Split Learning-Based Channel Prediction for 6G-Enabled LEO Satellite Systems

Kithmini Weththasinghe, *Graduate Student Member, IEEE*, Quynh Tu Ngo, *Senior Member, IEEE*, Ying He, *Senior Member, IEEE*, Beeshanga Jayawickrama, *Senior Member, IEEE*

***Abstract*—In sixth-generation communications, non-terrestrial networks provide scalable solutions to bridge connectivity gaps, particularly in areas where terrestrial infrastructure is lacking. Low Earth orbit (LEO) satellites are central to this effort due to their reduced signal propagation delays, which enable real-time, mission-critical communications, especially during emergencies when ground infrastructure is compromised. In such dynamic environments, accurate channel prediction is essential to maintaining consistent performance under varying conditions. However, conventional channel prediction methods that rely on uplink-to-downlink reciprocity, common in terrestrial time-division duplexing systems, are ineffective in LEO scenarios due to rapid channel variations, Doppler shifts, and channel aging. These challenges are intensified during unpredictable environmental conditions, such as natural disasters. To address this, we propose a hybrid offline-online learning framework for real-time downlink channel prediction based on uplink data in LEO systems. This approach enables an adaptive response to dynamic communication conditions. Additionally, a split learning strategy distributes computational load between satellites and user equipment, avoiding overloading LEO platforms. Simulation results demonstrate that the proposed model achieves lower normalised mean square error across various LEO configurations. The results validate the effectiveness of the proposed model under dynamic 6G channel conditions.**

***Index Terms*—Machine learning, channel prediction, low Earth orbit satellite, non-terrestrial network, split learning**

I. Introduction

Sixth-generation (6G) technology is set to revolutionise wireless communication by delivering unparalleled performance in terms of speed, reliability, and coverage. It aims to achieve extremely high data rates, ultra-low latency, and the ability to support a massive number of connected devices, paving the way for a wide range of advanced applications and use cases across industries. Achieving these ambitious goals requires a shift from traditional terrestrial-only network designs to hybrid architectures that incorporate non-terrestrial networks (NTNs) [1], ensuring connectivity even in the most remote and underserved areas. By leveraging communication platforms such as satellites and high-altitude platforms, NTNs provide flexible, scalable solutions to bridge connectivity gaps. These networks ensure redundancy and enhance resilience, particularly in disaster recovery scenarios or regions where terrestrial deployments are geographically unfeasible.

The authors are with the School of Electrical and Data Engineering, University of Technology Sydney, Sydney, NSW 2007, Australia (Email: {Kithmini.WeththasingheArachchige, QuynhTu.Ngo, Ying.He, Beeshanga.Jayawickrama}@uts.edu.au).

This work has been supported by the SmartSat CRC, whose activities are funded by the Australian Government's CRC Program.

3GPP release 17 introduces standardised support for NTNs, enabling direct 5G new radio (NR) communication between satellites and handheld user equipment (UE), within the sub-6 GHz frequency range (FR1) [2]. This release addresses the distinct physical-layer and system-level challenges posed by satellite communication, such as increased propagation delays, Doppler effects, and large coverage footprints, by enhancing synchronisation procedures, uplink timing advance mechanisms, and UE positioning capabilities. Release 17 specifies normative adaptations to the NR protocol stack to ensure reliable operation over satellite channels, thereby supporting service continuity and extending 5G coverage to remote and underserved regions lacking terrestrial infrastructure. Terrestrial 5G NR systems utilise time-division duplexing (TDD) for operation [3]. To ensure complete and seamless integration between terrestrial networks (TNs) and NTNs, 3GPP release 17 defines TDD mode as the operational basis for NTNs [2].

Satellites play a pivotal role in NTNs by delivering wide-area coverage and complementing terrestrial networks in a hybrid architecture. Geostationary orbit (GEO) satellites offer expansive coverage but suffer from higher latency, while low Earth orbit (LEO) satellites address this limitation with significantly lower propagation delays [1], [4]. This versatility allows satellites to support critical 6G applications. LEO satellites play a crucial role in meeting the low-latency and high-bandwidth demands of 6G networks. Their proximity to Earth significantly reduces signal propagation delays compared to the GEO satellites, making them particularly effective for real-time, mission-critical applications.

LEO satellite communication systems play a crucial role in emergency response scenarios [5], [6], particularly when terrestrial infrastructure is damaged or unavailable due to natural disasters or large-scale crises. Their ability to rapidly provide wide-area coverage enables reliable data transmission and continuous connectivity, which are essential for coordinating disaster management and recovery operations. Recent advances in LEO satellite constellations, including adaptive beamforming, inter-satellite links, and flexible payloads, further enhance network resilience by supporting real-time communication and dynamic resource allocation [7], [8]. These features ensure robust connectivity even in remote or inaccessible regions, positioning LEO satellites as a critical component of next-generation emergency communication networks.

In such time-critical environments, LEO satellite channel prediction for beamforming becomes a key enabling technology. Beamforming allows satellites to efficiently direct transmission power toward affected areas, improving spectral

efficiency while reducing interference [9]. Accurate prediction of channel conditions, such as path loss, shadowing, and atmospheric attenuation, is essential for effective beamforming, particularly under the rapidly changing propagation environments characteristic of disaster scenarios. By leveraging real-time measurements and predictive algorithms, satellite systems can dynamically adapt their beam patterns to maintain high-quality and stable links for emergency responders, even when ground-based networks are compromised [10]. The unique dynamics of LEO satellite systems further complicate channel prediction. The high orbital velocity of LEO satellites, combined with long propagation delays between the satellite and user equipment (UE), causes the channel state information (CSI) available at the satellite to rapidly become outdated. While terrestrial base stations often rely on uplink-to-downlink channel reciprocity under time-division duplexing (TDD) operation, this assumption is invalid mainly in LEO systems. Severe Doppler shifts and fast time-varying channels lead to pronounced channel aging, significantly degrading the accuracy of downlink CSI inferred from uplink measurements [8]. As a result, conventional terrestrial channel prediction techniques are insufficient, motivating the development of satellite-specific prediction and beamforming strategies tailored to the highly dynamic LEO environment [7].

A. Related Work

Recent studies have applied machine learning to improve channel prediction accuracy in dynamic wireless systems. Long short-term memory (LSTM)-based channel prediction models have been proposed in [7], [11]–[13]. In [11], an LSTM-based deep learning channel prediction model was proposed to address channel aging in LEO satellite systems. While the work acknowledges the difficulty of obtaining accurate instantaneous CSI in the presence of Doppler shifts and long propagation delays, it proposes a data-driven solution that exploits temporal channel correlations to predict future CSI. The model was trained offline, which may not be feasible in dynamic and highly variable satellite environments. In [12], a channel prediction framework was proposed for LEO satellite communications, addressing the challenge of channel aging through an LSTM-based architecture. The proposed method predicts the channel for the next time slot by leveraging a sequence of previously estimated CSI, and dynamically updates its predictions as new CSI becomes available. To enhance prediction accuracy, the model incorporates channel aging effects during training, ensuring that inference is based solely on data that is practically available. In contrast to existing approaches, this work adopts a realistic NR NTN channel model that uses a frequency-selective fading tapped-delay-line (TDL) structure to better represent the satellite-user propagation environment. The work by [13] investigates orbital-dependent CSI prediction for LEO satellite-to-IoT links. To bypass the need for precise positional information on restricted IoT devices, the authors propose a single-hidden-layer neural network designed for offline temporal correlation learning. This lightweight method exploits the periodic motion of satellites by using historical real, imaginary, and sign-pair CSI data, ensuring accurate predictions with very low computational cost. In these frameworks, the dependency on offline training and the absence of adaptive online learning mechanisms limit their applicability in scenarios with rapidly evolving channel conditions.

The relationship between uplink and downlink channels is analysed in [14] and [15], where the authors explore mapping techniques to predict downlink CSI from uplink measurements. Uplink-to-downlink channel mapping is a critical technique applicable to both TDD and FDD systems, particularly in NTNs. In contrast to terrestrial TDD systems, where channel reciprocity is maintained by low propagation delays and minor Doppler effects, LEO satellite networks cannot rely on such assumptions. The extreme relative velocity of these satellites triggers substantial Doppler shifts and extended propagation latencies. These factors lead to rapid channel aging, which ultimately makes downlink CSI estimation based on reciprocity inaccurate for LEO environments. Consequently, robust uplink-to-downlink channel mapping becomes indispensable for maintaining accurate, timely CSI prediction in LEO systems, regardless of duplexing mode.

In [7], a deep learning-based framework was proposed for downlink CSI prediction from uplink measurements in LEO satellite networks, addressing both TDD and frequency division duplexing (FDD) scenarios. The framework employs a hybrid approach, utilising convolutional neural networks (CNNs) to identify spatial patterns and LSTMs to model the time-varying nature of the channel state information. Model training was performed offline using ideal satellite channel models, with both training and testing relying on synthetic CSI generated under predefined channel conditions. However, in dynamic or emergency scenarios where real-time adaptation is critical, and channel conditions significantly deviate from those seen during training, the applicability of an offline-trained model is limited, as it may fail to generalise effectively to unseen propagation environments. In the context of 5G NR systems, an online channel estimation (CE) model designed for terrestrial uplink scenarios in urban environments was proposed in [16], utilising type 1 demodulation reference signals (DMRSs) as defined in 3GPP [17]. A hybrid deep learning framework for 5G channel prediction is presented in [18], where an offline-trained model is refined via online fine-tuning to achieve robust performance and high training efficiency across diverse field conditions.

To address the growing demand for scalable and resource-aware distributed intelligence, split learning has emerged as an effective paradigm for partitioning model training across heterogeneous computing entities [19], [20]. While these works establish the theoretical foundations for SL, they primarily provide high-level architectural overviews and do not address specific signal processing tasks in non-terrestrial environments. In the context of terrestrial wireless networks, [21] introduced an SL architecture that divides deep neural network training across system nodes to reduce local computational and memory burdens. However, the resource-allocation strategies in [21] are tailored for static or low-mobility terrestrial nodes and do not account for the extreme Doppler shifts and rapid topology changes inherent in satellite links. Broadening this approach to non-terrestrial domains, [22] introduced a

privacy-centric split learning framework tailored for satellite environments that accounts for typical limitations in onboard processing and bandwidth. Nevertheless, a notable gap persists in the literature: the specific utility of split learning for real-time prediction of the LEO satellite downlink channel remains unexplored. Existing frameworks, such as [22], focus on general training efficiency and privacy rather than the specific computational bottleneck of high-frequency channel-state information acquisition. Our work addresses this gap by employing the concept of split learning specifically for LEO downlink channel prediction. By partitioning the prediction model, we offload computationally intensive processing to ground stations, thereby significantly reducing the onboard computational complexity of the LEO satellite.

B. Motivations and Contributions

Emergency communication relies on continuous, low-latency, and reliable connectivity, a requirement that is increasingly met by LEO satellite systems when terrestrial infrastructure is unavailable. However, LEO satellite channels are highly dynamic due to rapid satellite motion, large Doppler shifts, and time-varying atmospheric effects. During emergency situations, these variations are further exacerbated by unpredictable environmental conditions, making the acquisition of accurate and timely channel state information (CSI) particularly challenging. Most existing channel-prediction approaches for satellite communications rely on offline training with predefined, static channel models. While effective under nominal conditions, such models are unable to adapt to abrupt channel variations caused by real-time environmental changes. As a result, prediction accuracy degrades rapidly, leading to inefficient beamforming and unreliable links, outcomes that are unacceptable in mission-critical emergency scenarios.

To address these limitations, this work adopts an offline–online channel prediction framework that leverages uplink measurements to predict the downlink channel in a TDD LEO satellite system. Offline pre-training captures general channel characteristics, while online fine-tuning enables fast adaptation to real-time channel variations without incurring prohibitive training delays. This design directly addresses the need for rapid, reliable communication in emergency environments, where channel conditions can change unpredictably and instantly. In addition, to cope with the limited onboard computational resources of LEO satellites, we employ a split learning strategy that partitions model training between the UE and the satellite. By offloading part of the computational workload to the UEs, the satellite is relieved from intensive processing tasks while still benefiting from adaptive channel prediction. This cooperative learning mechanism enables scalable deployment and sustained performance under resource constraints. In summary, this study is motivated by the need for a practical, adaptive, and resource-efficient channel prediction solution for LEO satellite emergency communications. By combining offline–online learning with split learning, the proposed framework enables reliable, efficient, and responsive learning during emergency situations, providing a significant step forward in improving satellite communication resilience in critical times. The main contributions of this study are summarised below.

