence of the momentum wheel failure on the satellite attitude, propellant consumption and momentum wheel performance is obtained. At the same time, the causes of the failure are analyzed.

2、 Aerospace fault manual View

brief: The book covers the anomalies, failures and accidents in the design, manufacture, ground test, flight test and operation of carrier rockets, satellites, spaceships, space stations, space shuttles and missile weapons in the United States, the Soviet Union, China, France, Japan, Germany, India and other countries. The occurrence of anomaly, failure and accident, failure cause, loss and improvement measures and countermeasures are introduced in detail.

Fig. 9. Visual interface of fault diagnosis.

testing phase, we demonstrated the response speed and accuracy of the system in handling actual fault cases. In the inter-satellite simulation phase, we tested the interactions and link optimization capabilities of the system within a complex, dynamic inter-satellite network.

A. System Deployment and Architecture

We implemented a KubeEdge-based system [21] for cloud-edge collaboration. As shown in Fig. 8, the cloud side uses ground servers as the control plane, with the LEO satellite serving as an edge node, enabling collaborative reasoning on the cloud edge. The system adopts the B/S architecture to facilitate data access, storage, management, analysis, and visualization.

The satellite-end system is deployed on the dedicated space-grade computer of BUPT-1, providing edge services, such as telemetry monitoring, real-time fault detection, routing calculation, and data transmission. This system comprises sensors, an MQTT broker, an edge core, and docker components. The sensor monitors satellite status data and transmits them to the MQTT broker via a message queue. The edge core handles workload execution, device management, and message processing on the edge node. It receives and executes commands from the cloud core while sending the status and heartbeats of the satellite edge nodes. Docker uses pods as the smallest deployable and manageable computing units, encapsulating satellite services as a pod and deploying them on a worker node. To achieve stable and low-latency satellite-ground communication, we implemented the delivery of IP datagrams on the spatial link layer protocols of CCSDS. Satellite applications can access extensive resources on ground servers through a proxy server using the User Datagram Protocol (UDP).

The ground-end system is deployed on a central controller using a *Linux* server and a *Neo4j* database to manage the knowledge graph. The ground server continuously monitors the data packets forwarded by the proxy server in real-time and deploys multiple microservices to process various requests collaboratively. The cloud core within the master coordinates edge-side microservices, device management, message routing, and cloud-edge synchronization. The cloud core also interacts with the Kubernetes control plane for global coordination and device management. The Docker component on the cloud side deploys microservices in different pods, including satellite monitoring, fault diagnosis, location, and attribution. Upon receiving real-time requests from the satellite, the server initiates a query on the knowledge graph and then calls the UDP service to forward the results back to the satellite. The intermediate data for fault diagnosis are recorded in the relational database. To aid in decision-making, the system provides a user-friendly interactive visual interface, enabling users to intuitively understand the satellite operation status and view fault inference results and detailed traceability information.

B. Fault Diagnosis and Response

To verify the functionality of the proposed system, BUPT-1 satellite connected to the Tongchuan station on March 20, 2023. Control signaling was then delivered through the satellite-ground communication network to enable the call process between the satellite and the ground station. We selected momentum wheel fault diagnosis as a test case, as illustrated in Fig. 9. The primary goal of this test was to evaluate the real-time fault detection and response speed. Initially, the system performed real-time monitoring of payload telemetry. The onboard application automatically identified a momentum wheel fault and sent control commands to the edge network proxy to establish satellite-ground communication. Crucially, the onboard application transmitted the fault information, along with other operational statuses (e.g., control circuit, gyroscope, and related components), to the ground server. Simultaneously, the ground server monitored the downlink data and triggered a fault alarm to notify the technician. The fault diagnosis module then suggested potential fault causes, presenting clear explanations through a knowledge graph. Additionally, the visualized interface provided fault records, detailed reasoning processes, similar fault cases, and relevant documentation to support decision-making. This setup was designed to validate the real-time fault detection and its ability to quickly analyze faults.

