INL/JOU-25-82885-Revision-0

Link Scheduling in Satellite Networks via Machine Learning over Riemannian Manifolds

January 2025

Changing the World's Energy Future

Joarder Jafor Sadique, Ahmed S. Ibrahim, Imtiaz Nasim

Idaho National Laboratory

INL/JOU-25-82885-Revision-0

Link Scheduling in Satellite Networks via Machine Learning over Riemannian Manifolds

Joarder Jafor Sadique, Ahmed S. Ibrahim, Imtiaz Nasim

January 2025

Idaho National Laboratory
Idaho Falls, Idaho 83415

http://www.inl.gov

Prepared for the
U.S. Department of Energy
Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517

IEEE Open Journal of the Communications Society

Received 29 December 2024; accepted 20 January 2025. Date of publication 23 January 2025; date of current version 7 February 2025.
Digital Object Identifier 10.1109/OJCOMS.2025.3533296

Link Scheduling in Satellite Networks via Machine Learning Over Riemannian Manifolds

JOARDER JAFOR SADIQUE [1] (Graduate Student Member, IEEE), IMTIAZ NASIM [2] (Member, IEEE), AND AHMED S. IBRAHIM [1] (Senior Member, IEEE)

[1] Department of Electrical and Computer Engineering, Florida International University, Miami, FL 33199, USA
[2] Data Science and Applied Statistics, Idaho National Laboratory, Idaho Falls, ID 83415, USA

CORRESPONDING AUTHOR: J. J. SADIQUE (e-mail: jsadi004@fiu.edu)

This work was supported in part by the National Science Foundation under Award CNS-2144297.

ABSTRACT Low Earth Orbit (LEO) satellites play a crucial role in enhancing global connectivity, serving a complementary solution to existing terrestrial systems. In wireless networks, scheduling is a vital process that allocates time-frequency resources to users for interference management. However, LEO satellite networks face significant challenges in scheduling their links towards ground users due to the satellites' mobility and overlapping coverage. This paper addresses the dynamic link scheduling problem in LEO satellite networks by considering spatio-temporal correlations introduced by the satellites' movements. The first step in the proposed solution involves modeling the network over Riemannian manifolds, thanks to their representation as symmetric positive definite matrices. We introduce two machine learning (ML)-based link scheduling techniques that model the dynamic evolution of satellite positions and link conditions over time and space. To accurately predict satellite link states, we present a recurrent neural network (RNN) over Riemannian manifolds, which captures spatio-temporal characteristics over time. Furthermore, we introduce a separate model, the convolutional neural network (CNN) over Riemannian manifolds, which captures geometric relationships between satellites and users by extracting spatial features from the network topology across all links. Simulation results demonstrate that both RNN and CNN over Riemannian manifolds deliver comparable performance to the fractional programming-based link scheduling (FPLinQ) benchmark. Remarkably, unlike other ML-based models that require extensive training data, both models only need 30 training samples to achieve over 99% of the sum rate while maintaining similar computational complexity relative to the benchmark.

INDEX TERMS Convolutional neural network, LEO satellite, link scheduling, recurrent neural network, Riemannian geometry, symmetric positive definite matrices, spatio-temporal correlation.

I. INTRODUCTION

THE EMERGENCE of fifth-generation (5G) and beyond 5G (B5G) networks promises enhanced capacity, yet achieving universal connectivity remains a formidable challenge that B5G networks may only partially achieve [1]. The majority of the global population still lacks adequate Internet access due to the high cost and complex deployment issues of mobile base stations. As a complementary approach to providing seamless connectivity for a large number of users, low Earth orbit (LEO) satellite networks have demonstrated significant potential that may not be attainable by solely terrestrial networks [2]. LEO satellites operate at lower altitudes (between 500 to 2000 kilometers) compared to geostationary Earth orbit (GEO) and medium Earth orbit (MEO) satellites. This close proximity makes LEO satellite communication more effective in terms of latency, power consumption, and deployment expenses, thus facilitating the development of future wireless networks for achieving global coverage [3]. Inspired by the potential for widespread global connectivity, companies like SpaceX and OneWeb are spearheading initiatives to develop dynamic network architectures consisting of thousands of LEO satellites in orbit [4].

A dense deployment of LEO satellites may lead to increased levels of interference, while their continuous motion relative to a ground-base observer induces spatio-temporal correlation.

In regions where satellite coverage overlaps, user equipment (UE) might encounter considerable interference, resulting in major deterioration in performance [4], [5]. Additionally, the spatio-temporal correlation arises from the temporal evolution of their movement over time. Together, the interference and spatio-temporal correlation pose a challenge to link scheduling in LEO satellite communication towards ground UE, and this is the focus of this paper.

Link scheduling strategies are crucial for reducing excessive interference and involve carefully selecting and activating a subset of UE links. Also, link scheduling has significant real-world applications, particularly in satellite-based Internet of Things (IoT) systems and their integration with 5G networks. An efficient link scheduling strategy supports the seamless operation of IoT devices in remote and underserved areas, where terrestrial infrastructure is often limited or unavailable. Over the last few years, several satellite operators and IoT companies have recognized the potential for innovation and business opportunities in enabling direct IoT-to-satellite communication [6]. Such advancements can enhance critical sectors like environmental monitoring, smart agriculture, and disaster management, enabling robust connectivity and data exchange in challenging environments [7]. However, while LEO satellites provide global coverage, IoT devices with poor channel conditions face difficulties when directly offloading tasks to these satellites, leading to network congestion and high energy consumption [8]. Link scheduling models efficiently address these challenges by leveraging interference patterns and resource allocation strategies, ensuring reliable and energy-efficient communication. Moreover, the integration of satellite networks with 5G systems supports low-latency, high-capacity communication for real-time applications such as telemedicine, autonomous vehicles, and emergency response, thereby emphasizing the transformative potential of this area of research [9].

Link scheduling problem aiming to maximize the sum rate over all links is formulated as a non-convex combinatorial optimization problem [10]. Traditional link scheduling methods have mainly relied on techniques such as sequential link selection algorithms [11], iterative fractional programming algorithms [12], and distance-based link scheduling strategies [13]. Alternatively, deep learning (DL) approaches, such as deep reinforcement learning (DRL) [14] and convolutional neural networks (CNN) [15], have been applied to achieve sum rates comparable to optimization-based methods. Such DL models require large volumes of training data to train their models (i.e., as in [16], [17]). For example, a deep neural network (DNN) approach for link scheduling was proposed in device-to-device (D2D) network [18], requiring as many as $10,000$ training samples.

A. MOTIVATIONS AND CONTRIBUTIONS

Motivated by the significance of satellite link scheduling in real-world application and its associated challenges, we follow a different approach by reformulating the dynamic satellite networks over Riemannian manifolds. Previously, Riemannian geometry has been applied in communication systems to tackle various challenges, including the design of beamforming codebooks [19], the deployment of relays [20], link scheduling [10], [21], [22] and power allocation in random device-to-device (D2D) wireless networks [23], and covariance shaping in [24]. In this work, the entire topology of links is represented as a point over Riemannian manifolds. Specifically, the network structure around each UE link is represented as a graph, including any interfering links. This is then modeled as a symmetric positive definite (SPD) matrix, which can be represented as a single point over Riemannian manifolds (i.e., curved surfaces). The sequence of spatio-temporal correlated network topologies can be modeled as a series of points over Riemannian manifolds. Efficiently abstracting such local interference networks using single points over non-Euclidean surfaces (i.e., Riemannian manifolds) makes it possible to use simple machine learning (ML) models for extracting the spatio-temporal correlation and hence requiring fewer training samples, compared to existing DNN models over Euclidean (i.e., flat) surfaces.

In this paper, we propose two alternative ML-based approaches. Each is suited to different aspects of the link scheduling problem in LEO satellite networks. First, given the spatio-temporal nature of satellite movement and time-dependent network dynamics, we employ a recurrent neural network (RNN) over Riemannian manifolds to capture the evolution of temporal patterns in each satellite-to-user wireless link across time slots. The Riemannian metric, which is Stein metric in this case, is used as a similarity measure to accurately model time series forecasting. Time series forecasting also involves using the statistical recurrent unit (SRU) that analyzes the spatio-temporal correlations between satellite links at previous time slots to predict future link scheduling decisions. In [25], SRU was proposed as a tool for time series forecasting in Euclidean space. The SRU effectively captures long-term dependencies in time series by maintaining moving averages of the data, commonly referred to as summary statistics.

Second, using an alternative approach to account for the spatial variability of the overall network topology across time slots, we introduce a convolutional neural network (CNN) over Riemannian manifolds. This model utilizes the changing geographic positions of satellites and users, detecting spatial patterns. In summary and by leveraging the RNN to capture how each pair evolves over time and the CNN to capture how the overall network topology changes across time slots, we propose two different approaches to model the spatio-temporal correlation inherent in the dynamic nature of satellite networks. Our simulation outcomes are assessed through a comparison with the fractional programming technique, FPLinQ [12], designed to produce high-quality local optimum solutions.

The key contributions of this work are threefold, which are:

- Efficient modeling of link scheduling in LEO satellite networks over Riemannian manifolds, enabling ML models to capture the spatio-temporal correlations.
- By capturing the evolution of each satellite-UE pair over time, RNN-based approach achieves over 99% of sum rate with similar computational complexity while predicting link scheduling decisions up to 7 future time slots, using only 30 training samples of wireless network layouts.
- By capturing the evolution of each network topology over time, CNN-based method also maintains a comparable computational complexity and sum rate performance with an equal number of training samples of 30 wireless network layouts to predict 7 time slots ahead.

B. RELATED WORKS

In recent years, ML-based strategies have been increasingly adopted to address various scheduling challenges in non-terrestrial networks, particularly in satellite networks. These approaches aim to enhance network performance, ensure seamless connectivity, and optimize resource utilization. For instance, [26] provides an overview of AI/ML-driven techniques for addressing problems related to ISTNs. In [27], the authors focused on optimizing the space-HAPS-ground network to solve user scheduling problems under specific connectivity and power constraints, employing an ensemble deep neural network (EDNN) model for network optimization. Similarly, [28] proposed a deep reinforcement learning-based construction model to handle scheduling issues in agile earth observation satellites (AEOS) for large-scale satellite management. Furthermore, [29] applied a Q-learning approach to optimize task sequencing in satellite management systems. The optimization process primarily relied on a reinforcement learning-based memetic algorithm (RL-MA) to achieve energy-efficient satellite range scheduling. These works highlight the diverse applications of AI/ML in addressing complex scheduling problems in dynamic and non-terrestrial networks.