- We develop a TDD OFDM LEO satellite communication framework compliant with 5G NR NTN specifications for emergency scenarios, where rapid satellite motion and long propagation delays lead to severe CSI aging. To mitigate this, we formulate an uplink-to-downlink channel prediction problem at the satellite, enabling downlink CSI acquisition. Furthermore, we propose a hybrid training strategy that combines offline pre-training and online fine-tuning, significantly reducing training latency and enabling fast adaptation to time-varying emergency environments.
- We propose a deep learning-based channel prediction model that explicitly captures both spatial and temporal channel characteristics. The model employs time-distributed-convolutional layers to extract per-uplink spatial features while preserving temporal correlations across OFDM symbols. In addition, we introduce an optimised data reshaping pipeline and regularisation mechanisms, including dropout and weight constraints, to stabilise training and improve generalisation under rapidly changing LEO channel conditions.
- We design and implement a split learning framework tailored for LEO satellite systems, in which the neural network is partitioned between the UE and the satellite. This architecture substantially reduces LEO satellite onboard computational complexity and memory requirements while preserving prediction accuracy, an essential requirement for resource-constrained satellite platforms.
- We propose a cluster-based training method that groups UEs with similar propagation environments based on residual Doppler-shift analysis. A shared channel prediction model is trained per cluster using split learning, allowing the satellite to perform downlink channel prediction and beamforming for all UEs within the same cluster. This approach reduces training redundancy, improves scalability as user density increases, and supports multiple emergency users simultaneously.

The rest of the paper is organised as follows. Section II describes the system model, and Section III presents the problem formulation. Section IV introduces the proposed split learning-based channel prediction model. Section V outlines the simulation setup, followed by results and discussion in Section VI. Finally, Section VII concludes the paper.

II. System Model

We consider a massive MIMO OFDM system in a TDD scenario where the LEO satellite is equipped with a uniform planar array (UPA) consisting of N_a antennas and N_{RF} RF chains where $N_a \gg 1$ and $N_{RF} \leq N_a$, serving multiple users equipped with single-antenna UE as depicted in Fig. 1. UEs are uniformly distributed in the geographical area, and a UE's position is assumed to remain fixed relative to the LEO satellite, ensuring a continuous line-of-sight (LoS) link throughout the communication session. Additionally, the UE is modelled as a LEO satellite terminal with sufficient computational capability to support machine learning operations

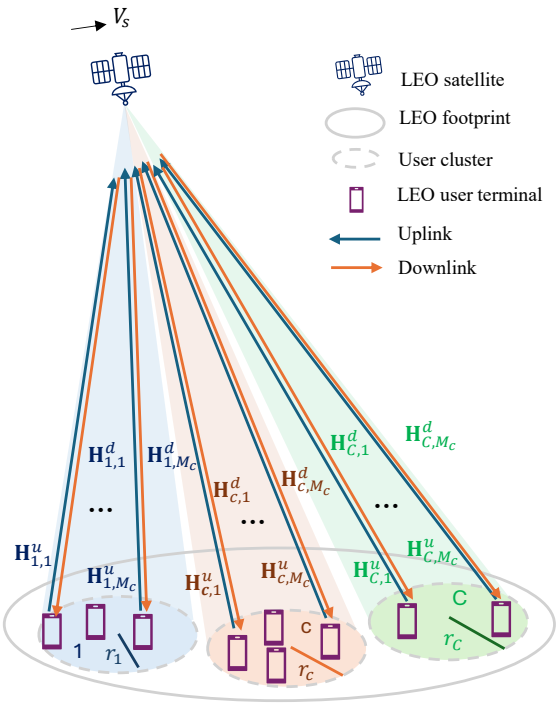


Fig. 1: System model.

[22]. By analysing GPS data, including user location, speed, and historical trajectories, users are assigned to clusters using a distance-based spatial clustering approach [23], [24]. The users are distributed across C distinct clusters, where users within the same cluster experience similar propagation and Doppler characteristics [25]. Each cluster covers a circular area on the Earth's surface with a fixed cluster radius r_c, which is selected based on Doppler spread analysis to ensure homogeneous channel conditions within the cluster. The set of clusters is indexed as $\mathcal{C} = 1, 2, \ldots, C$. As depicted in Fig. 1, each cluster $c \in \mathcal{C}$ contains M_c users. A UE in cluster c is denoted by $u_{c,m}$, where $m \in 1, 2, \ldots, M_c$. In this work, user positions are assumed to be static during the satellite service interval, therefore, cluster membership remains unchanged over the serving period.

The uplink and downlink channels are denoted by $\{.\}^U$ and $\{.\}^D$, respectively. The uplink channel from UE $u_{c,m}$ at cluster c to LEO satellite is denoted by $\mathbf{H}^U_{c,m} \in \mathbb{C}^{1\times N_a}$ where $\mathbf{H}^U_{c,m} = [\mathbf{H}^U_{c,m,1}, \mathbf{H}^U_{c,m,2}, \ldots, \mathbf{H}^U_{c,m,a}, \ldots, \mathbf{H}^U_{c,m,N_a}]$ and $\mathbf{H}^U_{c,m,a} \in \mathbb{C}^{N_S\times N_{SC}}$ denotes the uplink channel from UE $u_{c,m}$ to a-th antenna of the LEO satellite. N_S represents the number of OFDM symbols per OFDM slot and N_{SC} represents the number of subcarriers per each OFDM symbol. The downlink channel from LEO satellite to UE $u_{c,m}$ is denoted by $\mathbf{H}^D_{c,m} \in \mathbb{C}^{N_a\times 1}$ where $\mathbf{H}^D_m = [\mathbf{H}^D_{c,m,1}, \mathbf{H}^D_{c,m,2}, \ldots, \mathbf{H}^D_{c,m,a}, \ldots, \mathbf{H}^D_{c,m,N_a}]$ and $\mathbf{H}^D_{m,a} \in \mathbb{C}^{N_S\times N_{SC}}$ denotes the downlink channel from a-th antenna of the LEO satellite to UE $u_{c,m}$.

Consider user $u_{c,m}$ transmitting data $\mathbf{X}_{c,m}[n] \in \mathbb{C}^{N_S\times N_{SC}}$ in slot n where $\mathbf{X}_{c,m}[n] = [X_{c,m}[n,1], \ldots, X_{c,m}[n,s], \ldots, X_{c,m}[n,N_S]]$ and $X_{c,m}[n,s]$ is the s-th OFDM symbol of the n-th slot. The received signal at LEO satellite is denoted by $Y_{c,m}[n]$ as below:

$$Y_{c,m}[n] = \lambda^U_{c,m}\mathbf{H}^U_{c,m}[n]\mathbf{X}_{c,m}[n] + Z[n] \tag{1}$$

where $Z_{c,m} \sim \mathcal{CN}(0, \sigma_n^2)$ is the additive white Gaussian noise with zero mean and variance σ_n^2.

The objective of this study is to develop a downlink channel prediction framework for a LEO satellite system in an emergency scenario. In this architecture, we train the model using uplink channel estimates before deploying it directly onto the satellite. By performing onboard channel prediction, the system can implement effective beamforming strategies to boost operational efficiency

A. OFDM Slot Structure

In this study, we adopt a Type 1, DMRS 1+1 configuration based on an OFDM frame structure [17]. Each time slot consists of N_S symbols, indexed from 0 to $N_S - 1$, where N_p symbols are reserved for DMRS, and the remaining symbols are dedicated to data transmission [16]. Each symbol spans N_{SC} subcarriers. For a setup where $N_p = 2$, let $\mathbf{p} = [p_1, p_2]$ identify the DMRS symbol indices, while the set of data-only symbol indices is defined as $\mathbf{d} = \{0, ..., N_S - 1\} - \mathbf{p}$. Additionally, $\mathbf{s}_p$ and $\mathbf{s}_d$ represent the index vectors for the DMRS and data subcarriers within the $\mathbf{p}$ OFDM symbols, respectively.

B. Satellite Channel Model

We employ the NTN-TDL-D model, defined in 3GPP TR 38.811 [8] as the satellite channel model in our system. It consists of three taps: the first follows a Rician distribution, while the remaining exhibit non-LoS Rayleigh fading. According to the 3GPP TR 38.811 specifications [8], the Doppler shift f_D resulting from satellite motion is a function of the orbital characteristics, the elevation angle (ϵ), and the carrier frequency f_C. This relationship is mathematically defined as follows,

$$f_D = \left(\frac{R_E}{R_E + A_S} cos(\epsilon)\right)\frac{V_S f_c}{c} \tag{2}$$

where A_S is the satellite altitude and V_S is the satellite speed. c is the speed of light. R_E denotes the Earth's radius. For the n-th slot, the satellite channel $\mathbf{H}_{c,m}$ between UE $u_{c,m}$ at cluster c and LEO satellite is expressed as below for uplink and downlink in (3) and (4), respectively:

$$\mathbf{H}^U_{c,m}[n] = \sqrt{P_m\mathcal{L}_{c,m}(t)^{-1}G_m}h_{c,m}[n]e^{j2\pi f_{D_c}t} \tag{3}$$

where P_m is the transmitted power and G_m is the transmit gain at user $u_{c,m}$. f_{D_c} is the Doppler shift due to satellite movement,

$$\mathbf{H}^D_{c,m}[n] = \sqrt{P_c\mathcal{L}_{c,m}(t)^{-1}G_c(\phi_{c,m};\epsilon_{c,m})}h_{c,m}[n]e^{j2\pi f_{D_c}t} \tag{4}$$

where P_c is the transmitted power to the c-th cluster, $\mathcal{L}_{c,m}(t)$ is the path loss between user $u_{c,m}$ and the LEO satellite at time t, which varies depending on the user's position within the cluster. The path loss $\mathcal{L}_{c,m}$ for a cluster c is expressed

as a function of the slant distance $d_{c,m}$, carrier frequency f_c, and elevation angle $\epsilon_{c,m}$, represented as $\mathcal{L}_{c,m}(d_{c,m}; f_c; \epsilon_{c,m})$. The path loss consists of both the free-space path loss $\mathcal{L}^{FS}_{c,m}$ and atmospheric attenuation $\mathcal{L}^{g}_{c,m}$ defined as follows, $\mathcal{L}_c(d_{c,m}; f_c; \epsilon_{c,m}) = \mathcal{L}^{FS}_{c,m}(d_{c,m}; f_c) + \mathcal{L}^{g}_{c,m}(f_c; \epsilon_{c,m})$, with $\mathcal{L}^{FS}_{c}(d_{c,m}; f_c) = \left(\frac{4\pi d_{c,m}}{c/f_c}\right)^2$ and $\mathcal{L}^{g}_{c}(f_c; \epsilon) = \frac{A_{zen} f_c}{\sin \epsilon_{c,m}}$, where A_{zen} represents the zenith attenuation, which can be omitted if the carrier frequency is less than 10 GHz [8]. The slant distance at the beam centre is calculated as, $d_{c,m} = \sqrt{R_E^2 \sin^2(\epsilon_{c,m}) + A_S^2 + 2A_S R_E} - R_E \sin(\epsilon_{c,m})$.

C. Doppler Analysis

The relative motion of LEO satellites causes a Doppler shift, which affects users by altering the perceived signal frequency. This variation is most noticeable when user terminals are in the satellite's orbital plane, and can result in signal distortion or degradation. For analytical tractability in residual Doppler evaluation, the satellite beam is modelled as a flat surface with co-planar centre and edges and with the central beam aligned to the orbital plane, consistent with standard NTN channel modelling practices [8] and prior Doppler and beam size analyses [26]. Doppler precompensation in LEO satellites targets the beam centre, ensuring that users at the beam centre experience minimal Doppler distortion. However, users away from the beam centre encounter residual Doppler shift (RDS), which reflects the discrepancy between the compensated and observed Doppler shifts. The magnitude of RDS increases with distance from the beam centre [26], [27].

With the above system configuration, since we have clusters with cluster radius r_c, which is significantly lower than the LEO beam radius R_b, where $r_c \ll R_b$. We analyse the Doppler shift and maximum RDS observed by the cluster edge user for a range of r_c values as depicted in Fig. 2a and Fig. 2b, respectively. If the cluster radius is 10 km, the maximum RDS is less than 5 kHz. Since users within each cluster experience the same propagation characteristics and the maximum RDS effect is small, we use a single user per cluster for training the channel-prediction model. In this context, we denote the selected user in cluster c as u_c and explain the training data collection and model training steps accordingly.