TABLE VII
LATENCY FOR DIFFERENT PACKETS

Request id	Package (bytes)	Latency (s)			
		T_{dc} (s)	T_{kc} (s)	T_{ul} (s)	T_{total} (s)
1	58	0.82	0.09	0.89	1.80
2	79	1.03	0.11	0.91	2.05
3	91	0.94	0.11	0.87	1.92
4	140	0.91	0.13	0.97	2.01
5	169	0.83	0.12	0.89	1.84
6	79	1.14	0.09	1.01	2.24
7	91	0.69	0.10	0.80	1.59
8	192	0.79	0.13	0.81	1.73
9	153	0.85	0.11	0.84 s	1.80
10	143	0.83	0.12	0.84	1.79

We conducted ten tests targeting various faults. Table VII shows the performance of data transmission and inference delay. The packet size of the ten downlink requests ranged from approximately 50 to 200 bytes. The result validates that data transmission is controlled at a low level, saving the bandwidth of the satellite-ground communication link. The maximum delay for the downstream T_{dc} and upstream links T_{ul} was under 1.2 seconds. The inference delay on the ground server T_{kc} averaged 0.1 seconds. The total delay ranged from 1.59 seconds to 2.24 seconds, with an average delay of 1.88 seconds, confirming the capability of the framework for fast KS. Onboard experiments on BUPT-1 revealed that the total service time is significantly shorter than the 6-8 minute satellite-ground communication window, provided the satellite can access the ground station. However, due to Earth's rotation and high-speed movement of the satellite, BUPT-1 satellite revisits the same ground station every 26 hours. This can result in potential delays for service requests awaiting transmission of fault status through the satellite-ground link.

VI. RELATED WORK

A. Satellite Knowledge Service

Satellite KS aim to enhance system reliability and task performance through effective information management and intelligent inference. Recent advances in Big Data [22][23], artificial intelligence (AI) [24], [25], and cloud computing [26] have significantly driven progress in this field.

The effective integration and management of satellite data are essential for maximizing the utility of satellite systems, which generate large volumes of heterogeneous data [27], including remote sensing images [28], attitude data [29], and environmental information [30]. To address these challenges, Li et al. [31] proposed a cloud-based data management framework that enables unified access and efficient processing of satellite data. Building on this, Zhang et al. [32] demonstrated that data fusion techniques can improve the availability and accuracy of remote sensing data by combining information from multiple sources. Furthermore, Lin et al. [33] introduced a federated learning framework for LEO satellites, which effectively addresses resource heterogeneity, communication constraints, and model staleness to improve overall processing efficiency.

Intelligent decision support systems are crucial in satellite task planning [34][35]. For example, Zhang et al. [36] employed multi-agent reinforcement learning to improve the efficiency and success rate of multi-satellite missions. Li et al. [37] introduced an online micro-batch scheduling framework with a pre-trained reinforcement learning model to optimize the Earth observation satellite task scheduling. Wu et al. [38] proposed a dynamic task planning method for multi-source remote sensing satellites, optimizing target scheduling and resource use. Collectively, these methods improve satellite operational efficiency and task planning effectiveness. However, existing methods that rely on reinforcement learning and dynamic scheduling still face limitations regarding scalability and adaptability in highly dynamic environments. To address this gap, our approach combines lightweight onboard models with knowledge graphs on the ground, enabling scalable and adaptive task execution in dynamic satellite networks.

B. Dynamic Satellite Network

Research on dynamic satellite networks primarily focuses on optimizing performance [39], reliability [40], [41], and resource utilization [42], [43]. A significant body of work addresses network architecture [44] and performance evaluation within flexible environments, employing adaptive algorithms and reinforcement learning [45] to facilitate real-time topology. Lyu et al. [46] introduced a constrained multi-agent reinforcement learning algorithm for satellite routing that optimizes packet delay, energy efficiency, and packet loss constraints. Building on this foundation, Mao et al. [44] incorporated regional network partitioning and adaptive mechanisms to enhance routing efficiency and meet diverse quality of service requirements.