While link scheduling in satellite networks is a key component of integrated satellite-terrestrial network (ISTN) to ensure seamless global coverage, several works have focused on other aspects of ISTN, such as dynamic spectrum sharing and spectrum sensing for achieving the same objective. Authors in [30] formulated a dynamic spectrum-sharing strategy utilizing the same spectrum resources in satellite and terrestrial terminals. In [31], improvement in spectrum efficiency was proposed by utilizing non-orthogonal multiple access and cognitive radio techniques. Given the extensive coverage provided by satellite sub-networks, various satellite-driven spectrum sensing schemes have been introduced, playing a vital role as a prerequisite for dynamic spectrum sharing [32], [33].

Satellite link scheduling has not received as much attention as other scheduling strategies [34] such as inter-satellite link (ISL) scheduling [35], [36], [37], satellite range scheduling [38], [39], and satellite imaging scheduling [40], [41]. Some research has explored the integration of routing and link scheduling techniques [36], [42] to boost the throughput. Subsequently, a dynamic approach to scheduling satellite topology allowed for swift network reconstruction and helped reduce the decline in routing performance [42]. In [43], authors investigated strategies for link allocation and power distribution to address network cost optimization issues.

In addition to LEO-based networks, MEO and GEO architectures have also been explored for link scheduling and resource management. MEO satellites, due to their moderate altitude, offer a trade-off between coverage area and latency, making them suitable for applications like navigation and certain communication systems [44]. Traditional non-AI-based schemes in GEO architectures, such as heuristic algorithms and greedy scheduling techniques, have been widely used for satellite imaging scheduling and range scheduling tasks [40], [45]. For instance, classical time-division multiple access (TDMA) and frequency-division multiple access (FDMA) methods have been implemented in GEO satellites for resource allocation, though these approaches often suffer from inefficiencies in dynamic environments [46]. Additionally, the use of linear programming and graph-based algorithms has been prominent in conventional satellite scheduling but lacks the adaptability required for highly dynamic LEO networks.

Several works in the literature utilize spatio-temporal correlations for scheduling. To address the issue related to thermal-aware scheduling in high-performance computing, spatio-temporal correlation of the temperature evolution over time was considered in [47]. Authors in [48] introduced scheduling problem for remote estimation in a wireless sensor networks considering the spatio-temporal dependency of the broadcasting observations. By retaining moving averages of statistics, long-term dependencies in time series data can be captured [25], and it has been successfully utilized in a range of time series prediction tasks in the past [49], [50].

Outline: The remainder of the paper is organized as follows. Section II provides an introduction to the preliminary concepts of Riemannian manifolds and outlines major parameters for LEO satellite networks. Section III describes the system model. Section IV focuses on the formulation of the problem. Section V explores machine learning models for link-state prediction, presenting both RNN over Riemannian manifolds and CNN over Riemannian manifolds. Section VI showcases the performance of the proposed link scheduling techniques and offers an analysis of the computational complexity of these methods. Finally, Section VII concludes with a key summary.

II. SYSTEM MODEL

Figure 1 depicts the major parameters of the satellite communication network. In this scenario, the LEO satellite

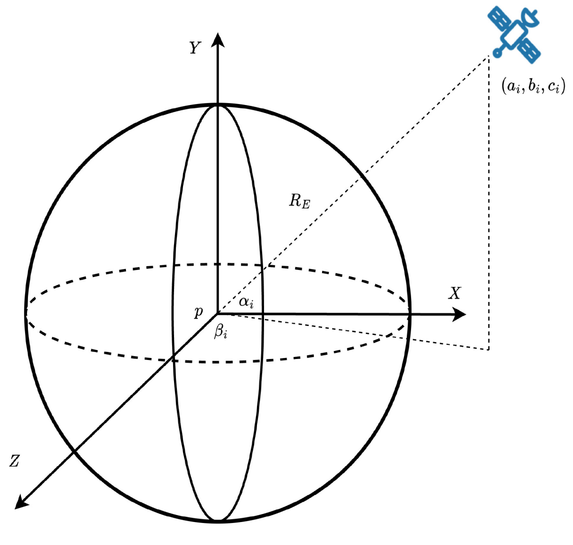

FIGURE 1. Major parameters of LEO satellite communication.

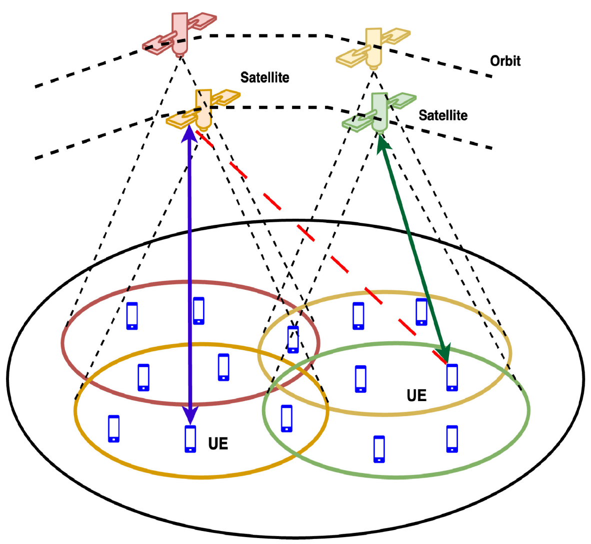

FIGURE 2. LEO satellite communication network layout.

communication system is represented in a Cartesian coordinate system, where a single LEO satellite travels at a certain velocity and operates from γ_i altitude. The origin of the coordinates is denoted by p, and the line corresponding to the longitude 0 is placed in the XpY plane. For a satellite $s_i \in S$, latitude, longitude, and altitude are denoted as α_i, β_i, and γ_i, respectively. The values of α ranges from $-\pi/2$ to $\pi/2$ and β ranges from $-\pi$ to π. Then the coordinates of s_i at time slot t can be computed as

$$\left.\begin{aligned} a_i &= (\gamma_i + R_E)\cos\alpha_i \sin\beta_i \\ b_i &= (\gamma_i + R_E)\cos\alpha_i \cos\beta_i \\ c_i &= (\gamma_i + R_E)\sin\alpha_i \end{aligned}\right\} \quad (1)$$

where R_E denotes the radius of the earth.

As shown in Figure 2, every satellite in $S = \{s_1, s_2, \ldots, s_S\}$ has a footprint, and a number of UEs are connected to each satellite. Using eq. (1), the position of satellites for each time slot is computed. Basically, K satellite-UE pairs are formed with S satellites. As a transmitter, each satellite makes pairs with all UEs within the area covered by their footprints from a certain altitude. We focus on the downlink scenario, in which each satellite applies superposition coding to send information to the active UEs that are associated with it. It is assumed that each UE decodes its information, considering signals for other UEs as interference. Interference terms include signals coming from its own satellite towards other active UEs within its vicinity.

We assume a single-antenna for each UE. Using an information-theoretic approach, the capacity of the link towards q-th UE at a time slot t from its satellite is given by

$$R_q^t(\mathbf{c^t}) = B\log_2\left(1 + \frac{P_\tau c_q^t |h_{qs}^t|^2 g_{qs}^t}{\sum_{s'\in\mathcal{S}, i\neq q} P_\tau c_i^t |h_{is'}^t|^2 g_{is'}^t + \sigma_n^2}\right), \quad (2)$$

where B is the bandwidth, P_τ denotes the fixed transmit power level of the satellites, and σ_n^2 is the noise variance. h_{qs} represents instantaneous channel gain due to small-scale fading effects. Here, $\mathbf{c^t} = [c_1^t, \ldots, c_K^t]^T$ acts as the indicator vector for link states, where $c_q^t = 1$ indicates that q-th satellite link is activated and $c_q^t = 0$ otherwise. In each scheduling slot, a subset of links are activated to transmit at the same time.

The channel between LEO satellite and UE is denoted as g_{qs}. In densely deployed LEO satellite networks with high mobility, obtaining instantaneous channel state information (CSI) is challenging due to the rapid changes in the environment [51]. Therefore, we employ statistical CSI based on distance metrics to estimate the channel conditions without relying on instantaneous CSI. Using statistical CSI is supported by [52] and [53].

The value of the channel gain can be obtained from the following link budget equation [32]

$$g_{qs}^t = G_{r,max} G_t \left(\frac{c}{4\pi f_c d_{q,s}^t}\right)^2 10^{A_g/10} 10^{A_c/10}, \quad (3)$$

where $G_{r,max}$ represents the maximum gain of the receiving antennas of the UE and G_t is the gain of the LEO satellite. $c = 3 \times 10^8\ ms^{-1}$ represents the speed of light, f_c is the operating frequency of satellite in Hertz. A_g is the gaseous absorption factor and A_c represents cloud or fog absorption factor in dB. $d_{q,s}^t$ denotes the distance between the s-th satellite and the q-th UE at time slot t.

The model shown in Figure 2 can be represented as a weighted and directed finite graph $\mathcal{G}^t(V, E)$, where t is the time slot ($t = 1, 2, \ldots, T$), V denotes the set of $n = 2K$ nodes, and E represents the set of m edges that includes the communication links and the interfering links to all neighbors. The edges comprise the communication links between the satellite-UE pairs and the interfering links to all neighbors at any time slot t. The incidence matrix $\mathbf{D^t} \in \mathrm{R}^{n\times m}$ of graph $\mathcal{G}^t$ at time slot t is the matrix with l-th column given by edge vector $\mathbf{e}_l^t$. The edge vector $\mathbf{e}_l^t \in \mathrm{R}^n$ is specified as having $e_{l_q}^t = 1$, $e_{l_s}^t = -1$, and all other entries set to zero, for edges linking nodes q and s, where $q, s \in K$ and $q \neq s$. The weight matrix $\mathbf{W^t} \in \mathrm{R}^{m\times m}$ is specified as a diagonal matrix in which each diagonal entry denotes the weight of the l-th edge. The edge weight is determined by the Euclidean distance between its two nodes, and this distance is represented with a finite precision of r bits via uniform quantization [54], reducing the range from infinite to 2^r possible values.