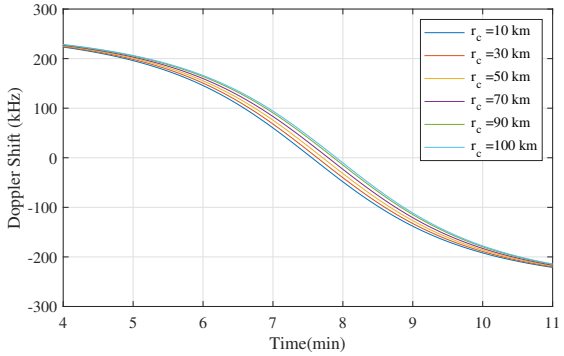
D. Channel Estimation

We utilise minimum mean square error (MMSE)-based CE at both the uplink and downlink, specifically at the LEO satellite and the UE, respectively, due to its ability to closely approximate the ideal channel response [28].

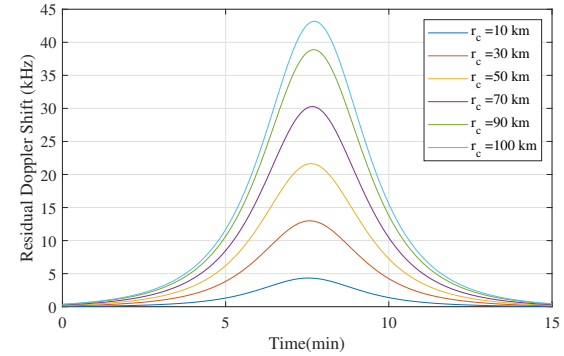
1) Uplink CE at LEO Satellite: The received pilot signals at LEO satellite in frequency domain is denoted by $\mathbf{Y}^U_{\mathbf{p}}$, where $\mathbf{Y}^U_{\mathbf{p}} = [\mathbf{Y}^U_{p_1}, \mathbf{Y}^U_{p_2}]$ and $\mathbf{Y}^U_{p_i}$ is expressed as,

$$\mathbf{Y}^U_{p_i} = \mathbf{X}_{p_i} \mathbf{H}^U_{p_i} + \mathbf{Z}_{p_i} \ . \tag{5}$$

Here, the uplink channel frequency response and white Gaussian noise are represented by the column vectors $\mathbf{H}^U_{p_i}$ and $\mathbf{Z}_{p_i}$, respectively. The diagonal matrix $\mathbf{X}_{p_i}$ accounts for the transmitted symbols. The LEO satellite performs MMSE based uplink CE on received pilots at $(\mathbf{p}, \mathbf{s}_p)$ yielding $\tilde{\mathbf{H}}^U_{p_1,\mathbf{s}_p}$ and



(a) Doppler shift variation over time.



(b) Maximum residual Doppler shift variation over time.

Fig. 2: Doppler shift and maximum residual Doppler shift under different cluster radii r_c where 10 km $\leq r_c \leq$ 100 km.

further employs linear interpolation and extrapolation ($\mathcal{F}_{I,E}$) to infer data symbols at $(\mathbf{p}, \mathbf{s}_d)$ yielding $\tilde{\mathbf{H}}^U_{p_1,\mathbf{s}_d}$ as

$$\tilde{\mathbf{H}}^U_{p_i,\mathbf{s}_d} = \mathcal{F}_{I,E}(\tilde{\mathbf{H}}^U_{p_i,\mathbf{s}_p}). \tag{6}$$

Therefore, the estimated uplink channel at DMRS symbols is denoted by $\tilde{\mathbf{H}}^U_{\mathbf{p}} = [\tilde{\mathbf{H}}^U_{p_1}, \tilde{\mathbf{H}}^U_{p_2}]$.

2) Downlink CE at UE: The LEO satellite transmits pilots and data symbols to the UE. The received signal at UE can be defined as,

$$\bar{\mathbf{Y}} = \mathbf{H}^D \bar{\mathbf{X}} + \bar{\mathbf{Z}}. \tag{7}$$

UE performs downlink CE based on the received pilot signals following the similar steps mentioned in (5)-(6). The estimated downlink channel at UE is denoted as $\tilde{\mathbf{H}}^D$ and the estimated channel corresponding to the DMRS symbols $\tilde{\mathbf{H}}^D_{\mathbf{p}}$. The LEO satellite has information about the uplink channel, while the UE has information about the downlink channel.

3) Relationship between Uplink and Downlink: For a given user m, the uplink and the downlink share common physical transmission paths and similar spatial propagation characteristics. Given the shared spatial paths of uplink and downlink transmissions for user m, the two channels exhibit a strong correlation. As a result, a deterministic mapping function between the uplink and downlink channels can be established, provided that the relationship between user position and channel state remains bijective [14]. It is shown in [15] that bijectiveness holds with high probability in most practical propagation scenarios.

Remark: The mapping function $\Psi_{U\to D}$ between uplink and the downlink channel can be denoted as

$$\Psi_{U\to D}\colon \left\{\mathbf{H}_m^U\right\}\to \left\{\mathbf{H}_m^D\right\}. \tag{8}$$

Let θ_u denote the position of a user who communicates via two possible antenna configurations: $\mathcal{A}_1$ at frequency f_1, and $\mathcal{A}_2$ at f_2. To describe the relationship between these channels, we introduce the mapping function $\Psi_{A_1,f_1\to A_2,f_2}$

$$\Psi_{A_1,f_1\to A_2,f_2}\colon \{\mathbf{H}_{m,A_1}(f_1)\}\to\{\mathbf{H}_{m,A_2}(f_2)\} \tag{9}$$

where $H_{m,A_i}(f)$ denotes the channel response between user u_m at position x_m and antenna set A_i at frequency f.

We establish the channel-to-channel relationship by combining position-to-channel and channel-to-position mapping functions. This dual-stage analysis facilitates the derivation of a robust transformation between the two channel domains. Therefore, for a given static communication environment, there exists a deterministic mapping function from the position x_m to the channel $\mathbf{H}_m$ at every antenna element a [14]. We define the position-to-channel mapping functions $g_{A_1,f_1}(.)$ and $g_{A_2,f_2}(.)$ as $g_{A_1,f_1}\colon\{x_u\}\to\mathbf{H}_{m,A_1}(f_1)$ and $g_{A_2,f_2}\colon\{x_u\}\to\mathbf{H}_{m,A_2}(f_2)$. Next, we analyse the existence of a channel-to-position mapping.

Assumption 1: Position-to-channel mapping function $g_{A_1,f_1}\colon\{x_u\}\to\mathbf{H}_{m,A_1}(f_1)$ is bijective.

The bijectiveness condition is satisfied with high probability under typical real-world propagation scenarios [15], thereby supporting its applicability in practical system models. We define the channel-to-position mapping as $g^{-1}_{A_1,f_1}\colon\mathbf{H}_{m,A_1}(f_1)\to\{x_u\}$. By leveraging the bidirectional relationship between spatial coordinates and channel responses, the existence of the mapping function $\Psi_{A_1,f_1\to A_2,f_2}$ is proposed as follows. For a propagation environment satisfying Assumption 1, a mapping function $\Psi_{A_1,f_1\to A_2,f_2}$ exists such that channel states can be transformed across varying antenna sets and carrier frequencies, denoted as,

$$\Psi_{A_1,f_1\to A_2,f_2}=g_{A_2,f_2}\circ g^{-1}_{A_1,f_1}\colon\{\mathbf{H}_{m,A_1}(f_1)\}\to\{\mathbf{H}_{m,A_2}(f_2)\}.$$

For TDD scenario, $f_1=f_2=f_c$ and channel mapping can be denoted as

$$\Psi_{A_1,f_c\to A_2,f_c}\colon\{\mathbf{H}_{m,A_1}(f_c)\}\to\{\mathbf{H}_{m,A_2}(f_c)\}. \tag{10}$$

Let A_1 and A_2 denote the antenna arrays at the LEO satellite and the UE, respectively, forming the uplink and downlink channels. Under this configuration, (10) can be further simplified to (8).

III. Problem Formulation

For a given user $u_{c,m}$, the LEO satellite needs to predict the downlink channel $\mathbf{H}^D_{c,m}$ characteristics based on the estimated uplink channel $\tilde{\mathbf{H}}^U_{c,m}$ to perform beamforming. We consider a LEO satellite operating at an altitude of $A_S=600$ km and elevation angle $\epsilon=90^0$, resulting in an approximate one-way propagation delay of 2 ms. In a TDD scenario, to avoid collisions between uplink and downlink, we define the slot structure based on the propagation delay. We define the frame structure on the LEO satellite, where each frame consists of N_U consecutive uplinks followed by a downlink, and a special slot consisting of 4 idle slots separates frames. An example of uplink-downlink slot structure for LEO TDD satellite system is depicted in Fig. 3. We assume that the proposed transmission pattern repeats over time, ensuring the periodic availability of downlink channel estimates along with the corresponding historical uplink observations.

The objective is to predict the downlink channel using historical uplink channels as highlighted in Fig. 3. We consider D, S, and U denote downlink, special and uplink, respectively. For notational simplicity, we henceforth denote the N_U consecutive uplinks and downlink channels of a given user $u_{c,m}$, as $\mathbf{H}^U$ and $\mathbf{H}^D$, respectively, omitting the user-specific subscript $(.)_{c,m}$. The channel-to-channel mapping from historical uplink channels to predicted downlink channels is denoted by $\Psi_{U\to D}$,

$$\Psi_{U\to D}\colon\left\{\mathbf{H}_i^U\right\}\to\left\{\mathbf{H}_i^D\right\} \tag{11}$$

where $\mathbf{H}_i^D$ denotes the predicted downlink channel using historical uplink channels $\mathbf{H}_i^U$ for frame i. The uplink–downlink mapping function $\Psi_{U\to D}$ jointly captures temporal channel evolution, propagation geometry, and Doppler dynamics. This coupling results in a highly nonlinear transformation that depends on latent and time-varying channel parameters, rendering closed-form analytical characterisation intractable in practical systems [14]. To overcome this limitation, $\Psi_{U\to D}$ is approximated using a deep neural network, which provides a flexible, data-driven representation capable of learning the implicit uplink–downlink relationship directly from observed channel realisations without requiring explicit parametric modelling [14], [15], [29].

To train a deep learning model solely on a LEO satellite, both uplink and downlink channel data are required as input and labels, respectively. However, the LEO satellite has access only to uplink channel data, while downlink data resides at the UE. Additionally, the LEO satellite has limited computational resources. To address these challenges, we propose a split learning-based approach for downlink channel prediction. The model is partitioned into two submodels: one deployed on the LEO satellite and the other on the UE. This partitioning enables collaborative training while offloading part of the computation to the UE, thereby reducing the LEO satellite's computational burden.

Let the overall split learning model be denoted by a function f as

$$f\big(\tilde{\mathbf{H}}_i^U\big)=f_{UE}\Big(f_{LEO}\left(\tilde{\mathbf{H}}_i^U;\boldsymbol{\Theta}_{LEO}\right);\boldsymbol{\Theta}_{UE}\Big)=\hat{\mathbf{H}}_i^D \tag{12}$$

where $\tilde{\mathbf{H}}_i^U$ is the estimated uplink channel data available at LEO satellite for frame i, which is used as the model input. $f_{LEO}(\tilde{\mathbf{H}}_i^U;\boldsymbol{\Theta}_{LEO})$ is the submodel deployed on the LEO satellite with parameters $\boldsymbol{\Theta}_{LEO}$. $f_{UE}(\cdot;\boldsymbol{\Theta}_{UE})$ is the submodel deployed on the UE with parameters $\boldsymbol{\Theta}_{UE}$. The model output is $\hat{\mathbf{H}}_i^D$. The training objective is to minimise the loss function $\mathcal{L}$ between the predicted $\hat{\mathbf{H}}_i^D$ and actual downlink channel $\tilde{\mathbf{H}}_i^D$

$$\min_{\boldsymbol{\Theta}_{LEO},\boldsymbol{\Theta}_{UE}}\mathcal{L}\Big(\hat{\mathbf{H}}_i^D,\tilde{\mathbf{H}}_i^D\Big)=\min_{\boldsymbol{\Theta}_{LEO},\boldsymbol{\Theta}_{UE}}\mathcal{L}\Big(f(\tilde{\mathbf{H}}_i^U),\tilde{\mathbf{H}}_i^D\Big)$$
$$=\min_{\boldsymbol{\Theta}_{LEO},\boldsymbol{\Theta}_{UE}}\mathcal{L}\Big(f_{UE}\big(f_{LEO}\left(\tilde{\mathbf{H}}_i^U;\boldsymbol{\Theta}_{LEO}\right);\boldsymbol{\Theta}_{UE}\big),\tilde{\mathbf{H}}_i^D\Big).$$

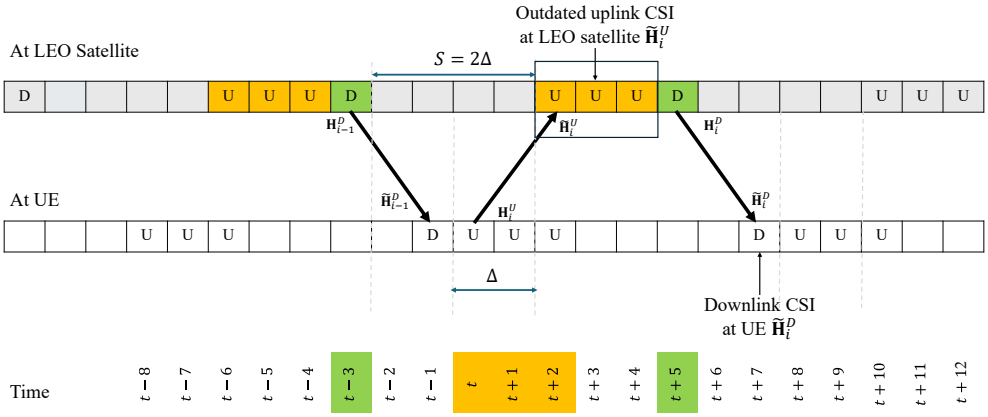


Fig. 3: An example of uplink downlink slot structure for LEO TDD satellite system considering channel aging due to propagation delay for LEO satellite with $A_S = 600$ km and $\epsilon = 90^0$.