The increasing demand for high-speed, reliable satellite communication in remote regions, along with the growing volume of data generated by modern applications such as real-time imaging and global internet coverage, has made efficient satellite data transmission an urgent challenge [47]. Lin et al. [48] proposed a specific time-evolution diagram model with a continuous two-stage heuristic algorithm to optimize transmission scheduling, effectively addressing the scheduling challenges faced by LEO imaging satellites. Regarding resource management, Zhao et al. [49] presented a decentralized resource allocation strategy for multi-satellite networks, optimizing transmit power and beam patterns to minimize traffic mismatch.

In terms of security [50], Deng et al. [51] developed a blockchain-based Privacy Protection Protocol that leverages smart contracts to secure cross-platform information dissemination in LEO satellite networks. This protocol enhances query efficiency and privacy control and improves resistance against malicious attacks while reducing on-chain transaction delays. While existing studies have significantly advanced the performance and security of dynamic satellite networks, they still face challenges in handling complex topology changes and making real-time decisions. To address these issues, our model employs spatio-temporal modeling and optimization, enabling accurate real-time link prediction and path selection in dynamic environments.

VII. Discussion and Conclusion

To address the growing demand for efficient and stable knowledge services, we have developed a novel framework that improves satellite-ground collaboration. The proposed approach integrates advanced spatio-temporal modeling and optimization techniques to tackle challenges such as dynamic network topologies and limited communication resources. Experimental results demonstrate that the performance of our proposed framework outperforms baselines.

Our current implementation has two main limitations. First, it assumes reliable ground contact for knowledge inference, meaning that it would require enhancements such as local caching or on-board knowledge bases to maintain capabilities during prolonged disconnections. Second, while effective, our heuristic is not proven optimal; highly dynamic network conditions might benefit from more advanced algorithms or adaptive machine learning methods to improve stability and efficiency. Future work will address these issues by exploring collaborative knowledge sharing among satellites, including developing distributed knowledge graphs and enabling inter-satellite inference sharing. To scale the framework for large constellations, we will also investigate hierarchical or cluster-based approaches to maintain computational efficiency and robustness.

References

[1] A. C. Boley and M. Byers, “Satellite mega-constellations create risks in low earth orbit, the atmosphere and on earth,” *Sci. Rep.*, vol. 11, no. 10642, pp. 1–8, 2021.

[2] U. of Converned Scientists, ”*Uscs Satellite Database*,” 2024, Accessed: Nov. 13, 2024. [Online]. Available: https://www.ucsusa.org/resources/satellite-database

[3] C. Yang et al., “Towards efficient satellite computing through adaptive compression,” *IEEE Trans. Serv. Comput.*, vol. 17, no. 6, pp. 4411–4422, 2024.

[4] D. I. Elewaily, H. A. Ali, A. I. Saleh, and M. M. Abdelsalam, “Delay/disruption-tolerant networking-based the integrated deep-space relay network: State-of-the-art,” *Ad Hoc Netw.*, vol. 152, 2024, Art. no. 103307.

[5] Y. Chen et al., “Energy-aware satellite-ground co-inference via layer-wise processing schedule optimization,” in *Proc. 15th Asia-Pacific Symp. Internetware*, 2024, pp. 303–312.

[6] S. Wang, Q. Zhang, R. Xing, F. Qi, and M. Xu, “The first verification test of space-ground collaborative intelligence via cloud-native satellites,” *China Commun.*, vol. 21, no. 4, pp. 208–217, 2024.

[7] G. Zhang, Q. Yang, G. Li, J. Leng, and L. Wang, “A satellite incipient fault detection method based on local optimum projection vector and Kullback-Leibler divergence,” *Appl. Sci.*, vol. 11, no. 2, pp. 797–816, 2021.

[8] Q. Zhang et al., “Resource-efficient in-orbit detection of earth objects,” in *Proc. IEEE Conf. Comput. Commun.*, 2024, pp. 551–560.

[9] Y. Hao, C. Zhang, S. Chai, Z. Li, and X. Liu, “Satellite fault detection and diagnosis based on data compression and improved decision tree,” in *Proc. Chin. Automat. Congr.*, 2020, pp. 1686–1691.

[10] S. Bai, W. Wang, Z. Chen, and W. Yao, “Research on abnormal output current drop of solar array of a low earth orbit satellite,” *IEEE Aerosp. Electron. Syst. Mag.*, vol. 36, no. 5, pp. 48–58, 2021.