At time slot t, the Laplacian matrix $\mathfrak{L}_q^t \in \mathrm{R}^{m\times m}$ is given by

$$\mathfrak{L}_q^t = \mathbf{D^t}\mathbf{W^t}\left(\mathbf{D^t}\right)^{\mathbf{T}}, \tag{4}$$

where $\mathbf{T}$ representing matrix transposition. Since Laplacian matrices are positive semi-definite, a regularization step by adding a scaled identity matrix yields a regularized SPD Laplacian matrix [55] at time slot t, expressed as

$$\mathbf{S}_q^t = \mathbf{D^t}\mathbf{W^t}\left(\mathbf{D^t}\right)^{\mathbf{T}} + \beta\mathbf{I}, \tag{5}$$

where $\beta > 0$ is a regularization parameter and $\mathbf{I}$ is the $n \times n$ identity matrix.

III. PROBLEM FORMULATION

The Riemannian manifold $(\mathcal{M}, \mathcal{L})$ is a real differentiable manifold $\mathcal{M}$, where each tangent space is equipped with an inner product $\mathcal{L}$, known as the Riemannian metric. Additionally, the set of $n \times n$ SPD matrices, denoted as Sym_n^{++}, resides within the interior of convex cones, forming a specific class of Riemannian manifolds [56].

For satellite link scheduling, our goal is to determine the optimal combinations of the indicator vector $\mathbf{c^t}$ that maximize the total of instantaneous information-theoretic rates over the T time frame, as given by

$$\max_{\mathbf{c^t}} \frac{1}{T}\sum_{t=1}^{T}\sum_{q=1}^{K} R_q^t\left(\mathbf{c^t}\right) \quad s.t.\ \mathbf{c^t} \in \{0, 1\}^K. \tag{6}$$

Addressing the link scheduling problem in any wireless network demands proper interference modeling. Hence, we model the wireless network graph $\mathcal{G}_q^t$ at time slot t for link scheduling as shown in Figure 3. We consider three strategic ways of modeling this wireless network of satellite-UE pairs with three Laplacian matrices.

For the first matrix, graph $\mathcal{G}_q^t$, referred to as $\mathcal{G}_{\mathcal{D}_q}^t(V, E_{\mathcal{D}_q})$, corresponds only to the direct links between the intended satellite-UE pair at time slot t as depicted in Figure 3(a). This also provides information about the signal-to-noise ratio for a certain satellite-UE pair. The next two methods model the interference. Figure 3(b) depicts the formation of second matrix using the graph $\mathcal{G}_q^t$, where the intended pair experiences interference from the nearest scheduled pair at time slot t and referred to as $\mathcal{G}_{\mathcal{N}_q}^t(V, E_{\mathcal{N}_q})$. Lastly, in the third matrix, as depicted in Figure 3(c), we model the graph $\mathcal{G}_q^t$ considering the impact of interference to the nearest pairs when the intended pair is scheduled at time slot t and represented as $\mathcal{G}_{\mathcal{P}_q}^t(V, E_{\mathcal{P}_q})$. Here, $E_{\mathcal{P}_q}$ corresponds to the interference links from the scheduled link to the nearest pairs, along with all direct links of the nearby pairs.

Algorithm 1: Sequential SPD Points Modeling Over Manifold

Input: $\mathfrak{L}_{\mathcal{D}_\mathbf{q}}^\mathbf{t}$, $\mathfrak{L}_{\mathcal{N}_\mathbf{q}}^\mathbf{t}$, and $\mathfrak{L}_{\mathcal{P}_\mathbf{q}}^\mathbf{t}$, $\forall q \in K$, $t \in T$
Initialization: $\mathbf{S_q^t} = 0$, $\forall q \in K$, $t \in T$
for $t = 1$ **to** T **do**
  for $q = 1$ **to** K **do**
    `S1`: Calculate $\mathbf{S}_{\mathcal{D}_\mathbf{q}}^\mathbf{t}$, $\mathbf{S}_{\mathcal{N}_\mathbf{q}}^\mathbf{t}$, and $\mathbf{S}_{\mathcal{P}_\mathbf{q}}^\mathbf{t}$ from eq. (5).
    `S2`: Add SPD matrices of each time slot t and compute $\mathbf{S_q^t}$.
  end
end
return $\{\mathbf{S_q^t}\}_{t=1}^T$, $\forall q = \{1, \ldots, K\}$

Basically, all these three Laplacian matrices, $\mathfrak{L}_{\mathcal{D}_\mathbf{q}}^\mathbf{t}$, $\mathfrak{L}_{\mathcal{N}_\mathbf{q}}^\mathbf{t}$, and $\mathfrak{L}_{\mathcal{P}_\mathbf{q}}^\mathbf{t}$ at time slot t are positive semi-definite in nature. By following the regularization step addressed in eq. (5), three SPD matrices, $\mathbf{S}_{\mathcal{D}_\mathbf{q}}^\mathbf{t}$, $\mathbf{S}_{\mathcal{N}_\mathbf{q}}^\mathbf{t}$, and $\mathbf{S}_{\mathcal{P}_\mathbf{q}}^\mathbf{t}$ are formulated. Since adding SPD matrices produces another SPD matrix [57], we combine these three matrices by adding them to have complete interference information of the intended satellite-UE pair at a certain time instant t. Algorithm 1 describes the steps of formulating SPD matrices from semi-definite Laplacian matrices at time slot t.

IV. MACHINE LEARNING FOR LINK-STATE PREDICTION

This section explains details about modeling the link-state prediction problem via ML over Riemannian manifolds considering spatio-temporal correlation in LEO satellite network. At first, we introduce local graph modeling on RNN over Riemannian manifolds, where the time-correlated movement of each link is captured to predict the link-state across different time slots. Then, we address the same problem on CNN over Riemannian manifolds where the neural network captures the evolution of whole network topology over time utilizing geographical locations of interfering and interfered nodes.

A. LINK-STATE PREDICTION WITH RNN OVER RIEMANNIAN MANIFOLDS

In the link scheduling problem model, after capturing the network layout for each time slot t, the necessary information for devising effective scheduling is retained as a sequence of SPD points on the manifold. Handling sequential data of this nature is achieved through the use of RNN architectures, which are commonly applied in tasks like machine translation [58]. The fundamental idea of RNN is to estimate the conditional probability distribution of the output sequence given the input sequence [59].

The correlated temporal movements of satellites can be expressed as a sequence of SPD points, $\mathbf{S}_q^1, \ldots, \mathbf{S}_q^T \in$

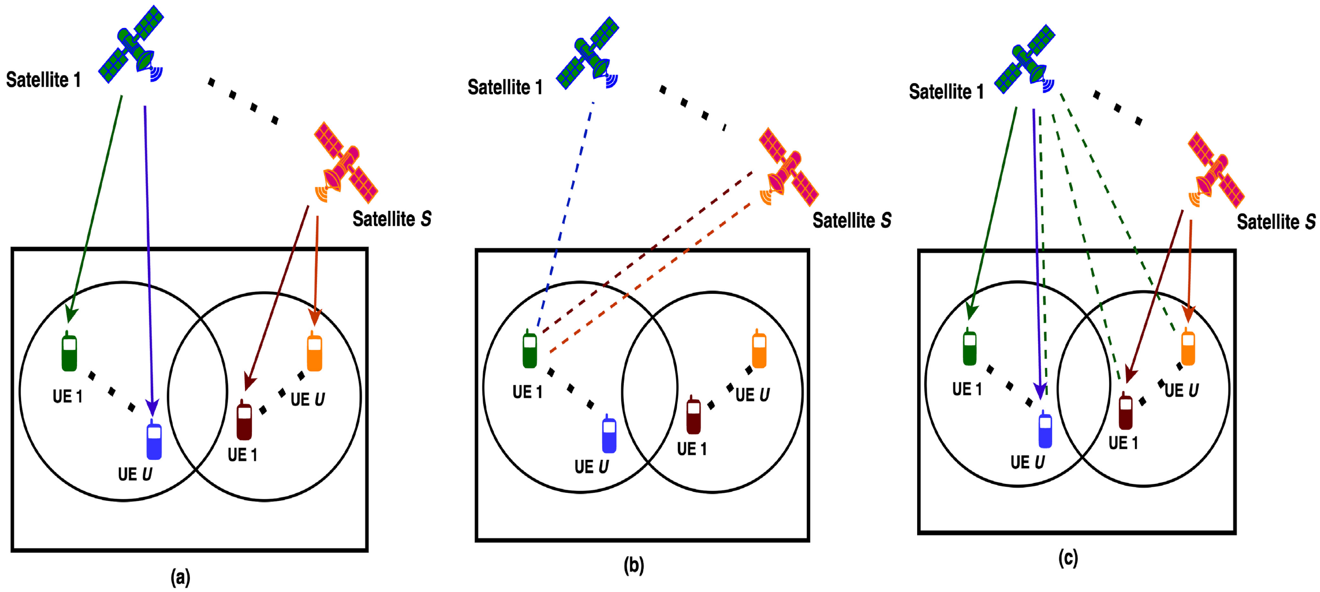

FIGURE 3. Satellite-UE wireless network graph modeling at time slot t: (a) modeling direct links from satellite to UE, (b) modeling interfering links to the UE of intended link from nearby satellites, (c) modeling interfering links from the satellite of intended link to nearby UE.

Sym_n^{++}, and are incorporated using recurrent statistics Ψ_q^t. Recurrent statistics refers to statistical measures that are continuously updated over time. In this case, recurrent statistics would represent how current states depend on past SPD points, capturing long-term dependencies between SPD points over time. Geodesic distances between SPD points, which reflect the temporal dependencies among satellite-UE pairs at consecutive time slots, can be measured using Riemannian metrics like the Stein metric [60]. Utilizing the Stein metric, the distance between $\mathbf{S}_q^{t-1}$ and $\mathbf{S}_q^t$ can be computed as

$$\mathcal{Y}\left(\mathbf{S}_q^{t-1}, \mathbf{S}_q^t\right) = \sqrt{\log\det\left(\frac{\mathbf{S}_q^{t-1} + \mathbf{S}_q^t}{2}\right) - \frac{1}{2}\log\det\left(\mathbf{S}_q^{t-1} \cdot \mathbf{S}_q^t\right)}, \tag{7}$$

where the log det function represents the logarithm of the determinant of $\mathbf{S}_q^{t-1}$ and $\mathbf{S}_q^t$, capturing the distance metric as a measure of changes within the local geometry of Riemannian manifolds of SPD points.