IV. Proposed Split Learning-Based Method to Predict Downlink from Uplink

A. Background of Split Learning

Split learning is an advanced machine learning technique that distributes computational and data privacy burdens across multiple entities. In split learning, a single model is partitioned into distinct segments, typically divided between a client and a server. This partitioning allows different entities to collaborate on training the model without requiring complete data or model sharing.

In the split learning workflow, the client processes its local input data through the initial layers of the model up to a predefined cut layer. This results in intermediate feature representations, rather than the raw data, which are then transmitted to the server. The server continues the forward propagation from the cut layer using its portion of the model, calculating the final output and the associated loss during training.

Subsequently, the server initiates backpropagation, computing gradients with respect to its local model parameters. These gradients are sent back to the client, where client-side backpropagation occurs, enabling local weight updates for the client-side model layers. This iterative process continues, allowing both parties to collaborate in model training without either exposing raw data or requiring full model disclosure.

B. Split Learning in LEO Satellite System

In this context, we propose a split learning-based offline-online training framework for channel prediction. The LEO satellite and the ground user UE jointly train a shared machine-learning model. The end-to-end architecture of the proposed training framework is depicted in Fig. 4. The model is partitioned into two submodels: one deployed on the LEO satellite (SM_{LEO}) and the other on the UE (SM_{UE}). We assume that the UE has an end device with sufficient computing power to handle the model training, forward propagation, and backpropagation processes for the UE-side model. Smashed data, such as intermediate activations and backpropagation gradients, are exchanged between the satellite and the UE.

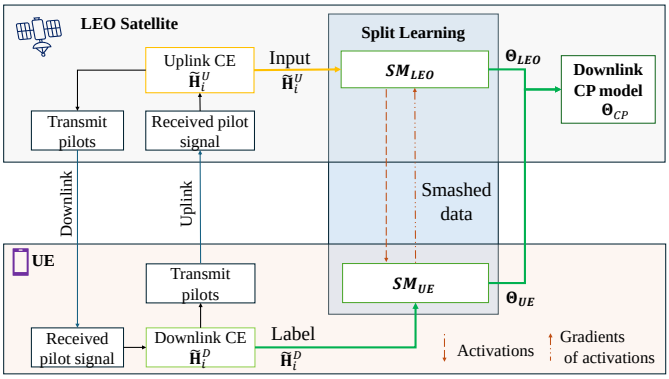


Fig. 4: End-to-end architecture of the proposed training framework, emphasising the input datasets and associated labels annotations for each machine learning model.

In TDD systems, guard periods are introduced to prevent interference between uplink and downlink transmissions caused by timing misalignment. This challenge becomes more pronounced in satellite communications, where considerable propagation delays can disrupt synchronisation. To mitigate this, a timing advance (TA) is employed to compensate for the signal travel time from the UE to the satellite, ensuring accurate uplink alignment as depicted in Fig. (3). Consequently, the length of guard periods must be carefully dimensioned to account for the associated propagation delay and maintain interference-free operation.

The LEO satellite conducts uplink CE on DMRS symbols utilising the MMSE method, based on the received DMRS symbols as explained in section II-D. The estimated number of consecutive uplink symbols, denoted N_U, is then used by the LEO satellite as input to the split learning model SM_{LEO} during training. Subsequently, the LEO satellite transmits downlink. The UE then performs downlink CE on DMRS symbols using the MMSE method, relying on the received DMRS symbols as explained in section II-D. The estimated downlink is then used as the label for the split learning model SM_{UE}.

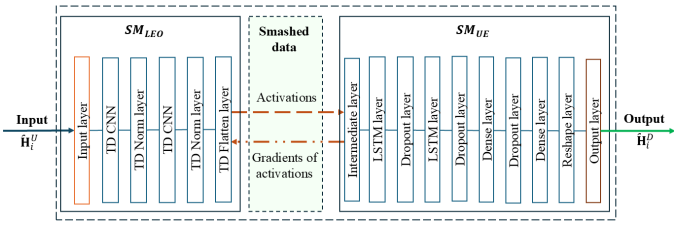


Fig. 5: Proposed training model structure with two sub-models at LEO satellite SM_{LEO} and at UE SM_{UE}.

The comprehensive architecture of the proposed split-learning-based model is depicted in Fig. 5. The satellite-side model, SM_{LEO}, in this split learning architecture is designed primarily for efficient spatial feature extraction from high-dimensional wireless signal data. The input to this model has four dimensions (N_U, N_p, N_{SC}, N_{CH}), where N_U is the number of uplinks, N_p is the number of DMRS symbols, N_U is the number of subcarriers per OFDM symbol and $N_{CH} = 2$ accounts for the real and imaginary parts of the complex-valued input, as the model operates exclusively on real-valued data representations. A time-distributed (TD) wrapper is employed to apply all subsequent operations independently across the N_U axis, enabling parallel feature extraction per uplink slot. The architecture comprises two successive 2D convolutional layers with filter sizes of $N_K \times N_K$ and output channel depths of N_{F1} and N_{F2}, respectively. Both convolutional layers use rectified linear units (ReLUs) to preserve spatial dimensions and introduce nonlinearity. Each convolutional block is followed by layer normalisation to stabilise the training and accelerate the convergence. The output of the second convolutional layer is flattened via TD Flatten, resulting in a fixed-length feature vector per uplink slot. This makes the data compatible with the input format required by an LSTM layer for sequential modelling.

In sequential tasks such as channel prediction, maintaining the temporal structure of the input is crucial. While standard CNNs are effective at extracting spatial features, they treat the input as a single block, effectively collapsing the time dimension and losing valuable sequential information. The TD-CNN module is employed to extract spatial features from the CSI while preserving the temporal structure of the uplink sequence. By applying shared convolutional kernels independently at each time step, the TD-CNN enables consistent spatial feature extraction without mixing temporal information across frames. This design enables the model to capture per-uplink spatial channel characteristics and their temporal evolution, making TD-CNNs particularly suitable for time-varying CSI scenarios.

The proposed submodel at UE, SM_{UE}, incorporates a stacked LSTM architecture for temporal sequence modelling of the extracted spatial features. The LSTM layers maintain internal memory states that capture long-term temporal dependencies across multiple uplink intervals, which is essential for accurate channel prediction in dynamic wireless environments where channel state information exhibits extended temporal correlations. The first LSTM layer contains N_L hidden units with return sequence enabled, ensuring that a full sequence of outputs is passed to the subsequent recurrent layer. This allows the model to capture long-range dependencies across the time dimension N_U. A dropout layer is applied afterwards to mitigate overfitting by randomly deactivating a fraction of the LSTM outputs during training. The second LSTM layer, also with N_L units, processes the full sequence and returns only the final hidden state, effectively summarising the temporal information into a fixed-length vector. Another dropout layer follows for regularisation. The resulting vector is then passed through a dense layer with N_D units, followed by a leakyReLU activation function to introduce nonlinearity and improve gradient flow. After a final dropout operation, the model applies a fully connected dense layer to project the output to the desired size $N = N_p N_{SC} N_{CH}$, which is reshaped using a reshape layer of (N_p, N_{SC}, N_{CH}) to match the original spatial format of the data. This structure allows the model to learn temporal dynamics across input sequences and reconstruct structured output data in the same spatial format. Finally, the output layer computes the loss by comparing model predictions with the reference labels, using normalised mean squared error (NMSE) as the loss function.

The split between the satellite-side and UE-side models is intentionally designed to accommodate the asymmetric computational and communication constraints of LEO satellite systems. The satellite processes the uplink signal as the model input, while the UE processes the downlink signal, which serves as the ground-truth label during training. This division naturally aligns with the physical signal flow and enables a distributed learning framework without requiring full end-to-end processing on the satellite.

Onboard satellite processing is limited to lightweight feature extraction, comprising convolution and normalisation layers. These operations are selected to minimise computational load, memory footprint, and energy consumption, all of which are critical constraints in space-based platforms. By restricting the satellite-side model to low-complexity operations, the architecture avoids deploying recurrent or fully connected layers on the satellite, which would otherwise incur prohibitive processing and latency costs. The UE, equipped with comparatively abundant computational resources, executes the computationally intensive components of the model, including temporal sequence modelling and dense transformations. This partitioning enables efficient utilisation of heterogeneous resources, supports scalability across multiple satellites and UEs, and adheres to the principles of split and distributed learning.

The cut layer is placed after the Time-Distributed (TD) Flatten operation to balance communication overhead against onboard computation. At this point, the intermediate activations preserve the temporal structure of the uplink signal while compressing spatial dimensions into compact feature vectors. Earlier splits would require transmitting high-dimensional feature maps, substantially increasing bandwidth consumption, whereas later splits would relocate sequential modelling, e.g., LSTM recurrence, to the satellite, significantly increasing computational complexity. The communication overhead is determined by the size of the smashed data transmitted from the satellite to the UE. The intermediate activation tensor has dimension $N_B \times N_U \times (N_p \times N_{SC} \times N_{F_2})$, resulting in a

per-batch communication cost that scales linearly with these parameters. This overhead is explicitly quantified t o assess the trade-off between model accuracy and bandwidth usage. By placing the cut after TD Flatten, the proposed architecture minimises transmitted data volume while preserving task-relevant temporal features, making it suitable for bandwidth-constrained satellite-to-ground links.

In this study, we assume perfect synchronisation and reliable exchange of smashed data, consistent with standard practice in split and distributed learning frameworks. The exchanged activations and gradients are typically of a much lower rate than raw CSI or data samples and are less latency-critical. Moreover, existing NTN systems employ timing synchronisation, error control coding, and retransmission mechanisms, which can be leveraged to support the reliable exchange of intermediate representations.

C. Training Data Generation

To predict the subsequent downlink channel, we utilise a sequence of N_U uplink channels, as illustrated in Fig. 3. Accordingly, the input training data at the LEO satellite, denoted by $\mathbf{H}_i^{IN}$, comprises the estimated uplink channel responses at DMRS symbol positions. Specifically, $\mathbf{H}_i^{IN} = \left\{\Re\left\{\mathbf{H}_i^{IN,C}\right\}, \Im\left\{\mathbf{H}_i^{IN,C}\right\}\right\}$, where $\mathbf{H}_i^{IN,C} = \{\tilde{\mathbf{H}}_{\mathbf{p}}^{U,1}, \tilde{\mathbf{H}}_{\mathbf{p}}^{U,2}, \ldots, \tilde{\mathbf{H}}_{\mathbf{p}}^{U,N_U}\}$, and $\tilde{\mathbf{H}}_{\mathbf{p}}^{U,k}$ denotes the estimated complex-valued uplink channel at the DMRS positions $\mathbf{p}$ for the k-th uplink. The operators $\Re\{\cdot\}$ and $\Im\{\cdot\}$ extract the real and imaginary parts, respectively, of a complex-valued input. The corresponding training label at the UE side is the estimated downlink channel at the same DMRS positions, denoted by $\mathbf{H}_i^{OUT}$, $\mathbf{H}_i^{OUT} = \left\{\Re\left\{\mathbf{H}_i^{OUT,C}\right\}, \Im\left\{\mathbf{H}_i^{OUT,C}\right\}\right\}$, where $\mathbf{H}_i^{OUT,C} = \tilde{\mathbf{H}}_{\mathbf{p}}^{D}$ is the complex-valued estimated downlink channel at DMRS positions.