[11] C. Peng, F. Xia, M. Naseriparsa, and F. Osborne, “Knowledge graphs: Opportunities and challenges,” *Artif. Intell. Rev.*, vol. 56, no. 11, pp. 13071–13102, 2023.

[12] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, “Graph attention networks,” *stat*, vol. 1050, no. 20, pp. 10–48550, 2017.

[13] F. Teng, Y. Zhu, E. Zhang, X. Hu, Q. Sun, and L. Feng, “A knowledge service framework for fault diagnosis of low-earth orbit satellite constellation,” in *Proc. IEEE Int. Conf. Web Serv.*, 2023, pp. 669–676.

[14] Z. Lu, R. Zhi, and W. Ma, “Quick routing response to link failure in low-earth orbit satellite networks,” in *Proc. IEEE Int. Conf. Comput. Commun.*, 2022, pp. 690–695.

[15] S. Zhang, H. Tong, J. Xu, and R. Maciejewski, “Graph convolutional networks: A comprehensive review,” *Comput. Social Netw.*, vol. 6, no. 1, pp. 1–23, 2019.

[16] J. Schmidhuber and S. Hochreiter, “Long short-term memory,” *Neural Comput*, vol. 9, no. 8, pp. 1735–1780, 1997.

[17] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, “A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii,” in *Proc. 6th Int. Conf. Parallel Problem Solving from Nature PPSN VI*, 2000, pp. 849–858.

[18] B. Haeupler, R. Hladık, V. Rozhoň, R. E. Tarjan, and J. Tet˘k, “Universal optimality of dijkstra via beyond-worst-case heaps,” in *Proc. 65th Annu. Symp. Foundations Comput. Sci.*, 2024, pp. 2099–2130.

[19] O. K. Sulaiman, A. M. Siregar, K. Nasution, and T. Haramaini, “Bellman ford algorithm-in routing information protocol (rip),” in *Proc. J. Physics: Conf. Ser.*, vol. 1007, 2018, Art. no. 012009.

[20] R. Yacouby and D. Axman, “Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models,” in *Proc. 1st workshop Eval. comparison NLP Syst.*, 2020, pp. 79–91.

[21] Y. Xiong, Y. Sun, L. Xing, and Y. Huang, “Extend cloud to edge with kubeedge,” in *Proc. IEEE Symp. Edge Comput.*, 2018, pp. 373–377.

[22] D. B. Rawat, R. Doku, and M. Garuba, “Cybersecurity in Big Data era: From securing Big Data to data-driven security,” *IEEE Trans. Serv. Comput.*, vol. 14, no. 6, pp. 2055–2072, 2019.

[23] N. A. Ochuba, D. O. Olutimehin, O. G. Odunaiya, and O. T. Soyombo, “Reviewing the application of Big Data analytics in satellite network management to optimize performance and enhance reliability, with implications for future technology developments,” *Magna Scientia Adv. Res. Rev.*, vol. 10, no. 2, pp. 111–119, 2024.

[24] K. Thangavel et al., “Artificial intelligence for trusted autonomous satellite operations,” *Prog. Aerosp. Sci.*, vol. 144, 2024, Art. no. 100960.

[25] T. Meuser et al., “Revisiting edge ai: Opportunities and challenges,” *IEEE Internet Comput.*, vol. 28, no. 4, pp. 49–59, 2024.

[26] S. Wang and Q. Li, “Satellite computing: Vision and challenges,” *IEEE Internet Things J.*, vol. 10, no. 24, pp. 22514–22529, 2023.

[27] R. Kotawadekar, “Satellite data: Big data extraction and analysis,” in *Proc. Artif. Intell. Data Mining*, 2021, pp. 177–197.

[28] D. Li, M. Wang, F. Yang, and R. Dai, “Internet intelligent remote sensing scientific experimental satellite luojia3-01,” *Geo-Spatial Inf. Sci.*, vol. 26, no. 3, pp. 257–261, 2023.

[29] C. Yang, W. Lu, and Y. Xia, “Reliability-constrained optimal attitude-vibration control for rigid-flexible coupling satellite using interval dimension-wise analysis,” *Rel. Eng. System Saf.*, vol. 237, 2023, Art. no. 109382.