The recurrent statistics at time slot t depend on the SPD points from previous time slots. Therefore, Ψ_q^t is not only a function of $\mathbf{S}_q^t$ but also of $\mathbf{S}_q^{t-1}$, which in turn depends on $\mathbf{S}_q^{t-2}$, and so forth. These dependencies are represented by the exponential moving average of recurrent statistics at time slot t (i.e., summary statistics, ξ_q^t). In other words, summary statistics provide a way to understand and analyze how the position of satellites changes over time as observed from specific ground locations. To compute summary statistics utilizing the weighted Frechét mean, we can use the following [61]

$$\left(\xi_q^t\right)^\kappa = \operatorname*{argmin}_{\xi} \sum_{t=1}^{T} \kappa \mathcal{Y}^2\left(\left(\xi_q^{t-1}\right)^\kappa, \Psi_q^t\right), \quad \forall\kappa \in \mathbb{J}, \tag{8}$$

where $\mathbb{J}$ denotes the set of different time scales, with κ as the scaling parameter and $\kappa \in [0, 1)$.

To predict link-states over N_T consecutive time slots ahead, indicating activity or inactivity, we utilize summary statistics. The complete time frame T is partitioned using the sliding window technique [10] into two stages: one for training and the other for testing. Both the training and testing stages for link-state prediction necessitate multiple steps of the algorithm, as summarized in Algorithm 2 and illustrated in Figure 4.

1) TRAINING STAGE

As we see in Figure 4, the network of each link is captured in the sequential local graph as SPD points over the Riemannian manifold at time slot t. The model receives these points as input features, and for training the supervised model, we utilize the scheduling decisions as targets generated by the optimal FPLinQ [12] scheduler which operates with a predefined limit on the maximum number of iterations.

The algorithm inputs features from N input window of size μ into the model, concurrently updating the recurrent unit Ψ_K^t by calculating summary statistics $(\xi_K^t)^\kappa$ as in eq. (8). It then predicts a continuous link scheduling decision variable $\mathcal{Q}_K^t$ for the K-th link at time slot t_{train}. Similarly, the process continues for time slot $t_{train}+1$ and so on. Using the sigmoid activation function, the algorithm converts the continuous outputs into discrete values to predict binary scheduling decisions. An additional rounding function is applied to convert the output into decimal values, either 1 or 0, where 1 indicates link activation and 0 represents inactivity. The model then compares the predicted scheduling decisions with the target values. Subsequently, the recurrent unit calculates the error and adjusts the trainable parameters accordingly. The algorithm continues to repeat these steps, predicting N_T

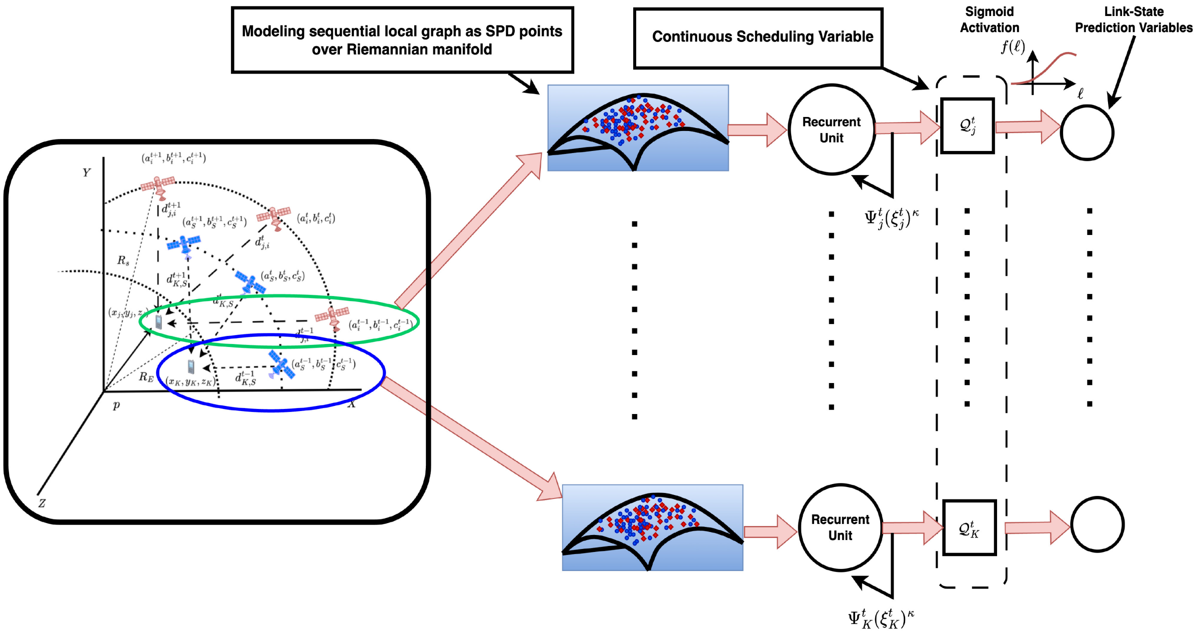

FIGURE 4. **Satellite-UE link-state prediction using RNN over Riemannian manifolds model utilizing individual recurrent unit for each link at time slot *t*.**

Algorithm 2: RNN Over Riemannian Manifolds-Based Link-State Prediction Algorithm

Data: SPD matrices $\mathbf{X}$, FP scheduling labels $\mathbf{Y}$
Input: Training SPD matrices $\mathbf{X}_{\text{train}}$, Testing SPD matrices $\mathbf{X}_{\text{test}}$, Training labels $\mathbf{Y}_{\text{train}}$, Testing labels $\mathbf{Y}_{\text{test}}$
Output: Predicted scheduling decisions $\hat{\mathbf{Y}}$

1. Initialize Model Parameters:
`S1`: Define network parameters, including number of links, sequence length, and batch size.
`S2`: Initialize SPD-SRU cells for recurrent processing.

2. Load Data:
`S3`: Load the SPD matrices and FP scheduling labels.
`S4`: Split the data into training and testing sets.

3. Reshape Data:
`S5`: Reshape $\mathbf{X}_{\text{train}}$ and $\mathbf{X}_{\text{test}}$ into sequences for RNN input.
`S6`: Reshape $\mathbf{Y}_{\text{train}}$ and $\mathbf{Y}_{\text{test}}$ to match the target format for link scheduling.

4. Training Procedure:
for $epoch = 1$ **to** $\mathcal{E}$ **do**
  `S7`: Pass training SPD matrices $\mathbf{X}_{\text{train}}$ and labels $\mathbf{Y}_{\text{train}}$ through the RNN.
  `S8`: Calculate the `softmax_cross_entropy_with_logits` loss between predicted and actual labels.
  `S9`: Update weights using an optimizer like `Adadelta`.
end

5. Evaluation:
`S10`: Evaluate model performance on $\mathbf{X}_{\text{test}}$ and $\mathbf{Y}_{\text{test}}$.
`S11`: Calculate prediction accuracy using the `sigmoid` outputs and comparison with true labels.

6. Save Results:
`S12`: Save the predicted scheduling decisions for further analysis.

consecutive decisions for training batches, N_{train} of size μ, i.e., windows $\{W_1, \ldots, W_{N_{train}}\}$, until the process is complete.

2) TESTING STAGE

During the testing stage, feature SPD matrices from the other portion of the time frame partition are introduced in batches, N_{test}, each with the same size μ, i.e., windows$\{W_{N_{train}+1}, \ldots, W_{N_{test}}\}$. At this stage, predictions of N_T successive time slots ahead rely solely on the summary statistics, with no updates to the weights or biases. The rest of the algorithm follows the same procedure as in the training stage. The scalability of this recurrent satellite-UE network enables it to efficiently accommodate an increasing number of links, while its dynamic nature allows it to adapt to changing network conditions over time.

B. LINK-STATE PREDICTION WITH CNN OVER RIEMANNIAN MANIFOLDS

CNN focuses on learning spatial patterns within the SPD matrices over time, effectively capturing the interference each link causes to and receives from its neighbors across each network topology. Unlike RNN, which is designed to model temporal dependencies, CNN does not require information about the sequential time-correlated movements of satellites to make accurate predictions. When it comes to capturing input data structure, CNN excels at extracting features from two-dimensional matrix-shaped data, with SPD matrices being a prime example of such data [62]. This gives CNN a strong advantage in processing SPD matrices effectively [63]. Link-state prediction process with CNN over Riemannian manifolds is summarized in Algorithm 3 and illustrated in Figure 5.

1) STRUCTURE OF CNN

As shown in Figure 5, we employ a 2D convolutional layer (`Conv2D`) with a set kernel size and trainable filters, applied over input SPD matrices. This layer detects local spatial features by sliding the filters across each SPD matrix, capturing key interference patterns between

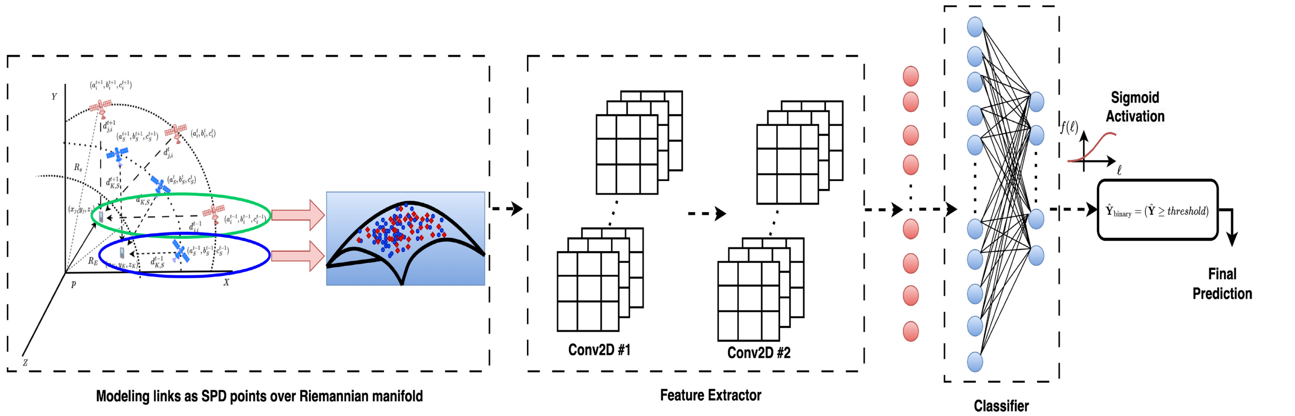

FIGURE 5. Satellite-UE link-state prediction using CNN over Riemannian manifolds model utilizing spatial correlation among all links at time slot *t*.

links. Next, a `MaxPooling` layer downsamples the spatial dimensions of the feature maps while capturing the dominant interference patterns essential for accurate scheduling decisions. Subsequently, a `BatchNormalization` layer is applied, stabilizing and standardizing activations to account for variations across batches and speed up convergence. This process is followed by `Dropout` layers that randomly deactivate a portion of neurons to prevent overfitting.