D. Model Training

To improve the model training efficiency in online training by reducing the training delay, we employ an offline-online training scheme. The same model is trained offline on the ground with ideal data. We employ the NMSE loss and Adam optimiser. Trained model parameters $\hat{\mathbf{\Theta}}_{LEO}$ and $\hat{\mathbf{\Theta}}_{UE}$ are then shared with the LEO satellite and UE, respectively.

The entire processing pipeline for split-learning-based model training is detailed as follows. We consider the training batch size to be N_B. At the LEO satellite, we collect N_B training samples and use them for training. For any batch b, $\tilde{\mathbf{H}}_b^{IN}$ denotes the training batch input to the SM_{LEO} at LEO satellite and $\tilde{\mathbf{H}}_b^{IN} = \left\{\tilde{\mathbf{H}}_i^{IN}, \tilde{\mathbf{H}}_{i+1}^{IN}, \ldots, \tilde{\mathbf{H}}_{i+N_B-1}^{IN}\right\}$. During the forward pass, LEO satellite computes intermediate data $\mathbb{I}_{LEO}$ as, $\mathbb{I}_{LEO} = f_{LEO}(\tilde{\mathbf{H}}_b^{IN}; \mathbf{\Theta}_{LEO})$ where $\mathbf{\Theta}_{LEO}$ represents the parameters of submodel SM_{LEO}. Then the LEO satellite shares $\mathbb{I}_{LEO}$ with the UE. Subsequently, UE computes the output $\hat{\mathbf{H}}_b^{OUT}$ as, $\hat{\mathbf{H}}_b^{out} = f_{UE}(\mathbb{I}_{LEO}; \mathbf{\Theta}_{LEO})$, where $\mathbf{\Theta}_{UE}$ represents the parameters of submodel SM_{UE}. UE then performs the NMSE loss calculation using the predicted output $\hat{\mathbf{H}}_b^{OUT}$ and the label $\tilde{\mathbf{H}}_b^{OUT}$ as,

$$\mathcal{L}\left(\hat{\mathbf{H}}_b^{OUT}, \tilde{\mathbf{H}}_b^{OUT}\right) = \frac{1}{N_B}\sum_{n=1}^{N_B} \frac{\left\|\tilde{\mathbf{H}}_b^{OUT} - \hat{\mathbf{H}}_b^{OUT}\right\|^2}{\left\|\tilde{\mathbf{H}}_b^{OUT}\right\|^2}. \quad (13)$$

During the backward pass, UE computes gradients $\nabla\mathbf{\Theta}_{UE}$ and $\nabla\mathbb{I}_{LEO}$. UE updates the model parameters as,

$$\mathbf{\Theta}_{UE} = \hat{\mathbf{\Theta}}_{UE} - \eta\nabla\mathbf{\Theta}_{UE}. \quad (14)$$

Then UE sends gradients $\nabla\mathbb{I}_{LEO}$ back to LEO satellite where LEO satellite calculate gradients $\nabla\mathbf{\Theta}_{LEO}$ using backpropagation as

$$\frac{\partial\mathcal{L}}{\partial\mathbf{\Theta}_{LEO}} = \frac{\partial\mathcal{L}}{\partial\mathbb{I}} \cdot \frac{\partial\mathbb{I}}{\partial\mathbf{\Theta}_{LEO}}. \quad (15)$$

Then LEO satellite updates model parameters as,

$$\mathbf{\Theta}_{LEO} = \hat{\mathbf{\Theta}}_{LEO} - \eta\nabla\mathbf{\Theta}_{LEO}. \quad (16)$$

After I_t iterations, the weight matrix $\mathbf{\Theta}_{UE}$ is shared with the LEO satellite and at LEO satellite the global weight matrix $\mathbf{\Theta}_{CP}$ is generated by combining $\mathbf{\Theta}_{LEO}$ and $\mathbf{\Theta}_{UE}$. The proposed algorithm is summarised in Algorithm 1.

Algorithm 1 Split Learning-Based Model Training

```
Initialise parameters Θ_LEO and Θ_UE
for Iteration i_t = 1, 2, ..., I_t do
    for Mini-batch b = 1, 2, ..., N_Z do
        LEO satellite: Collect training input H̃_b^IN
        UE: Collect labels H̃_b^OUT
        LEO Satellite SM_LEO Forward Pass:
            Compute intermediate data 𝕀_LEO
            Send 𝕀_LEO to UE
        UE SM_UE Forward Pass:
            Compute Ĥ_b^out
            Compute NMSE loss ℒ
        UE SM_UE Backward Pass:
            Compute gradients ∇Θ_UE and ∇𝕀_LEO
            Update Θ_UE using (14)
            Send ∇𝕀_LEO to LEO satellite
        LEO satellite SM_LEO Backward Pass:
            Compute gradients ∇Θ_UE using (15)
            Update Θ_LEO using (16)
    end for
end for
```

E. Downlink Channel Prediction

Once the online model training is completed, the trained model is fully deployed on the LEO satellite to predict the downlink channel from the estimated uplinks. The complete setup of the proposed channel prediction model is depicted in Fig.6. We consider the uplink channel sequence of N_U number of uplinks at the LEO satellite as the input to the model, and the predicted downlink channel is denoted by $\hat{\mathbf{H}}^D$. Consequently, beamforming can be performed using the predicted downlink channel. The NMSE is adopted as the primary evaluation metric because it directly quantifies the accuracy of the

predicted channel coefficients. Under zero-forcing downlink beamforming with imperfect channel state information, the impact of channel estimation error characterised by the NMSE can be captured through an effective signal-to-noise ratio Γ_{eff} [30], [31] given by, $\Gamma_{\text{eff}} \approx \frac{P_m (1-\text{NMSE})}{P_m \text{NMSE} + \sigma_n^2}$. Data rate η can be modeled as $\eta = log_2(1 + \Gamma_{\text{eff}})$. Minimising NMSE translates to improved beamforming accuracy and a higher data rate.

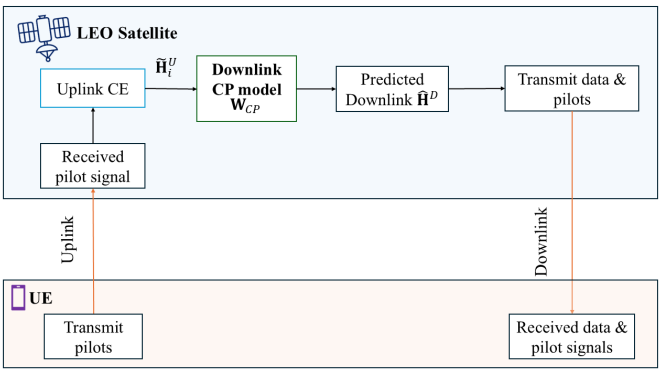


Fig. 6: Architecture of the proposed split learning-based channel prediction model.

F. Computational Complexity Analysis

We perform a computational complexity analysis of the model using Big $\mathcal{O}$ notation to characterise how the number of operations scales with input dimensions and model parameters. The computational complexity of the TD convolutional layer and the flatten layer is $\mathcal{O}\left(N_U N N_F N_K^2\right)$ and $\mathcal{O}\left(N_U N N_F\right)$, respectively. By considering the dominant term, the complexity of the LEO side model SM_{LEO} is $\mathcal{O}\left(N_U N N_F N_K^2\right)$. The computational complexity of the LSTM layers 1 and 2 is $\mathcal{O}\left(N_U N_L^2 + N N_L N_F\right)$ and $\mathcal{O}\left(N_U N_L^2\right)$, respectively. Finally, computational complexity of the last two dense layers $\mathcal{O}\left(N_L N_D\right)$ and $\mathcal{O}\left(N_D N\right)$, respectively. The complexity of the UE side model SM_{UE} is $\mathcal{O}\left(N_U N_L^2 + N(N_L N_F + N_D) + N_D N_L)\right)$. Table I presents a detailed computational complexity analysis of the proposed models, separately evaluated for the LEO satellite and the UE sides. This breakdown quantitatively demonstrates the reduction in computational burden achieved by split learning, highlighting its efficiency in distributing model training across network segments.

TABLE I: Computational Complexity Analysis

Model	Computational Complexity
SM_{LEO}	$\mathcal{O}\left(N_U N N_F N_K^2\right)$
SM_{UE}	$\mathcal{O}\left(N_U N_L^2 + N(N_L N_F + N_D) + N_D N_L)\right)$

V. Simulation Setup

A. System Setup

We consider a TDD massive MIMO-OFDM system employing a LEO satellite. The satellite is equipped with a UPA consisting of $N_a = 256$ antenna elements and $N_{RF} = 12$ RF chains. The user distribution is modelled using $C = 5$ user clusters, each with a radius of $r_c = 10$ km and supporting up to $M_c = 100$ uniformly distributed users per cluster. For channel modelling, we adopt the 3GPP NR NTN TDL-D channel model specified in [8] as the baseline propagation environment. To account for atmospheric impairments, rain attenuation effects are incorporated according to the ITU-R P.838-3 model [32]. Spatially heterogeneous rain conditions are assumed across different channel clusters to assess the robustness of the proposed model under varying precipitation scenarios. The key simulation parameters, including satellite orbital characteristics, OFDM slot structure, and channel model configurations, are summarised in Table II.

TABLE II: Table of System Parameters

Parameter	Value
Satellite altitude A_S	600 km
Centre frequency f_c	11.275 GHz
No. of DMRS symbols N_p	2
DMRS positions $\mathbf{p}$	[2,11]
DMRS type	Type I
Slot duration	1 ms
No. of OFDM symbols per slot N_S	14
Subcarrier spacing	15 kHz
No. of subcarriers per slot N_{SC}	300
Cyclic prefix	Normal
No. of consecutive uplinks N_U	5

TABLE III: Key Hyperparameters

Hyperparameter	Value
Batch size N_B	256
Learning rate	$1e-3$
Model Optimiser	Adam
Loss function	NMSE

B. Hyperparameter Setup

The simulation setup involves a model designed with several key hyperparameters. The input shape is (N_U, N_p, N_{SC}, N_{CH}). The model includes convolution layers with $N_{F1} = 16$ and $N_{F2} = 32$ filters of size $N_K = 3$ and ReLU activation, followed by TD layer normalisation and a flatten layer. Two LSTM layers, each with $N_L = 512$ units, process the sequence, and dropout of 0.25 is applied after each LSTM to regularise the model. Additionally, two dense layers are used, one with $N_D = 1024$ units and leakyReLU activation, and a second with $N_O = N_p N_{SC} N_{CH}$ units for the output. The final output is reshaped into (N_p, N_{SC}, N_{CH}) to match the expected format. The hyperparameters used in model training are summarised in Table III.

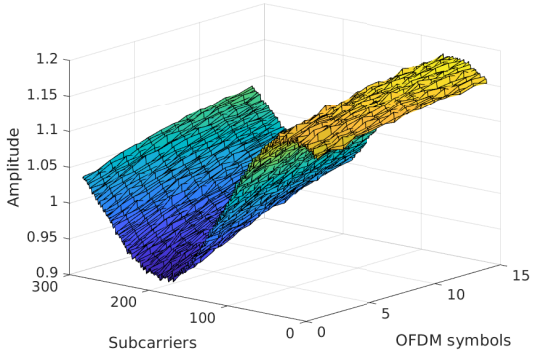
C. Data Generation

A dataset covering an SNR range from -10 dB to 20 dB was generated following the procedure described in Section IV-C. A total of 5000 consecutive OFDM slots were considered to capture temporal channel variations. For each SNR level, the dataset was randomly partitioned into 80% for training and 20% for testing. This stratified split across all SNR values ensures that the training dataset fully spans the targeted operating SNR range and avoids bias toward specific channel conditions.

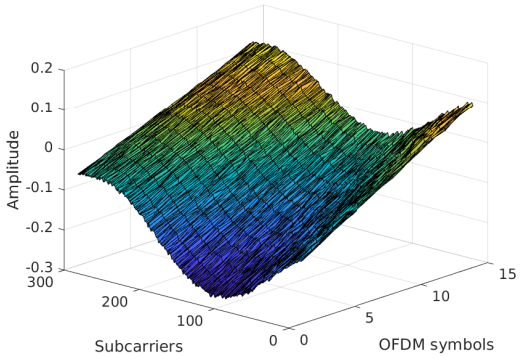
Each sample in the dataset consists of an input–label pair constructed from consecutive uplink and downlink channel observations. The input tensor has dimensions $(5, 2, 300, 2)$, where the dimensions correspond to the number of input time steps, DMRS symbols, subcarriers, and real and imaginary components, respectively. The corresponding label tensor has dimensions $(2, 300, 2)$, representing the predicted downlink CSI for one future downlink slot. This data structure enables the model to jointly exploit spatial, spectral, and temporal correlations in the channel.