[30] J. Ren, J. Yang, F. Wu, W. Sun, X. Xiao, and J. C. Xia, “Regional thermal environment changes: Integration of satellite data and land use/land cover,” *Iscience*, vol. 26, no. 2, 2023, Art. no. 105820.

[31] C. Xu et al., “Cloud-based storage and computing for remote sensing Big Data: A technical review,” *Int. J. Digit. Earth*, vol. 15, no. 1, pp. 1417–1445, 2022.

[32] X. Zhu, F. Cai, J. Tian, and T. K.-A. Williams, “Spatiotemporal fusion of multisource remote sensing data: Literature survey, taxonomy, principles, applications, and future directions,” *Remote Sens.*, vol. 10, no. 4, 2018, Art. no. 527.

[33] Z. Lin, Z. Chen, Z. Fang, X. Chen, X. Wang, and Y. Gao, “Fedsn: A federated learning framework over heterogeneous leo satellite networks,” *IEEE Trans. Mobile Comput.*, vol. 24, no. 3, pp. 1293–1307, 2024.

[34] K. Thangavel et al., “Trusted autonomous operations of distributed satellite systems using optical sensors,” *Sensors*, vol. 23, no. 6, 2023, Art. no. 3344.

[35] J. Long, S. Wu, X. Han, Y. Wang, and L. Liu, “Autonomous task planning method for multi-satellite system based on a hybrid genetic algorithm,” *Aerospace*, vol. 10, no. 1, 2023, Art. no. 70.

[36] G. Zhang, X. Li, G. Hu, Y. Li, X. Wang, and Z. Zhang, “Marl-based multi-satellite intelligent task planning method,” *IEEE Access*, vol. 11, pp. 135517–135528, 2023.

[37] G. Li, X. Li, J. Li, J. Chen, and X. Shen, “Ptmb: An online satellite task scheduling framework based on pre-trained Markov decision process for multi-task scenario,” *Knowl.-Based Syst.*, vol. 284, 2024, Art. no. 111339.

[38] Q. Wu, J. Pan, and M. Wang, “Dynamic task planning method for multi-source remote sensing satellite cooperative observation in complex scenarios,” *Remote Sens.*, vol. 16, no. 4, 2024, Art. no. 657.

[39] Q. Zhang, X. Ruolin, L. Yuanzhe, Z. Ao, X. Mengwei, and W. Shangguang, "Research progress on intelligent technologies for satellite edge computing," *J. Softw.*, pp. 1–18, 2025, doi: 10.13328/j.cnki.jos.007410.

[40] Z. Zhu, R. Xing, Y. Zhang, C. Yu, A. Zhou, and S. Wang, "Poster: An end-to-end study on performance and reliability of LEO satellite-terrestrial data links," in *Proc. ACM SIGCOMM 2024 Conference: Posters Demos*, 2024, pp. 83–85.

[41] R. Wang, M. A. Kishk, and M.-S. Alouini, "Reliability analysis of multi-hop routing in multi-tier LEO satellite networks," *IEEE Trans. Wireless Commun.*, vol. 23, no. 3, pp. 1959–1973, 2023.

[42] A. Furutanpey, Q. Zhang, P. Raith, T. Pfandzelter, S. Wang, and S. Dustdar, "Fool: Addressing the downlink bottleneck in satellite computing with neural feature compression," *IEEE Trans. Mobile Comput.*, vol. 24, no. 8, pp. 6747–6764, 2025.

[43] Q. Chen, Z. Guo, W. Meng, S. Han, C. Li, and T. Q. Quek, "A survey on resource management in joint communication and computing-embedded sagin," *IEEE Commun. Surv. Tut.*, vol. 27, no. 3, pp. 1911–1954, 2024.

[44] B. Mao, X. Zhou, J. Liu, and N. Kato, "On an intelligent hierarchical routing strategy for ultra-dense free space optical low earth orbit satellite networks," *IEEE J. Sel. Areas Commun.*, vol. 42, no. 5, pp. 1219–1230, 2024.