The network also includes a second `Conv2D` layer, which further processes the refined features from the SPD matrices, supported by another sequence of `MaxPooling`, `BatchNormalization`, and `Dropout` layers. After the convolutional layers, the feature maps are flattened into a 1D vector by the `Flatten` layer, then passed to a fully connected `dense layer` with ReLU activation. This layer combines the extracted features to predict link states, effectively making scheduling decisions based on the observed interference structure.

The network's predictions are compared to true scheduling labels (from FPLinQ), and the error, computed via a `sigmoid_cross_entropy_with_logits` loss function, is backpropagated to adjust the CNN's weights. This iterative process, optimized by the `Adam` optimizer, allows the CNN to converge toward accurate scheduling predictions that align with the FPLinQ benchmark.

2) TRAINING PROCESS

The CNN model is trained with a fixed number of epochs. Similar to RNN, the training batches, N_{train} of window size μ is used, which means the model processes μ samples of SPD points at a time before updating its weights. The true link-state decisions, labeled as the target outputs and derived from the FPLinQ benchmark, serve as the ground truth for supervised learning. These target labels are included in each batch, allowing CNN to compare its predictions against the actual scheduling decisions.

3) PREDICTION PROCESS

After the training stage, the CNN model makes predictions on test batches, N_{test} of the same window size μ to predict the scheduling decision of N_T successive time slots ahead. We set a threshold level for the predicted output to convert them into binary decisions (either 0 or 1) for link scheduling.

Algorithm 3: CNN Over Riemannian Manifolds-Based Link-State Prediction Algorithm

Data: SPD matrices $\mathbf{X}$, FP scheduling labels $\mathbf{Y}$
Input: Training SPD matrices $\mathbf{X}_{\text{train}}$, Testing SPD matrices $\mathbf{X}_{\text{test}}$, Training labels $\mathbf{Y}_{\text{train}}$, Testing labels $\mathbf{Y}_{\text{test}}$
Result: Predicted scheduling decisions $\hat{\mathbf{Y}}$
1. Prepare Training and Testing Data:
for $i = 1$ **to** N_{graphs} **do**
  `S1:` Extract SPD matrices $\mathbf{X_i}$ from $\mathbf{X}$ for N_{links} links.
  `S2:` Extract labels $\mathbf{Y_i}$ from $\mathbf{Y}$.
end
`S3:` Split data into training and testing sets $\mathbf{X}_{\text{train}}, \mathbf{X}_{\text{test}}, \mathbf{Y}_{\text{train}}, \mathbf{Y}_{\text{test}}$.
2. Reshape and Normalize Data:
`S4:` Reshape $\mathbf{X}_{\text{train}}$ and $\mathbf{X}_{\text{test}}$ to match the CNN input requirements.
`S5:` Normalize the input data using a standard scaling method.
3. Training Loop:
for $epoch = 1$ **to** $\mathcal{E}$ **do**
  `S6:` Train on the training data $\mathbf{X}_{\text{train}}, \mathbf{Y}_{\text{train}}$ with a defined batch size and validation split.
end
4. Prediction and Evaluation:
`S7:` Predict $\hat{\mathbf{Y}} = \text{model.predict}(\mathbf{X}_{\text{test}})$.
`S8:` Apply a threshold to convert predictions into binary decisions.
`S9:` Compute the accuracy between the predicted and true labels.
5. Save Results:
`S10:` Save the binary predictions for further analysis.
return

V. PERFORMANCE EVALUATION AND DISCUSSION

This section outlines the performance of our proposed machine learning-based link-state prediction to evaluate its sum rate performance when compared to other scheduling benchmarks. The simulation outcomes are acquired using the MATLAB R2022b and Python 3.9 platform with the Win11 system, the processor: Intel(R) Core(TM) i9-12900K

CPU @ 3.20 GHz, the RAM: 64.0 GB, and the system type: 64-bit operating system.

A. DATASET

We utilize a real-world dataset gathered from 16 ground station-satellite pairs operating in the X-band, the most widely used downlink band for Earth imagery satellites today. The dataset incorporates augmented weather data, including precipitation intensity, precipitation probability, and cloud cover, obtained via the Dark Sky weather API [64]. This augmentation allows the dataset to inherently capture the effects of varying environmental conditions on satellite-ground communications.

The dataset includes data from ground stations located in Wisconsin, Hawaii, Antarctica, Guam, and Florida. For our analysis, we focus on data from Florida involving four satellite links: NOAA-20/JPSS1, AQUA, TERRA, and SNPP, with available elevation and azimuth angles recorded at different time slots.

B. SIMULATION SETUP & DESIGN PARAMETERS

In this work, considering the available dataset, we introduce a total of four ($S = 4$) satellites whose positions are determined by the elevation and azimuth angles for each time slot. Besides, we randomly generate locations of UE within a field length of 20,000 to 35,000 km, considering the total coverage region of all four satellites. In addition, we also set up uniform footprint radii of satellites ranging between 1000 to 4000 km. We assume a finite time frame of $T = \mu + 2N - 1$ time slots, where we use N input window of size μ in the training stage, $N_{train}\{1, \ldots, \mu + N - 1\}$ and N input window of same size μ in testing stage, $N_{test}\{\mu + N, \ldots, \mu + 2N - 1\}$. So, the dataset is used to determine the positions of satellites over T time slots, resulting in a network of T distinct layouts. To gain insight into the behavior of various link scheduling schemes, we consider that 20 UEs are served within the overall field length of four satellites, which corresponds to a total of 20 links. However, the overall model remains scalable and can accommodate any number of links within the dynamic network. The rest of the simulation parameters are presented in Table 1.

In our experiment, we compare the proportion of link activation and the average sum rate performance achieved by the trained RNN and CNN over Riemannian manifolds models with each of the following benchmarks.

- *FPLinQ:* This fractional programming-based algorithm iterates 100 times to optimize link scheduling.
- *All active:* This heuristic activates all available links simultaneously, disregarding interference and power constraints.
- *Random:* Each link is scheduled with a 50% probability, resulting in a randomly selected subset of active links in each instance.
- *Strongest link:* Links are prioritized based on direct channel strength, with a fixed proportion of the strongest links being activated. The optimal percentage of active links is chosen based on the average activation ratio observed in the FPLinQ target.

TABLE 1. Simulation parameters.

Parameter	Symbol	Value
Operating frequency	f_c	12 GHz
LEO satellite altitudes	γ	$\{705, 825, 705, 825\}$km
Satellite transmit antenna gain	G_t	30 dBi
Receiver antenna gain	G_r	37.2 dBi
Transmit power	P_τ	20 dBW
Bandwidth	B	400 MHz
Noise spectral density		-134 dBm/Hz
Gaseous absorption factor	A_g	2 dB
Cloud or fog absorption factor	A_c	1 dB

TABLE 2. RNN over Riemannian manifolds model design parameters.

Parameter	Value
Number of recurrent unit	01/satellite-UE link
Batch size	22
Number of batches	8
Scales used for summary statistics	$\mathbb{J} = \{0.01, 0.25, 0.50, 0.90, 0.99\}$

TABLE 3. CNN over Riemannian manifolds model design parameters.

Block	Layers	Size and Kernel	Activation
Feature Extractor	Conv2D #1	$32 \times (3 \times 3)$	ReLU
	Max-pool #1	(2×2)	-
	Conv2D #2	$64 \times (3 \times 3)$	ReLU
	Max-pool #2	(2×2)	-
Classifier	Fully Connected	1×128	sigmoid

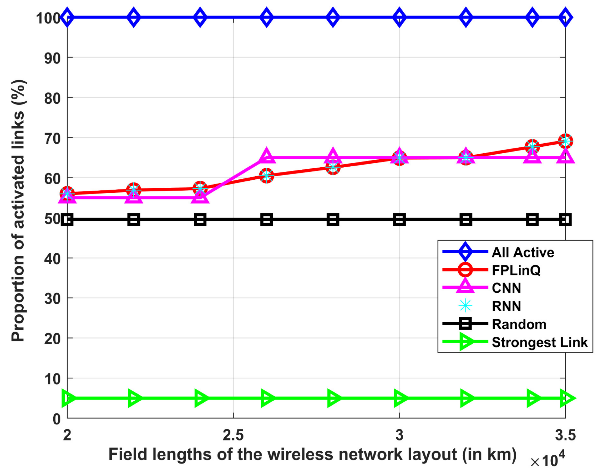

FIGURE 6. Different field lengths versus proportion of activated links when footprint radii of satellites are fixed at 4000 km.

For the case of RNN over Riemannian manifolds, we use a single recurrent unit for predicting link-state of each satellite link and the weighted combinations of the summary statistics of the previous time are computed in the recurrent unit using different time scales. For CNN over Riemannian manifolds, a rectified linear unit (ReLU) is used as an activation function at each neuron in the hidden layers. Besides these, other design parameters for RNN and CNN based models are included in Table 2 and Table 3, respectively.

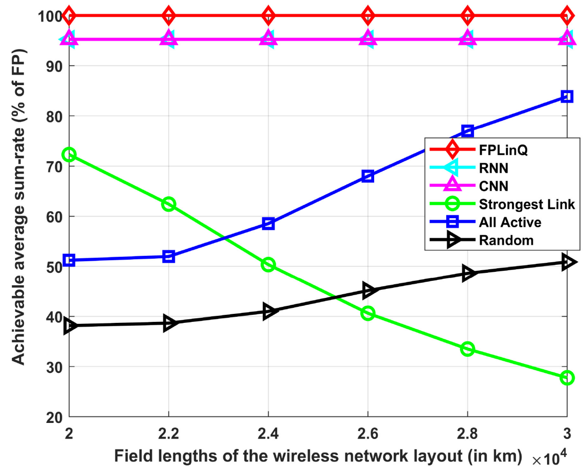

FIGURE 7. Various field lengths versus average sum rate achieved (as % of FP) with different scheduling methods when footprint radii is fixed at 4000 km.