VI. Results and Discussion

In the frequency domain, the actual wireless channel is represented by its complex frequency response, which comprises both real and imaginary components. Channel variation within a slot is captured to analyse the temporal and frequency selectivity of the channel. Fig. 7 illustrates the detailed behaviour of the channel coefficients across subcarriers within a single slot, highlighting the variations in both the real and imaginary parts. These components collectively characterise the complex gain of each subcarrier due to multipath fading and frequency-selective propagation. The figure serves to validate the channel's dynamic nature and is essential for understanding its impact on system performance.



(a) Real component



(b) Imaginary component

Fig. 7: Real and imaginary components of the ideal downlink channel variation considering a single slot.

A. Computational Efficiency

For training the proposed machine learning model, the NMSE loss function is employed as the optimisation criterion, enabling effective learning of the channel characteristics by minimising the NMSE. The training performance of the model is depicted in Fig. 8, which illustrates the evolution of the training loss as a function of the number of training epochs. The loss curves characterise the model's learning behaviour under three training paradigms: fully offline, fully online, and the proposed hybrid offline–online framework. A comparative analysis of the convergence behaviour across these approaches is conducted. In the fully offline training case, ideal channel data are utilised, whereas in the fully online training scenario, the model is trained using estimated channel data. As shown in Fig. 8, the fully online-trained model exhibits stable convergence, with the training loss decreasing monotonically and reaching a steady plateau after approximately 1500 epochs. This plateau suggests that the model has achieved near-optimal performance without any indication of significant overfitting. In contrast, the fully offline-trained model converges faster, reaching stability after around 1000 epochs. Notably, the proposed hybrid offline–online training strategy, which initialises the model using offline-trained weights and subsequently performs online fine-tuning under the light-rain scenario, achieves convergence within approximately 600 epochs. This corresponds to a nearly 60% reduction in convergence time compared with the fully online training approach. The accelerated convergence achieved by the hybrid framework demonstrates its efficiency and effectiveness in adapting pretrained models to new environmental conditions while requiring substantially reduced additional training effort.

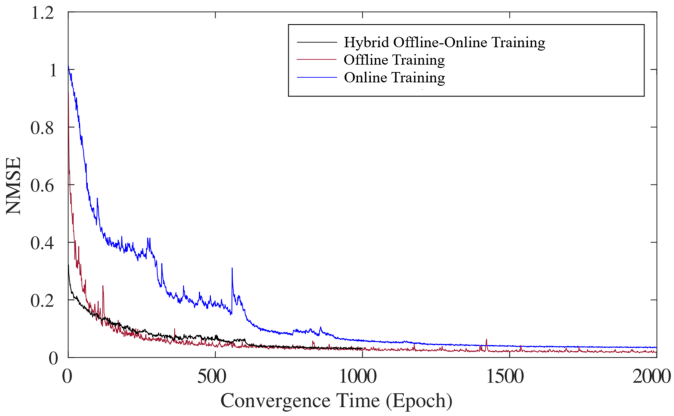


Fig. 8: NMSE loss over training iterations.

B. Processing Time

We conduct a comprehensive feasibility assessment of the proposed TD-CNN-LSTM model with split learning and hybrid online-offline training. Additionally, we consider the proposed TD-CNN-LSTM with split learning and online training, as well as the CNN-LSTM [7] with split learning and online training. In summary, we label all three models as follows:

- M_1: proposed TD-CNN-LSTM with split learning and hybrid online-offline training

- M_2: proposed TD-CNN-LSTM with split learning and online training
- M_3: CNN-LSTM [7] with split learning and online training

For benchmarking purposes and to enable a fair comparison under emergency scenarios, we also include the online-trained deep learning model introduced in M_3 [7]. Instead of the split learning, the entire model M_3 is trained end-to-end on the LEO satellite. This configuration necessitates the transmission of labelled data from the UE to the satellite, thereby introducing significant downlink communication overhead per frame.

Under the assumption of full parallel processing capability, such as with modern GPUs or specialised hardware like TPUs, the computation time of each layer in the neural network is determined not by the total number of operations, but by the depth of the computation graph, i.e., the longest chain of dependent operations. This allows for significant time savings, especially in layers composed of many independent element-wise or matrix operations. Starting with the convolutional layer wrapped in a TD wrapper, each output pixel is calculated via a dot product between the kernel and a local patch of the input. This operation requires $N_K^2 N_{CH}$ multiplications per output value, but with parallelism, all of them can be computed simultaneously. The additions involved in summing the dot product can be structured as a binary tree, with depth $log_2(N_K^2 N_{CH})$. As such, the forward pass for each pixel effectively takes time proportional to $T_m + log(N_K^2 N_{CH})T_a$, which is substantially faster than a serial implementation. Let T_m represent the time required to perform a single multiplication operation, and let T_a represent the time required for a single addition operation.

Next, the layer normalisation operation consists of several fully parallelisable steps: computing the mean and variance, normalising, and then applying a scale-and-shift transformation. Each of these steps operates element-wise or over simple reductions, so that the total time can be bounded by a small constant multiple of T_m. The flatten operation is a reshaping step and incurs no computation time. The first LSTM layer introduces a critical sequential dependency: each time step's hidden state depends on the previous one. Even with parallel matrix operations within each step, this recurrence leads to a total time that scales linearly with the number of time steps, N_U, so the effective time for the forward pass is $N_U T_m$.

The Dropout layers are element-wise and thus have trivial depth, each multiplies the input by a mask, leading to a single-step depth of T_m. The second LSTM still processes time steps sequentially but outputs only the final state; the timing remains linear in N_U. The subsequent dense layers each involve a matrix-vector multiplication, which, under full parallelism, takes constant time per layer because all neuron activations can be computed simultaneously. Similarly, the leaky ReLU and dropout layers that follow operate element-wise on a single time unit. The final dense and reshape layers are also completed in constant time under parallel assumptions, with reshape involving no computation.

In neural networks, the computational cost of the backward pass is typically about twice that of the forward pass, as it involves computing gradients for both the inputs and the weights. Therefore, the total computational cost during training is approximately three times that of the forward pass, accounting for the forward and backward passes and the gradient calculations required for optimisation. Therefore, by considering both forward and backward passes, the approximated model training time under full parallelism is $(6N_U + 18)T_m$, demonstrating how significant gains can be made when using parallel computation hardware effectively. In the proposed split learning scenario, since the LEO satellite and UE share activations and gradients of activations in each training batch, the total training can be approximated as $I_1 N_Z (N_B(6N_U+18)T_m+2T_P)$ where $I_T = I_1$ is the training iterations, N_Z is the number of batches, T_P is the propagation delay between the LEO satellite and the UE.

In contrast to the offline-online trained model M_1, the model M_2 is fully trained online, so $I_T = I_2$. The approximated total training time is $I_2 N_Z (N_B(6N_U + 18)T_m + 2T_P)$. The model M_3 is entirely trained on the LEO satellite. Consequently, the UE must continuously transmit label information, specifically the estimated downlink data, to the LEO satellite. This persistent exchange of labelling data increases communication frequency between the UE and the satellite, thereby increasing control channel overhead. The approximated total training time is $I_3 N_Z N_B((6N_U + 30)T_m + T_P)$. Finally, we define M_4 and M_5 as fully offline-trained models with TD-CNN-LSTM and CNN-LSTM, respectively, trained on ideal channel data.

Table IV summarises the processing time for model training, where I_1, I_2, I_3, I_4 and I_5 denote the convergence time in terms of training epochs for models M_1, M_2, M_3, M_4, and M_5, respectively. Offline-trained models M_4 and M_5 rely on ideal, labelled data and train exclusively on the ground, resulting in low per-batch computation times. However, their inability to adapt to dynamic channel conditions significantly limits their estimation accuracy in realistic LEO satellite environments. In contrast, fully online approaches M_2 and M_3 leverage estimated labels and enable continuous adaptation, but this flexibility comes at the cost of increased computational complexity and slower convergence, particularly when training is performed entirely onboard the satellite as in M_3. The proposed hybrid offline–online model M_1 effectively balances these competing objectives by combining offline pre-training with online refinement, while distributing computation between the satellite and the UE. For the proposed offline-online trained model M_1, $I_1 \ll I_2 < I_3$, making the convergence fast. This hybrid strategy achieves fast convergence, improved NMSE performance, and reduced onboard burden, highlighting its suitability for practical LEO satellite systems where both adaptability and computational efficiency are critical.

C. Feasibility of Implementation on LEO Satellite

To assess the feasibility of implementation on a LEO satellite platform, we analysed the computational latency of our proposed model. This analysis considers state-of-the-art, highly integrated onboard computing platforms employed in operational LEO satellites [33]. Assuming a hardware clock frequency of 1.5 GHz [34] and leveraging parallel processing capabilities, the estimated computational latency can be approximated. One clock cycle T_{clk} is about 0.67 ns. We

TABLE IV: Model Comparison

Model	Training	Training time	Training time per batch	Label data	Performance Analysis
M_1	Hybrid offline-online	$I_1 N_Z(N_B(6N_U+18)T_m + 2T_P)$	4.021 ms	Estimated	Joint offline–online training framework with computational load shared between the LEO satellite and the UE. Enables fast convergence and achieves high NMSE performance while maintaining manageable onboard processing complexity.
M_2	Online	$I_2 N_Z(N_B(6N_U+18)T_m + 2T_P)$	4.021 ms	Estimated	Fully online training strategy with computation distributed between the LEO satellite and the UE. While it adapts to dynamic channel conditions, it converges more slowly due to the lack of offline pre-training.
M_3 [7]	Online	$I_3 N_Z(N_B((6N_U+30)T_m) + T_P)$	2.028 ms	Estimated	Purely online training performed entirely on the LEO satellite. Results in high onboard computational complexity and limited learning capacity, longer training time, leading to inferior estimation performance.
M_4	Offline	$I_4 N_Z N_B(6N_U+18)T_m$	20.58 μs	Ideal	Fully offline-trained model using ideal labelled data on the ground station. Suffers from reduced performance due to the lack of online adaptation to real-time channel variations.
M_5 [7]	Offline	$I_5 N_Z N_B(6N_U+30)T_m$	28.82 μs	Ideal	Offline ground-trained model relying on ideal labeled data. Demonstrates degraded performance under practical deployment conditions due to model mismatch.

assume that the latency of the multiplication operation is four clock cycles; therefore, the time for a single multiplication $T_m = 4T_{clk} = 2.68$ ns. Under this hardware configuration, the total computational latency for one forward pass followed by one backward pass of the model during training is estimated at approximately 80.4 ns. For a full batch, this results in a cumulative latency of 20.58 μs. Furthermore, the inference latency, measured from the moment input data is received to the generation of the model output, is approximately 42.88 ns. The total training time for a single batch is denoted by T_B, and the inference time is denoted by T_I. Table V summarises the training and inference time required for the proposed model for different N_U slots. These estimates demonstrate that the model's computational demands are well within the processing capabilities of contemporary LEO satellite platforms, thereby confirming the practical feasibility of onboard training and inference.

TABLE V: Feasibility Analysis for Proposed Model M_1

No. UL slots, N_U	2	3	4	5
Training time, T_B **(μs)**	20.58	24.70	28.82	39.72
Inference time, T_I **(ns)**	26.80	32.16	37.52	42.88

The input data and corresponding labels are collected in real time at the LEO satellite and the UE, respectively. The communication latency introduced in this scheme is primarily governed by the propagation delay, which is a function of the slant distance between the satellite and the UE, typically ranging between 2 ms to 5 ms for LEO orbits. Furthermore, in this setup, the LEO satellite and the UE collaboratively train the model by exchanging smashed data rather than full raw inputs or model parameters, significantly reducing communication overhead. This architecture offers two key advantages: (i) it enables low-latency training without requiring full data transfer to the satellite, and (ii) it supports scalable model updates with minimal impact on the onboard computational resources. Combined with the demonstrated low computational latency of 80.4 ns per forward-backward pass, the proposed framework demonstrates the practical viability of deploying learning-enabled models on LEO satellites for dynamic, real-time applications such as emergency scenarios.