[45] J. Ou et al., "Deep reinforcement learning method for satellite range scheduling problem," *Swarm Evol. Computation*, vol. 77, 2023, Art. no. 101233.

[46] Y. Lyu, H. Hu, R. Fan, Z. Liu, J. An, and S. Mao, "Dynamic routing for integrated satellite-terrestrial networks: A constrained multi-agent reinforcement learning approach," *IEEE J. Sel. Areas Commun.*, vol. 42, no. 5, pp. 1204–1218, 2024.

[47] J. Chen et al., "A remote sensing data transmission strategy based on the combination of satellite-ground link and geo relay under dynamic topology," *Future Gener. Comput. Syst.*, vol. 145, pp. 337–353, 2023.

[48] X. Lin, Y. Chen, J. Xue, B. Zhang, L. He, and Y. Chen, "Large-volume leo satellite imaging data networked transmission scheduling problem: Model and algorithm," *Expert Syst. with Appl.*, vol. 249, 2024, Art. no. 123649.

[49] M. Zhao, H. Yu, J. Pan, Y. Jin, G. Lv, and Y. Jin, "Dynamic resource allocation for multi-satellite cooperation networks: A decentralized scheme under statistical CSI," *IEEE Access*, vol. 12, pp. 15419–15437, 2024.

[50] Y. Zhang, S. Zhao, J. He, Y. Zhang, Y. Shen, and X. Jiang, "A survey of secure communications for satellite internet based on cryptography and physical layer security," *IET Inf. Secur.*, vol. 2023, no. 1, 2023, Art. no. 5604802.

[51] X. Deng, J. Shao, L. Chang, and J. Liang, "A blockchain-based privacy protection protocol using smart contracts in LEO satellite networks," *Peer-to-Peer Netw. Appl.*, vol. 17, no. 2, pp. 800–818, 2024.

Fei Teng is a Professor and vice Dean at Southwest Jiaotong University, specializes in Big Data intelligence, industrial Big Data, and service computing. She has published over 80 papers in top journals like TC, TPDS, and IJCAI, and holds 20+ national patents. Leading projects such as the Sichuan Outstanding Youth Fund and the National Natural Science Foundation of China, she has received awards including the Sichuan Science and Technology Progress Award. Her research focuses on scalable and efficient solutions for large-scale data and dynamic systems.

Hao Zhao (Member, IEEE) received his B.S degree in the Faculty of Information Technology from China University of Mining and Technology in 2023. He is currently pursuing a master degree in School of Computer and Artificial Intelligence, University of Southwest Jiaotong. His main research interests include Deep Learning and Satellite Computing.

Qiyang Zhang (Member, IEEE) received the PhD degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 2024. He is currently a research assistant with the School of Computer Science, Peking University, Beijing, China. He is also a visiting student in Distributed Systems Group at TU Wien from 2022 to 2023. He has published papers in WWW, INFOCOM, TMC, TSC, IEEE NETWORK, etc. His research interests include Satellite Computing, Edge Intelligence.

Ranga Rao Venkatesha Prasad (Senior Member, IEEE) is an Associate Professor at ENSys group of TU Delft. His research interest is in the areas of Space-IoT, Tactile Internet, and 60 GHz mmWave networks. He has supervised 20 PhD students and more than 65 MSc students. He has close to 300 publications in peer-reviewed international journals and conferences and standards, and book chapters. He has served on the editorial board of IEEE JSAC, IEEE TMC, IEEE COMST, IEEE TGCN and many other IEEE Transactions. He was the vice-chair of IEEE Tactile Internet standardization workgroup and is now a mentor.

Schahram Dustdar (Fellow, IEEE) is Full Professor of Computer science heading the Research Division of Distributed Systems at the TU Wien, Austria. He is also part-time ICREA Research Professor at UPF Barcelona, Spain. He is Editor-in-Chief of Computing (Springer). He is an Associate Editor of IEEE Transactions on Services Computing, IEEE Transactions on Cloud Computing, ACM Transactions on the Web, and ACM Transactions on Internet Technology, as well as on the editorial board of IEEE Internet Computing and IEEE Computer. Dustdar is elected member of the Academia Europaea - The European Academy of Science.