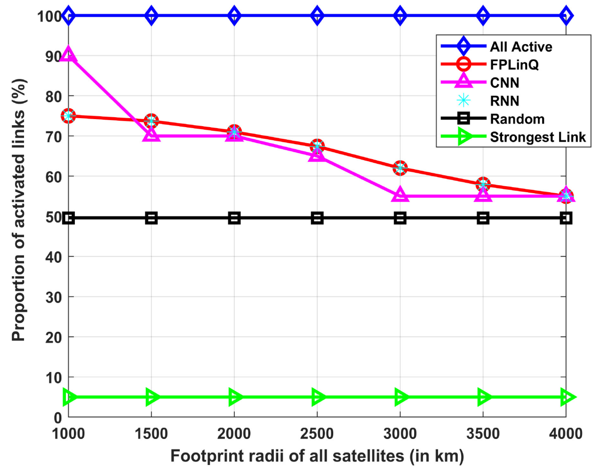

FIGURE 8. Different footprint radii of all satellites versus proportion of activated links when field length is fixed at 20×10^3 km.

C. SIMULATION OUTCOMES

We first start by analyzing the activation proportion (scheduling ratios) across different link densities, which refers to the number of links spread across a given field length when the footprint radii of each satellite held constant at 4000 km. Figure 6 illustrates that both RNN over Riemannian manifolds and CNN over Riemannian manifolds replicate the link activation pattern of the CSI-based FPLinQ [12], while other heuristic methods (i.e., all active, random, and strongest link) do not align with the state-of-the-art performance. Notably, RNN over Riemannian manifolds matches the FPLinQ curve precisely, while CNN over Riemannian manifolds closely follows it. Intuitively, the proportion of link activation is expected to increase as the overall field length of the network expands, indicating a general reduction in interference among the links.

Figure 7 presents a comparison of the average sum rate performance of RNN over Riemannian manifolds and CNN over Riemannian manifolds along with other heuristic methods when the field length of wireless network layout is varied under constant footprint radii of 4000 km for 20 satellite-UE pairs. The results are shown as a percentage of the FPLinQ benchmark. As seen from the figure, both RNN over Riemannian manifolds and CNN over Riemannian manifolds reach as close to 96% of the average sum rate of FPLinQ benchmark. In comparison to these two methods, only ‘all active’ heuristic achieves close to 83% sum rate of FPLinQ benchmark for the maximum considered field length of $30,000$ km. So, it is clear from the average sum rate performance that both neural networks maintain stability even with varying field lengths.

In contrast to the previous result shown in Figure 6, considering 20 links, the activation proportion decreases across varying footprint radii when the field length is fixed at 20,000 km, as depicted in Figure 8. At constant field length, an increase in the footprint radii of each satellite, or the area covered by each satellite, leads to greater

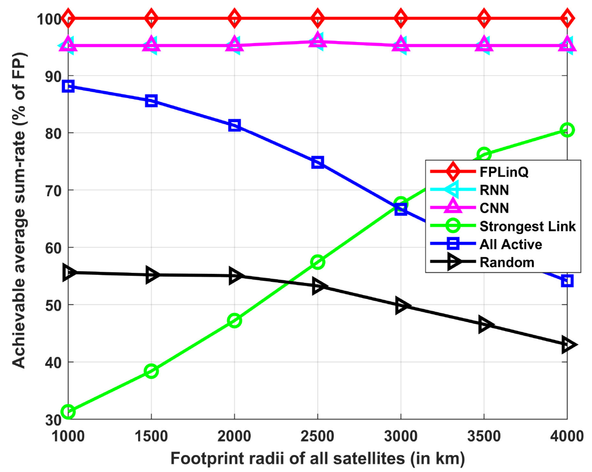

FIGURE 9. Various footprint radii versus average sum rate achieved (as % of FP) with different scheduling methods when field length is fixed at 20×10^3 km.

interference between links. Consequently, the proportion of activated links gradually declines as the footprint radii grow. The RNN over Riemannian manifolds closely follows the exact activation pattern, while the CNN over Riemannian manifolds exhibits a similar pattern to FPLinQ. However, the other heuristic methods fail to achieve the state-of-the-art performance.

Figure 9 illustrates the average sum rate performance of both neural networks with 20 links under varying footprint radii, with a fixed field length of $20,000$ km. Similar to their performance under varying field lengths, RNN over Riemannian manifolds and CNN over Riemannian manifolds consistently achieve over 95% of the average sum rate of the FPLinQ benchmark, even as the footprint radii of all satellites expand. Based on the results from Figures 8 and 9, it is evident that both neural networks can accurately predict scheduling decisions while maintaining performance levels that closely match the FPLinQ benchmark in terms of sum rate. While the ‘strongest link’ heuristic method comes

TABLE 4. Significance of the number of training samples predicting successive time slots ahead when the total number of links is 40.

Training Layouts	20	30	35	40
Time slots ahead	7	7	8	8
Average Sum rate (%) (RNN)	97.52	99.25	97.65	99.25
Average Sum rate (%) (CNN)	98.94	99.39	99.79	99.79

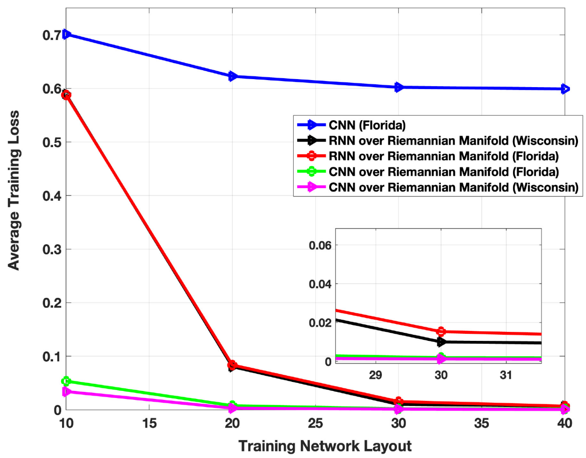

FIGURE 10. Training loss versus training network layout graph for different regional data modeled with and without Riemannian manifold representation.

TABLE 5. Sum rate (as % of FP) comparison for different regional data modeled with and without Riemannian manifold representation.

RNN over Riemannian manifolds (Florida)	RNN over Riemannian manifolds (Wisconsin)	CNN over Riemannian manifolds (Florida)	CNN over Riemannian manifolds (Wisconsin)	CNN (Florida)
95.79	95.43	95.68	95.24	64.45

within 80% of the FPLinQ benchmark at the maximum footprint radius of 4000 km, the other two methods fall short, showing even worse performance in maintaining sum rate as the footprint radii increase.

Table 4 presents the achievable sum rate performance of both models for a total of 40 links. Notably, with just 30 training samples, the RNN over Riemannian manifolds is capable of predicting $N_T = 7$ successive time slots ahead, achieving over 99% of the sum rate compared to the FPLinQ benchmark. Similarly, the CNN over Riemannian manifolds for the same number of links, also predicts $N_T = 7$ consecutive time slots ahead using only 30 training samples. The model also attains more than 99% of the sum rate relative to the FPLinQ benchmark. Since there is no significant improvement in performance with an increasing number of training samples, we can confidently conclude that 30 training samples are sufficient to train both RNN and CNN based models, demonstrating their robustness in maintaining high performance with a limited dataset.

For 20 links, Figure 10 compares the training losses of RNN and CNN models across two regions, Florida and Wisconsin, as the number of training network layouts increases from 10 to 40. The results demonstrate that irrespective of the geographical locations of two ground stations, both RNN over Riemannian manifold and CNN over Riemannian manifold converge within 30 training graphs, highlighting their efficiency in capturing the spatio-temporal characteristics of satellite networks. The RNN over Riemannian manifold models starts with relatively higher training loss at 10 graphs but exhibits a steep decline, stabilizing around 30 network graphs. Similarly, the CNN over Riemannian manifold models begin with lower initial loss and demonstrate a steady decline with fewer oscillations, stabilizing at 30 network graphs.

The consistency of results across two geographical regions, Florida and Wisconsin, highlights the robustness of RNN and CNN models with Riemannian manifold representation in achieving reliable performance. This confirms the sufficiency of only 30 training network graphs and validates the proposed models' ability to generalize effectively to new satellite network layouts while maintaining prediction accuracy.

In contrast, CNN without Riemannian manifold representation begins with a significantly higher initial training loss (~ 0.7) and shows a slower decrease, failing to reach a minimum value as that of Riemannian-based models. This trend suggests that the lack of Riemannian manifold representation limits the model's ability to effectively learn from the structured SPD matrices.

Similarly, the sum rate comparison graph across the dataset for two regions in Table 5 illustrates the same consistency. Both RNN and CNN-based models with Riemannian manifold representation using the dataset of two regions maintain over 95% of the average sum rate of FPLinQ benchmark for 20 links. But, the CNN-based model without manifold representation only manages to achieve 64.45% of average sum rate of the benchmark. These results verify the critical role of Riemannian manifolds in enhancing model performance and highlight the superiority of the proposed approach for link scheduling in dynamic environments.

D. COMPLEXITY ANALYSIS

In this section, we provide the complexity of FPLinQ algorithm first. Then, we examine the complexity of both neural networks in addressing the satellite link scheduling problem. Exploring the computational complexity of the RNN over Riemannian manifolds and then the CNN over Riemannian manifolds, we try to draw a comparison with the FPLinQ benchmark. The complexity of each method is analyzed in terms of its scalability with respect to the number of nodes, $\mathcal{N}$.

1) FPLINQ ALGORITHM

At each iteration, the FPLinQ algorithm's main computational load arises from performing matrix multiplication with the $\mathcal{N} \times \mathcal{N}$ channel coefficient matrix. This operation results in a per-iteration complexity of $\mathcal{O}(\mathcal{N}^2)$. Assuming convergence within a set number of iterations, the total runtime complexity therefore also scales as, $C_F = \mathcal{O}(\mathcal{N}^2)$.