D. Prediction Accuracy

We evaluate the NMSE performance of the proposed model under two different TDD configuration patterns, DSUUU and DSUUD [35], over a range of E_b/N_0 values. The proposed approach is compared against two reference models: the baseline CNN–LSTM architecture presented in [7] and an LSTM-only model [29], for uplink-to-downlink (ULDL) channel prediction, as illustrated in Fig. 9. To ensure a fair comparison, the CNN–LSTM baseline adopts the same deep learning architecture, including all original layers and hyperparameters, as reported in [7]. The input tensor is reshaped from dimensions $N_U N_p \times N_{SC} \times N_{CH}$ to conform to the baseline model's input requirements. In addition to ULDL prediction, we also evaluate the NMSE performance of the proposed model for downlink-to-downlink (DLDL) prediction. The results indicate that both reference models exhibit limited effectiveness in the considered LEO satellite channel under the two TDD patterns. In contrast, the proposed model consistently outperforms the CNN–LSTM baseline and the LSTM-only model across the evaluated E_b/N_0 range. This performance improvement is attributed to TD-CNN's ability to preserve and exploit the spatial structure within each uplink snapshot, while the stacked LSTM layers effectively capture temporal correlations across successive uplinks. Furthermore, the hybrid offline–online fine-tuning strategy enables adaptation to distributional shifts encountered during operation, yielding consistently lower NMSE than both reference models.

We evaluate NMSE performance for downlink channel prediction with varying numbers of uplink slots, N_U. Specifically, we consider $N_U = 2$, $N_U = 3$, $N_U = 4$, and $N_U = 5$ as input configurations during model training, assuming the DSUUUUU TDD pattern. The corresponding NMSE results are presented in Fig. 10. Across the full range of E_b/N_0, the configuration with $N_U = 2$ systematically yields the worst NMSE performance, whereas the configuration with $N_U = 5$ consistently achieves the lowest NMSE. These observations clearly indicate the importance of increasing the number of uplink slots used for downlink channel prediction. A higher N_U provides a richer temporal and spatial context, enhancing the fidelity of the learned channel representation

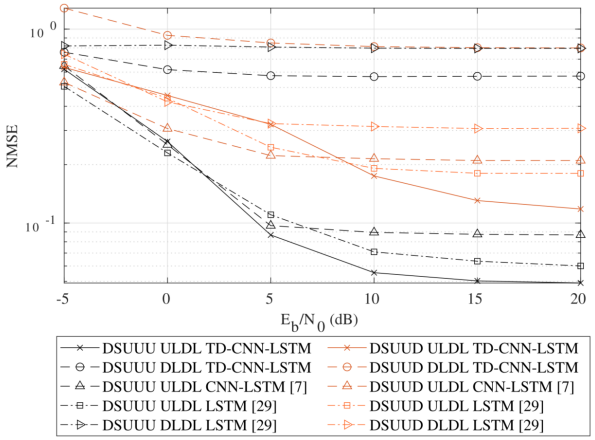


Fig. 9: Performance comparison of proposed TD-CNN-LSTM model for different TDD patterns.

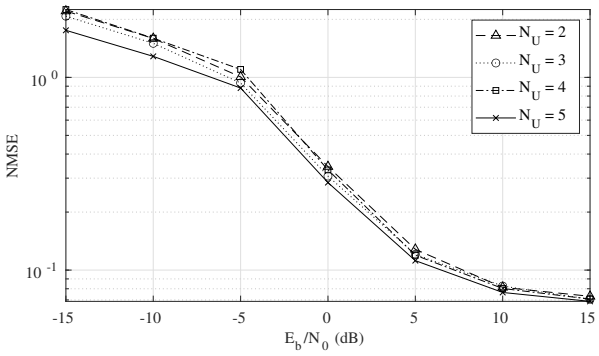
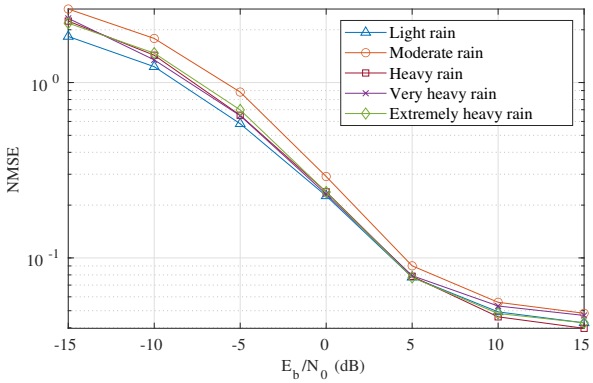
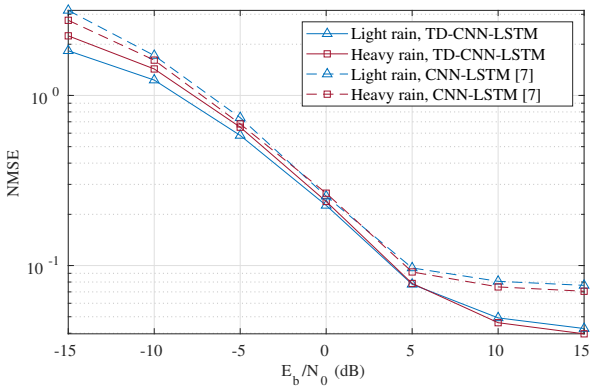


Fig. 10: Prediction performance over targeted E_b/N_0 range, for different uplink slots as training input.



(a) Analysis under different rain conditions for five clusters: light rain, moderate rain, heavy rain, very heavy rain, and extremely heavy rain.



(b) Comparison under same rain conditions: light rain, and heavy rain.

Fig. 11: Performance analysis under different rain conditions.

and significantly improving prediction accuracy.

We consider five distinct clusters, each representing a unique rain attenuation condition corresponding to different precipitation intensities: light, moderate, heavy, very heavy, and extremely heavy rainfall. Each scenario is parameterised by a specific rain intensity, measured in millimetres per hour (mm/h). Fig. 11a illustrates the NMSE performance across these clusters, each subjected to its respective rain attenuation scenario. The results demonstrate that the proposed model exhibits consistent performance across all rain conditions over the entire evaluated E_b/N_0 range. Furthermore, we compare the NMSE performance of the proposed model with [7], under light rain and heavy rain conditions as depicted in Fig. 11b. Under both scenarios, the proposed model achieves lower NMSE than the baseline, highlighting its improved robustness. These results enable a comprehensive assessment of emergency communication scenarios, offering critical insights into the system's resilience and performance under diverse, realistic environmental conditions.

Furthermore, the NMSE performance is analysed with respect to the cluster radius and the associated maximum RDS values, as shown in Fig. 12. The cluster radius is defined such that the RDS experienced by all users within a cluster is upper-bounded by a predefined maximum RDS derived from the RDS analysis. Two cluster radii are considered, namely $r_c = 4$ km and $r_c = 8$ km. For the case of $r_c = 4$ km, the maximum RDS is approximately 1.5 kHz, while increasing the cluster radius to $r_c = 8$ km results in a higher maximum RDS of approximately 3.5 kHz. It is observed that the proposed TD-CNN-LSTM model consistently outperforms the baseline model in [7] for both cluster configurations. In addition, superior NMSE performance is achieved with the smaller cluster radius $r_c = 4$ km, which can be attributed to the lower RDS compared to $r_c = 8$ km.

We evaluate the achievable data rate as a function of E_b/N_0 using the analytical expressions derived in Section IV-E, as illustrated in Fig. 13. The performance of the proposed channel prediction framework is assessed under two TDD configurations, namely DSUUU and DSUUD, and compared against the method in [7]. The proposed approach consistently achieves higher data rates than [7] for both TDD patterns.

The LSTM-only model proposed in [29] use consecutive historical downlinks as input to predict future downlink channel states. Under the ULDL setting, the model leverages multiple consecutive uplink slots to forecast the subsequent downlink channel, whereas in the DLDL setting, it utilises multiple historical discrete downlink slots to predict the upcoming downlink channel state. In contrast, the proposed approach considers temporally discrete historical downlink observations, necessitating the introduction of an additional

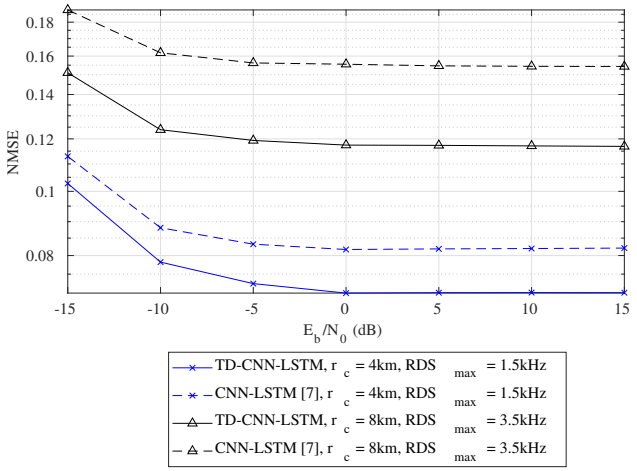


Fig. 12: NMSE performance analysis over different cluster radii and maximum RDS values.

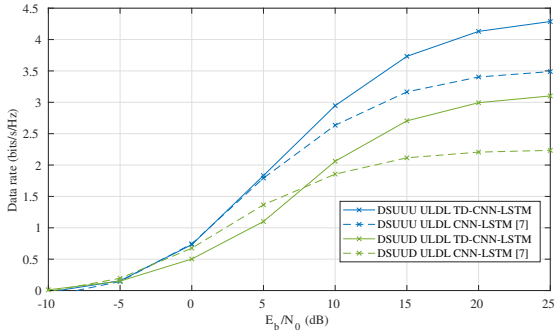


Fig. 13: Data rate comparison of proposed TD-CNN-LSTM model for different TDD patterns.

dimension to encode the temporal spacing between downlink instances, thereby improving temporal representation [36] for the LSTM-only model. The simulation results presented in Fig. 14 indicate that the ULDL-based prediction consistently achieves lower NMSE compared to the DLDL-based prediction. Furthermore, incorporating the time lag feature significantly improves NMSE performance for DLDL prediction.

To provide a more comprehensive evaluation of channel prediction performance, we analyse the statistical behaviour of phase and magnitude errors, as depicted in Fig. 15a and Fig. 15b. For each SNR level, the mean and standard deviation of both error components are computed over all samples, subcarriers, and OFDM symbols, enabling joint characterisation of average accuracy and error dispersion beyond the aggregate NMSE metric. Both phase and magnitude errors exhibit larger mean values and significantly higher variance at low SNR, reflecting increased sensitivity of complex channel coefficients to noise and deep fading, whereas at moderate and high SNRs the errors contract sharply and exhibit tightly bounded distributions, indicating stable and reliable reconstruction. These results align with the training objective, where the model is optimised using an NMSE loss applied to the real and

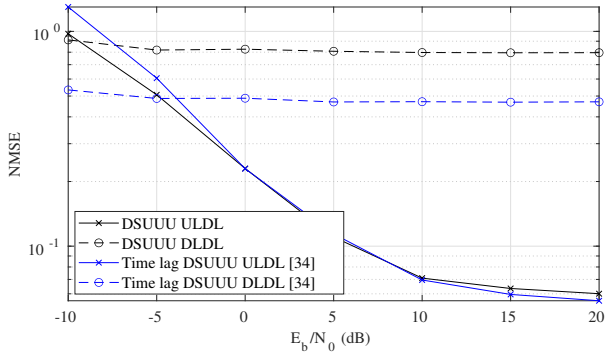
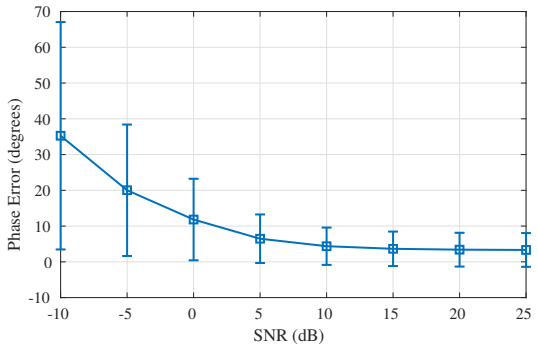
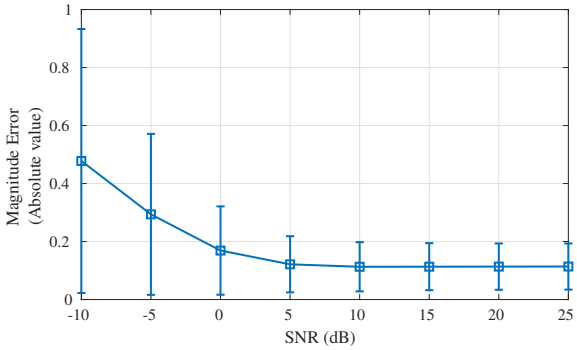


Fig. 14: Uplink-to-downlink (ULDL) and downlink-to-downlink (DLDL) prediction with additional feature time lag.

imaginary components of the complex channel coefficients, thereby jointly minimising amplitude and phase deviations. From a system perspective, the low phase-error variance is particularly significant, as it suggests that the predicted CSI is sufficiently accurate to support beamforming in LEO downlink scenarios. Overall, the results confirm that the proposed TD-CNN-LSTM framework effectively preserves magnitude and phase information across subcarriers under varying channel conditions.



(a) Phase error distribution in degrees.



(b) Magnitude error distribution.

Fig. 15: Statistical analysis of channel prediction errors across varying SNR levels.

VII. Conclusion and Future Work

This study presents an effective channel prediction framework tailored for LEO satellite systems that addresses the unique challenges of dynamic channel conditions, Doppler effects, and constrained computational resources. By integrating an offline-online learning approach, the proposed method enables real-time adaptation to rapidly changing environments, which is critical during emergency scenarios where reliable communication is essential. Furthermore, split learning distributes computational tasks across satellites and user equipment, making the approach practical for deployment on resource-limited LEO platforms. Simulation results validate the model's performance across various LEO configurations, demonstrating lower NMSE values and improved prediction accuracy as the number of consecutive uplink slots increases. These findings highlight the model's potential for enhancing the reliability and responsiveness of satellite communications in mission-critical 6G applications.

This study focuses on predicting a single downlink slot using multiple historical uplink observations, which is sufficient to demonstrate the effectiveness of the proposed split-learning-based framework. Extending the model to support the prediction of multiple downlink slots is left for future work. Rain attenuation is employed as a proxy for severe atmospheric impairments commonly encountered in NTN scenarios; however, since rain fading alone does not fully capture all propagation effects, extensions to other conditions will be investigated. Moreover, user locations are assumed to be fixed during the satellite service interval, resulting in static cluster membership and Doppler shifts solely induced by satellite motion. Future work will consider mobile-user scenarios with adaptive clustering, dynamic cluster maintenance, and varying Doppler conditions. Finally, while ideal synchronisation and reliable exchange of intermediate representations are assumed, the proposed framework can be extended to account for imperfect synchronisation, link errors, and asynchronous or delayed split learning updates, and evaluating the robustness of split learning under practical NTN impairments remains an important direction for future study.

References

[1] G. Araniti, A. Iera, S. Pizzi, and F. Rinaldi, "Toward 6G non-terrestrial networks," *IEEE Network*, vol. 36, no. 1, pp. 113–120, 2022.

[2] 3GPP, "Solutions for NR to support non-terrestrial networks (NTN)," 2023.

[3] 3GPP, "NR and NG-RAN overall description; stage-2," 2025.

[4] Q. T. Ngo and et al., "Optimizing spectrum sensing in cognitive GEO-LEO satellite networks: Overcoming challenges for effective spectrum utilization," *IEEE Vehicular Technology Magazine*, pp. 2–11, 2025.

[5] J. Zheng, X. Wang, and Q. Li, "Design of emergency rescue satellite communication system based on high-throughput satellite," in *2023 IEEE 6th International Conference on Electronics and Communication Engineering (ICECE)*, 2023, pp. 100–104.

[6] G. Raja and G. Saravanan, "Eco-friendly disaster evacuation framework for 6G connected and autonomous vehicular networks," *IEEE Transactions on Green Communications and Networking*, vol. 6, no. 3, pp. 1368–1376, 2022.

[7] Y. Zhang, A. Liu, P. Li, and S. Jiang, "Deep learning (DL)-based channel prediction and hybrid beamforming for LEO satellite massive MIMO system," *IEEE Internet of Things Journal*, vol. 9, no. 23, pp. 23 705–23 715, 2022.

[8] 3GPP, "3rd generation partnership project; technical specification group radio access network; study on new radio (NR) to support non-terrestrial networks," 2020.

[9] M. Vázquez, A. Pérez-Neira, D. Christopoulos, S. Chatzinotas, B. Ottersten, P.-D. Arapoglou, A. Ginesi, and G. Taricco, "Precoding in multibeam satellite communications: Present and future challenges," *IEEE Wireless Communications*, vol. 23, no. 6, pp. 88–95, 2016.

[10] O. Kodheli, E. Lagunas, N. Maturo, S. K. Sharma, B. Shankar, J. F. M. Montoya, J. C. M. Duncan, D. Spano, S. Chatzinotas, S. Kisseleff, J. Querol, L. Lei, T. X. Vu, and G. Goussetis, "Satellite communications in the new space era: A survey and future challenges," *IEEE Communications Surveys Tutorials*, vol. 23, no. 1, pp. 70–109, 2021.

[11] Y. Zhang, Y. Wu, A. Liu, X. Xia, T. Pan, and X. Liu, "Deep learning-based channel prediction for LEO satellite massive MIMO communication system," *IEEE Wireless Communications Letters*, vol. 10, no. 8, pp. 1835–1839, 2021.

[12] O. Abbasi and G. Kaddoum, "Channel aging-aware LSTM-based channel prediction for satellite communications," *IEEE Networking Letters*, vol. 6, no. 3, pp. 183–187, 2024.

[13] G.-Y. Chang, C.-K. Hung, and C.-H. Chen, "A CSI prediction scheme for satellite-terrestrial networks," *IEEE Internet of Things Journal*, vol. 10, no. 9, pp. 7774–7785, 2023.

[14] M. Alrabeiah and A. Alkhateeb, "Deep learning for TDD and FDD massive mimo: Mapping channels in space and frequency," in *53rd Asilomar Conference on Signals, Systems, and Computers*, 2019, pp. 1465–1470.

[15] J. Vieira and et al., "Deep convolutional neural networks for massive MIMO fingerprint-based positioning," in *IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2017, pp. 1–6.

[16] K. Weththasinghe, B. Jayawickrama, and Y. He, "Machine learning-based channel estimation for 5G New Radio," *IEEE Wireless Communications Letters*, vol. 13, no. 4, pp. 1133–1137, 2024.

[17] 3GPP, "3rd generation partnership project; technical specification group radio access network; NR; physical channels and modulation v16.10.0,," 2022.

[18] C. Luo, J. Ji, Q. Wang, X. Chen, and P. Li, "Channel state information prediction for 5G wireless communications: A deep learning approach," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 1, pp. 227–236, 2020.

[19] H. Hafi, B. Brik, P. A. Frangoudis, A. Ksentini, and M. Bagaa, "Split federated learning for 6G enabled-networks: Requirements, challenges, and future directions," *IEEE Access*, vol. 12, pp. 9890–9930, 2024.

[20] G. S. Hukkeri, R. H. Goudar, G. M. Dhananjaya, V. N. Rathod, and S. Ankalaki, "A comprehensive survey on split-fed learning: Methods, innovations, and future directions," *IEEE Access*, vol. 13, pp. 46 312–46 333, 2025.

[21] Z. Lin, G. Zhu, Y. Deng, X. Chen, Y. Gao, K. Huang, and Y. Fang, "Efficient parallel split learning over resource-constrained wireless edge networks," *IEEE Transactions on Mobile Computing*, pp. 1–16, 2024.

[22] J. Sun, C. Wu, S. Mumtaz, J. Tao, M. Cao, M. Wang, and V. Frascolla, "An efficient privacy-aware split learning framework for satellite communications," *IEEE Journal on Selected Areas in Communications*, pp. 1–1, 2024.

[23] J. Liu, X. Zhang, R. Zhang, T. Huang, and F. R. Yu, "Reliable and low-overhead clustering in LEO small satellite networks," *IEEE Internet of Things Journal*, vol. 9, no. 16, pp. 14 844–14 856, 2022.

[24] X. Zhu and Z. Wang, "Service throughput oriented clustering in LEO satellite networks," in *4th International Conference on Frontiers Technology of Information and Computer (ICFTIC)*, 2022, pp. 842–850.

[25] X. Zhu, C. Jiang, L. Kuang, N. Ge, and J. Lu, "Non-orthogonal multiple access based integrated terrestrial-satellite networks," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 10, pp. 2253–2267, 2017.

[26] K. Weththasinghe, Q. T. Ngo, Y. He, and B. Jayawickrama, "Optimising beam size in multibeam LEO satellite networks: Addressing interbeam interference, doppler shift, and frequency reuse," *IEEE Transactions on Aerospace and Electronic Systems*, pp. 1–15, 2024.

[27] Q. T. Ngo, Y. He, B. A. Jayawickrama, and E. Dutkiewicz, "A fast fuzzy DRL-based joint beam design and power allocation for multi-beam GEO-LEO coexisting satellite networks," *IEEE Transactions on Wireless Communication*, pp. 1–14, 2025.

[28] Y. Liu, Z. Tan, H. Hu, L. J. Cimini, and G. Y. Li, "Channel estimation for OFDM," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1891–1908, 2014.

[29] Y. Yang, F. Gao, G. Y. Li, and M. Jian, "Deep learning-based downlink channel prediction for FDD massive MIMO system," *IEEE Communications Letters*, vol. 23, no. 11, pp. 1994–1998, 2019.
[30] N. Jindal, "Mimo broadcast channels with finite-rate feedback," *IEEE Transactions on Information Theory*, vol. 52, no. 11, pp. 5045–5060, Nov. 2006.
[31] O. Raeesi, A. Gokceoglu, Y. Zou, E. Björnson, and M. Valkama, "Performance analysis of multi-user massive mimo downlink under channel non-reciprocity and imperfect csi," *IEEE Access*, vol. 5, pp. 22 517–22 534, 2017.
[32] Radiocommunication Sector of International Telecommunication Union, "Recommendation ITU-R P.838-3: Specific Attenuation Model for Rain for Use in Prediction Methods," 2005.
[33] "State-of-the-art small spacecraft technology," NASA, Tech. Rep., March 2025, accessed: 2025-05-20. [Online]. Available: https://www.nasa.gov/smallsat-institute/sst-soa/small-spacecraft-avionics/
[34] "Antelope DPU datasheet." KP Labs, Tech. Rep., accessed: 2025-05-20. [Online]. Available: https://www.kplabs.space/
[35] H. Kim, J. Kim, and D. Hong, "Dynamic TDD systems for 5G and beyond: A survey of cross-link interference mitigation," *IEEE Communications Surveys Tutorials*, vol. 22, no. 4, pp. 2315–2348, 2020.
[36] S. N. Shukla and B. M. Marlin, "Interpolation-prediction networks for irregularly sampled time series," *arXiv preprint arXiv:1909.07782*, 2019.

Kithmini Weththasinghe received the B.Sc. (Hons.) degree in Electronic & Telecommunication Engineering and the M.Sc. in Telecommunications from University of Moratuwa, Sri Lanka in 2016 and 2022, respectively. She is currently a Ph.D. candidate in the School of Electrical and Data Engineering, University of Technology Sydney, Ultimo, NSW, Australia. Her research interests include machine learning in wireless communications, satellite communications and cognitive radio.

Quynh Tu Ngo received the B.Sc. degree (Magna Cum Laude) in Electrical Engineering from California State University Los Angeles, CA, USA, in 2013, the M.Sc. degree in Telecommunications from Vietnam National University - University of Sciences, Ho Chi Minh City, Vietnam, in 2016, and the Ph.D. degree in Computer Science from La Trobe University, VIC, Australia, in 2023.
She is currently a Postdoctoral Research Fellow with the School of Electrical and Data Engineering, University of Technology Sydney, Australia. Her research interests include satellite communications, IoT networks, intelligent non-terrestrial networks, and machine learning in wireless communications and networking.

Ying He received PhD in Engineering (Telecommunications) from University of Technology Sydney, Australia, in 2017. She is currently a Senior Lecturer with the Faculty of Engineering and IT, University of Technology Sydney. Her research interests are wireless communication networks with machine learning, satellite communication, spectrum sharing and vehicular communication.

Beeshanga Jayawickrama received the B.E. degree (First-Class Hons.) in Telecommunications Engineering and the Ph.D. degree in Electronic Engineering from Macquarie University, Macquarie Park, NSW, Australia, in 2011 and 2015, respectively.
He is currently affiliated as a Visiting Fellow at University of Technology Sydney, Australia. He was extensively involved in spectrum sensing and interference mitigation research for spectrum access systems. His research interests include non-terrestrial networks, 5G/6G, cognitive radio, and signal processing.