2) RNN OVER RIEMANNIAN MANIFOLDS

The process of link-state prediction with RNN over Riemannian manifolds is twofold. First part involves sequential local graph modeling, where a series of SPD points are accumulated over the Riemannian manifold throughout the entire time frame T. The second part predicts the scheduling decisions for N_T time slots ahead. Since the first part involves accumulating SPD points over consecutive time slots within the overall time frame, the complexity for this part can be calculated as $\mathcal{O}(\mathcal{N}^2)$. In the second part, given a fixed number of iterations with a total of N_{test} batches, each of size μ, and adjustable training parameters $\mathcal{A}$, the complexity is calculated as $\mathcal{O}(\mathcal{N} N_{test} \mu \mathcal{A})$, which simplifies to approximately $\mathcal{O}(\mathcal{N})$ for large network size $\mathcal{N}$ [10]. The recurrent statistics are updated through operations on $\mathcal{N} \times \mathcal{N}$ SPD matrices. Thus, each operation on these matrices incurs an $\mathcal{O}(\mathcal{N}^2)$ computational cost as recurrent statistics overhead. Since operations are not nested within additional loops, the final breakdown of the computational complexity of link scheduling with RNN over Riemannian manifolds is $C_R = \mathcal{O}(\mathcal{N}^2) + \mathcal{O}(\mathcal{N}) + \mathcal{O}(\mathcal{N}^2) \approx \mathcal{O}(\mathcal{N}^2)$.

3) CNN OVER RIEMANNIAN MANIFOLDS

Similar to RNN over Riemannian manifolds, CNN over Riemannian manifolds also follows the same twofold operation. Thus, the first part of accumulation of SPD points corresponds to a complexity of $\mathcal{O}(\mathcal{N}^2)$. Next, assuming the discretized grid of size $\mathcal{K} \times \mathcal{K}$, filter dimension of $\mathcal{L} \times \mathcal{L}$, size of input feature vector to be v_0 for fully connected stage, and $(v_1, v_2, \ldots, v_n)$ representing the number of hidden units for each of the n hidden layers, the total complexity of the neural network can be computed as $\mathcal{O}\{\mathcal{K}^2 \times \mathcal{L}^2 + \mathcal{N} \times (v_0 v_1 + \cdots + v_{n-1} v_n + v_n)\} \approx \mathcal{O}(\mathcal{N})$ [16]. Hence, similar to RNN over Riemannian manifolds, CNN over Riemannian manifolds also has the same overall computational complexity, $C_C = \mathcal{O}(\mathcal{N}^2)$.

Thus, we see that the overall complexity of both neural networks (C_R and C_C) is the same as the CSI-based FPLinQ benchmark (C_F) [12]. However, since the proposed neural network-based models do not rely on CSI, their similar complexity enhances their robustness in achieving a high level of sum rate performance. Both RNN over Riemannian manifolds and CNN over Riemannian manifolds models even outperform those with lower complexity in [16], [65], as they require only a minimal number of training samples to achieve effective performance.

VI. CONCLUSION

In this paper, we propose RNN and CNN over Riemannian manifolds as a low-complexity solution approach to tackle the challenges of link scheduling in satellite networks. To achieve this, we first model the dynamic structure of the satellite network over Riemannian manifolds. We then explore the time and space domains with RNN and CNN, taking advantage of the spatio-temporal correlation that arises from satellite movements. Simulation results reveal that both neural networks achieve over 99% of the sum rate compared to traditional fractional programming-based methods such as the FPLinQ benchmark using only 30 training samples.

ACKNOWLEDGMENT

The authors would like to thank Dr. Rashed Shelim, a Postdoctoral Associate at Virginia Tech, VA, USA, for his help in solving some of the key problems associated with the machine learning coding part of this work.

REFERENCES

[1] H. Zhang, N. Liu, X. Chu, K. Long, A.-H. Aghvami, and V. C. M. Leung, "Network slicing based 5G and future mobile networks: Mobility, resource management, and challenges," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 138–145, Aug. 2017.

[2] H. Dong, C. Hua, L. Liu, W. Xu, S. Guo, and R. Tafazolli, "Joint beamformer design and user scheduling for integrated terrestrial-satellite networks," *IEEE Trans. Wireless Commun.*, vol. 22, no. 10, pp. 6398–6414, Oct. 2023.

[3] Z. Xiang, X. Gao, K.-X. Li, and X.-G. Xia, "Massive MIMO downlink transmission for multiple LEO satellite communication," *IEEE Trans. Commun.*, vol. 72, no. 6, pp. 3352–3364, Jun. 2024.

[4] B. Di, H. Zhang, L. Song, Y. Li, and G. Y. Li, "Ultra-dense LEO: Integrating terrestrial-satellite networks into 5G and beyond for data offloading," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 47–62, Jan. 2019.

[5] M. Y. Abdelsadek, H. Yanikomeroglu, and G. K. Kurt, "Future ultra-dense LEO satellite networks: A cell-free massive MIMO approach," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, 2021, pp. 1–6.

[6] M. Afhamisis and M. R. Palattella, "SALSA: A scheduling algorithm for LoRa to LEO satellites," *IEEE Access*, vol. 10, pp. 11608–11615, 2022.

[7] F. A. Tondo, M. Afhamisis, S. Montejo-Sánchez, O. L. A. López, M. R. Palattella, and R. D. Souza, "Multiple channel LoRa-to-LEO scheduling for direct-to-satellite IoT," *IEEE Access*, vol. 12, pp. 30627–30637, 2024.

[8] J. Zhao, S. Chen, C. Jin, H. Xing, and Y. Chen, "Data scheduling and resource allocation in LEO satellite networks for IoT task offloading," *Wireless Netw.*, vol. 30, pp. 7075–7085, Nov. 2024.

[9] W. Abderrahim, O. Amin, M.-S. Alouini, and B. Shihada, "Latency-aware offloading in integrated satellite terrestrial networks," *IEEE Open J. Commun. Soc.*, vol. 1, pp. 490–500, 2020.

[10] R. Shelim and A. S. Ibrahim, "Wireless link scheduling over recurrent Riemannian manifolds," *IEEE Trans. Veh. Technol.*, vol. 72, no. 4, pp. 4959–4968, Apr. 2023.

[11] N. Naderializadeh and A. S. Avestimehr, "ITLinQ: A new approach for spectrum sharing in device-to-device communication systems," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1139–1151, Jun. 2014.

[12] K. Shen and W. Yu, "FPLinQ: A cooperative spectrum sharing strategy for device-to-device communications," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, 2017, pp. 2323–2327.

[13] J. Yu et al., "Efficient link scheduling in wireless networks under rayleigh-fading and multiuser interference," *IEEE Trans. Wireless Commun.*, vol. 19, no. 8, pp. 5621–5634, Aug. 2020.

[14] I. Budhiraja, N. Kumar, and S. Tyagi, "Deep-reinforcement-learning-based proportional fair scheduling control scheme for underlay D2D communication," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3143–3156, Mar. 2021.

[15] Y. Shen, Y. Shi, J. Zhang, and K. B. Letaief, "A graph neural network approach for scalable wireless power control," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, 2019, pp. 1–6.

[16] W. Cui, K. Shen, and W. Yu, "Spatial deep learning for wireless scheduling," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1248–1261, Jun. 2019.

[17] C. Tatino, N. Pappas, I. Malanchini, L. Ewe, and D. Yuan, "Learning-based link scheduling in Millimeter-wave multi-connectivity scenarios," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2020, pp. 1–6.

[18] Z. Liu, Z. Chen, L. Luo, M. Hua, W. Li, and B. Xia, "Age of information-based scheduling for wireless device-to-device communications using deep learning," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, 2021, pp. 1–6.

[19] I. Nasim and A. S. Ibrahim, "Millimeter wave beamforming codebook design via learning channel covariance matrices over Riemannian manifolds," *IEEE Access*, vol. 10, pp. 119617–119629, 2022.

[20] I. Nasim and A. S. Ibrahim, "Relay placement for maximum flow rate via learning and optimization over Riemannian manifolds," *IEEE Trans. Mach. Learn. Commun. Netw.*, vol. 1, no. 1, pp. 197–209, Aug. 2023.

[21] A. S. Ibrahim, "Wireless link scheduling via interference-aware symmetric positive definite connectivity manifolds," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, 2021, pp. 1–5.

[22] R. Shelim and A. S. Ibrahim, "Geometric machine learning over Riemannian manifolds for wireless link scheduling," *IEEE Access*, vol. 10, pp. 22854–22864, 2022.

[23] R. Shelim and A. S. Ibrahim, "Learning wireless power allocation through graph convolutional regression networks over Riemannian manifolds," *IEEE Trans. Veh. Technol.*, vol. 73, no. 3, pp. 3652–3662, Mar. 2024.

[24] J. J. Sadique, I. Nasim, and A. S. Ibrahim, "Covariance shaping over Riemannian manifolds for massive MIMO communication," *IEEE Access*, vol. 11, pp. 142874–142883, 2023.

[25] J. B. Oliva, B. Póczos, and J. Schneider, "The statistical recurrent unit," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 2671–2680.

[26] M. Khalid, J. Ali, and B.-h. Roh, "Artificial intelligence and machine learning technologies for integration of terrestrial in non-terrestrial networks," *IEEE Internet Things Mag.*, vol. 7, no. 1, pp. 28–33, Jan. 2024.

[27] H. Dahrouj, S. Liu, and M.-S. Alouini, "Machine learning-based user scheduling in integrated satellite-HAPS-ground networks," *IEEE Netw.*, vol. 37, no. 2, pp. 102–109, Mar./Apr. 2023.

[28] M. Chen, Y. Du, K. Tang, L. Xing, Y. Chen, and Y. Chen, "Learning to construct a solution for the agile satellite scheduling problem with time-dependent transition times," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 54, no. 10, pp. 5949–5963, Oct. 2024.

[29] Y. Song, P. N. Suganthan, W. Pedrycz, R. Yan, D. Fan, and Y. Zhang, "Energy-efficient satellite range scheduling using a reinforcement learning-based memetic algorithm," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 60, no. 4, pp. 4073–4087, Aug. 2024.

[30] Z. Li, S. Han, M. Peng, C. Li, and W. Meng, "Dynamic multiple access based on RSMA and spectrum sharing for integrated satellite-terrestrial networks," *IEEE Trans. Wireless Commun.*, vol. 23, no. 6, pp. 5393–5408, Jun. 2024.

[31] R. Liu, K. Guo, K. An, S. Zhu, and H. Shuai, "NOMA-based integrated satellite-terrestrial relay networks under spectrum sharing environment," *IEEE Wireless Commun. Lett.*, vol. 10, no. 6, pp. 1266–1270, Jun. 2021.

[32] C. Zhang, C. Jiang, J. Jin, S. Wu, L. Kuang, and S. Guo, "Spectrum sensing and recognition in satellite systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 2502–2516, Mar. 2019.

[33] X. Ding, T. Ni, Y. Zou, and G. Zhang, "Deep learning for satellites based spectrum sensing systems: A low computational complexity perspective," *IEEE Trans. Veh. Technol.*, vol. 72, no. 1, pp. 1366–1371, Jan. 2023.

[34] Z. Liu, J. Liu, X. Liu, W. Yang, J. Wu, and Y. Chen, "Knowledge-assisted adaptive large neighbourhood search algorithm for the satellite–ground link scheduling problem," *Comput. Ind. Eng.*, vol. 192, Jun. 2024, Art. no. 110219.

[35] C. Deprez and G. Giorgi, "Operational envelope and link scheduling for inter-satellite links in next-generation GNSSs," in *Proc. IEEE Aerosp. Conf.*, 2021, pp. 1–13.

[36] O. Kondrateva, H. Döbler, H. Sparka, A. Freimann, B. Scheuermann, and K. Schilling, "Throughput-optimal joint routing and scheduling for low-earth-orbit satellite networks," in *Proc. 14th Annu. Conf. Wireless On-Demand Netw. Syst. Services (WONS)*, 2018, pp. 59–66.

[37] D. Yang, J. Yang, and P. Xu, "Timeslot scheduling of inter-satellite links based on a system of a narrow beam with time division," *GPS Solut.*, vol. 21, pp. 999–1011, Jul. 2017.

[38] Y. Song, J. Ou, J. Wu, Y. Wu, L. Xing, and Y. Chen, "A cluster-based genetic optimization method for satellite range scheduling system," *Swarm Evol. Comput.*, vol. 79, Jun. 2023, Art. no. 101316.

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

[40] J. Zhang and L. Xing, "An improved genetic algorithm for the integrated satellite imaging and data transmission scheduling problem," *Comput. Oper. Res.*, vol. 139, Mar. 2022, Art. no. 105626.

[41] Z. Zhou, E. Chen, F. Wu, Z. Chang, and L. Xing, "Multi-satellite scheduling problem with marginal decreasing imaging duration: An improved adaptive ant colony algorithm," *Comput. Ind. Eng.*, vol. 176, Feb. 2023, Art. no. 108890.

[42] Y. Qi, L. Yang, C. Pan, C. Chi, and Q. Huang, "A flexible topology reconstruction strategy based on deep Q-learning for balance performance and efficiency of STINs," *IEEE Trans. Netw. Service Manag.*, vol. 20, no. 2, pp. 1051–1064, Jun. 2023.

[43] R. Wang, W. Zhu, R. Ma, Y. Zhang, G. Liu, and W. Kang, "Inter-satellite link scheduling and power allocation method for satellite networks," *Wireless Netw.*, vol. 30, pp. 5547–5558, Aug. 2024.

[44] R. M. Ferre and E. S. Lohan, "Comparison of MEO, LEO, and terrestrial IoT configurations in terms of GDOP and achievable positioning accuracies," *IEEE J. Radio Freq. Ident.*, vol. 5, pp. 287–299, 2021.

[45] N. Brown, B. Arguello, L. Nozick, and N. Xu, "A heuristic approach to satellite range scheduling with bounds using lagrangian relaxation," *IEEE Syst. J.*, vol. 12, no. 4, pp. 3828–3836, Dec. 2018.

[46] X. Peng, H. Du, C. Cao, and M. Chen, "Flexible user mapping and resource allocation for enhanced system capacity in multi-beam GEO satellite systems," in *Proc. IEEE 100th Veh. Technol. Conf. (VTC)*, 2024, pp. 1–6.

[47] H. Sun, P. Stolf, and J.-M. Pierson, "Spatio-temporal thermal-aware scheduling for homogeneous high-performance computing datacenters," *Future Gener. Comput. Syst.*, vol. 71, pp. 157–170, Jun. 2017.

[48] V. Wattin Håkansson, N. K. D. Venkategowda, S. Werner, and P. K. Varshney, "Optimal transmission-constrained scheduling of Spatio-temporally dependent observations using age-of-information," *IEEE Sensors J.*, vol. 22, no. 15, pp. 15596–15606, Aug. 2022.

[49] A. R. de Miranda, T. M. G. de Andrade Barbosa, A. G. S. Conceição, and S. G. S. Alcalá, "Recurrent neural network based on statistical recurrent unit for remaining useful life estimation," in *Proc. 8th Brazil. Conf. Intell. Syst. (BRACIS)*, 2019, pp. 425–430.

[50] X. S. Nguyen, L. Brun, O. Lézoray, and S. Bougleux, "Learning recurrent high-order statistics for skeleton-based hand gesture recognition," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, 2021, pp. 975–982.

[51] E. Juan, M. Lauridsen, J. Wigard, and P. E. Mogensen, "5G new radio mobility performance in LEO-based non-terrestrial networks," in *Proc. IEEE Globecom Workshops (GC Wkshps*, 2020, pp. 1–6.

[52] K. An et al., "Exploiting multi-layer refracting RIS-assisted receiver for HAP-SWIPT networks," *IEEE Trans. Wireless Commun.*, vol. 23, no. 10, pp. 12638–12657, Oct. 2024.

[53] K. An, M. Lin, J. Ouyang, and W.-P. Zhu, "Secure transmission in cognitive satellite terrestrial networks," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 11, pp. 3025–3037, Nov. 2016.

[54] A. Grami, *Introduction to Digital Communications*. Waltham, MA, USA: Academic Press, 2015.

[55] L. Dodero, H. Q. Minh, M. S. Biagio, V. Murino, and D. Sona, "Kernel-based classification for brain connectivity graphs on the Riemannian manifold of positive definite matrices," in *Proc. IEEE 12th Int. Symp. Biomed. Imag. (ISBI)*, 2015, pp. 42–45.

[56] J. M. Lee, *Introduction to Riemannian Manifolds*, vol. 2. Cham, Switzerland: Springer, 2018.

[57] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2012.

[58] I. Sutskever, "Sequence to sequence learning with neural networks," 2014, *arXiv:1409.3215*.

[59] S. Zhang, W. Shen, M. Zhang, X. Cao, and Y. Cheng, "Experience-driven wireless D2D network link scheduling: A deep learning approach," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2019, pp. 1–6.

[60] H. Salehian, G. Cheng, B. C. Vemuri, and J. Ho, "Recursive estimation of the stein center of SPD matrices and its applications," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 1793–1800.

[61] R. Chakraborty et al., "A statistical recurrent model on the manifold of symmetric positive definite matrices," in *Proc. 32nd Conf. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–12.

[62] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 25, 2012, pp. 1–9.

[63] L. Chen, Z. Yu, and J. Yang, "SPD-CNN: A plain CNN-based model using the symmetric positive definite matrices for cross-subject EEG classification with Meta-transfer-learning," *Front. Neurorobot.*, vol. 16, Aug. 2022, Art. no. 958052.

[64] D. Vasisht, J. Shenoy, and R. Chandra, "L2D2: Low latency distributed downlink for LEO satellites," in *Proc. ACM SIGCOMM Conf.*, 2021, pp. 151–164.

[65] M. Lee, G. Yu, and G. Y. Li, "Graph embedding-based wireless link scheduling with few training samples," *IEEE Trans. Wireless Commun.*, vol. 20, no. 4, pp. 2282–2294, Apr. 2021.

JOARDER JAFOR SADIQUE (Graduate Student Member, IEEE) received the B.Sc. (Hons.) and M.Sc. degrees in applied physics and electronic engineering (currently named as, electrical and electronic engineering) from the University of Rajshahi, Rajshahi, Bangladesh, in 2010 and 2011, respectively. He is currently pursuing the Ph.D. degree with electrical and computer engineering department, Florida International University, Miami, FL, USA. He has been serving as an Assistant Professor with the Electrical and Electronic Engineering Department, Begum Rokeya University, Rangpur, Bangladesh, since October 2017 (on study leave). He was also a Faculty Member with the Electrical and Electronic Engineering Department, University of Information Technology and Sciences, Dhaka, Bangladesh, from May 2013 to March 2014. His main research interests include unmanned aerial vehicle-based millimeter wave communications, intelligent reflecting surface-assisted wireless networks, cooperative communications, and modern channel coding techniques.

IMTIAZ NASIM (Member, IEEE) received the B.S. degree in electrical and electronic engineering from the Khulna University of Engineering and Technology, Bangladesh, in 2015, the M.S. degree in electrical engineering from Georgia Southern University, Statesboro, GA, USA, in 2019, and the Ph.D. degree in electrical and computer engineering from Florida International University, Miami, FL, USA, in 2023. He is currently a Research Scientist with the Idaho National Laboratory, Idaho Falls, ID, USA. His research interests include vehicle-to-everything communications, reliability of autonomous vehicles with machine learning, human exposure to electromagnetic field in communications systems, and millimeter wave communications. He serves as a Reviewer for multiple journals and conferences, including IEEE TRANSACTIONS OF MOBILE COMPUTING, IEEE ACCESS, and IEEE GLOBECOM.

AHMED S. IBRAHIM (Senior Member, IEEE) received the B.S. and M.S. degrees in electronics and electrical communications engineering from Cairo University, Cairo, Egypt, in 2002 and 2004, respectively, and the Ph.D. degree in electrical engineering from the University of Maryland at College Park, College Park, MD, USA, in 2009. He is currently an Associate Professor with the Electrical and Computer Engineering Department, Florida International University (FIU), Miami, FL, USA. Prior to joining FIU, he was an Assistant Professor with Cairo University, a Wireless Research Scientist with Intel Corporation, and a Senior Engineer with Inter Digital Communications Inc. His research interests include various topics of wireless systems including drone-assisted millimeter wave communications, vehicular communications, and rethinking wireless networks through the lens of Riemannian geometry. He was a recipient of the NSF CAREER Award in 2022.