

Beam-hopping optimization for regenerative satellites using a hybrid LSTM-transformer model

Issiakou Worou Wouignanso¹, Olfa Ben Yahia¹, Stéphane Martel², Guillaume Mantelet², Gunes Karabulut Kurt¹, Brunilde Sansò¹

¹ Poly-Grames Research Center, Department of Electrical Engineering, Polytechnique Montréal, Montréal, QC, Canada, ² MDA Space, 21025 Trans-Canada Hwy, Sainte-Anne-de-Bellevue, QC H9X 3R2, Canada

Corresponding author: Issiakou Worou Wouignanso, issiakou.worou-wouignanso@polymtl.ca

This paper presents a hybrid Long Short-Term Memory (LSTM)–transformer model for optimizing Beam Hopping (BH) in multibeam satellite communication systems. BH is a dynamic resource allocation technique that improves spectral efficiency by selectively illuminating ground regions based on traffic demand and channel conditions. The proposed system takes as inputs traffic demand measurements, Channel State Information (CSI), and historical allocation data, and produces as output a BH schedule together with associated Key Performance Indicators (KPIs). The architecture combines three stacked bidirectional LSTM layers, which capture temporal traffic dynamics at multiple scales, with a transformer encoder, which identifies long-range inter-beam spatial correlations through multi-head self-attention. Training is driven by a four-term multi-objective loss that jointly optimizes Spectral Efficiency (SE), Jain’s fairness, Quality of Service (QoS) compliance and power utilization. Monte Carlo simulations with stochastic channel and traffic realizations demonstrate that the proposed method improves SE compared with the Deep learning-Based Optimization (DBO) benchmark. The hybrid model also outperforms a pure transformer baseline trained under identical conditions, while reducing training time and maintaining a favourable trade-off between performance and computational cost. In addition, the proposed approach achieves balanced performance in terms of fairness, QoS compliance, power utilization and Signal-to-Interference-plus- Noise Ratio (SINR). A sensitivity analysis of the loss-weighting coefficients further characterizes the trade-off between SE and fairness.

Keywords: Beam hopping (BH), Long Short-Term Memory (LSTM), multibeam, resource allocation, satellites, spectral efficiency, transformer

1. INTRODUCTION

With the rapid evolution of communication systems and the explosive growth of user terminals, traditional single wide beam satellites can no longer meet service requirements. In large multibeam architectures, traffic demand is highly imbalanced across beams: demand remains unmet in hot spots while capacity is underutilized in cold spots, reducing revenue and overall resource efficiency [1]. Consequently, the satellite industry has shifted focus toward Non-Geostationary Orbit (NGSO) with multiple beams operating at higher frequency bands (e.g., Ku and Ka) to serve the broadband market better. However, achieving high data rates under highly non-uniform demand requires efficient resource allocation.

Satellite resources are scarce and costly, making their efficient allocation essential. In recent years, flexible resource-allocation strategies enabled by flexible payload architectures have emerged as promising solutions. In [2], the authors presented a novel vision for future flexible payload architectures, named Extremely High Throughput Satellite (EHTS). In practice, payload flexibility can be realized through dynamic resource allocation across spectrum, time, and transmit power, allowing the system to adapt resource distribution to spatial and temporal traffic variations.

In this context, Beam Hopping (BH) is one of the key techniques for improving payload flexibility and increasing Spectral Efficiency (SE) in response to non-uniform and time-varying traffic demand. Unlike fixed-beam systems, BH dynamically steers and activates a subset of beams to illuminate specific regions for a designated dwell time. Beam activations follow a beam illumination pattern that adapts to time-varying demand [3]. However, the dynamic and unpredictable nature of traffic requires intelligent algorithms for optimal beam scheduling and efficient resource allocation.

Addressing this scheduling problem in Low Earth Orbit (LEO) satellite systems is particularly challenging, since BH optimization must account not only for time-varying traffic demand but also for constraints that are specific to the space segment. First, the continuous mobility of LEO satellites produces a time-varying coverage footprint, leading to frequent changes in the served cells, traffic distribution, elevation angles, and channel conditions [4]. Therefore, scheduling decisions must anticipate future traffic and channel variations over several time slots. Second, the satellite payload operates under strict constraints on onboard computational power, memory, and thermal dissipation, which limit the complexity of real-time BH algorithms. Third, LEO satellite channels are highly affected by several propagation impairments, including fading, atmospheric turbulence, path loss due to long transmitter–receiver distances, cloud formations, weather effects, masking effects from obstacles and shadowing, and Doppler shifts [5]. Fourth, the number of simultaneously active beams is bounded by the onboard power budget and frequency reuse plan, creating a combinatorial resource-allocation problem whose search space grows as 2^N with the number of beams N . Finally, traffic demand over the satellite coverage area exhibits strong geographic and diurnal non-uniformities, which are further amplified by the continuous motion of the LEO footprint across different regions [6, 7]. These specificities motivate learning-based methods capable of capturing temporal dependencies and inter-beam spatial correlations simultaneously, which is precisely the objective of the proposed architecture.

Given these challenges, several approaches have been proposed to address BH optimization under dynamic traffic and resource constraints. BH has been studied from several perspectives. Convex optimization methods formulate BH as a structured problem but suffer from high computational complexity that precludes real-time use [7]. Game-theoretic formulations [8] model BH as a multi-agent game, but convergence is not guaranteed in highly dynamic settings. Heuristic and linear programming approaches fail to adapt to real-time traffic variations [6, 7]. The seminal work of Lei et al. [1] introduced a Deep learning-Based Optimization (DBO) algorithm that combines a Fully Connected - Deep Neural Network (FC-DNN) with a linear programming solver.

The DNN predicts a cardinality feature vector $\mathbf{v} \in \{0, 1\}^N$ indicating which snapshot cardinalities appear in the optimal BH cycle; this dramatically reduces the search space fed to the solver. The DBO algorithm achieves near-optimal performance with computation times orders of magnitude lower than branch-and-bound, making it the most relevant Deep Learning (DL) benchmark for the present work. Reinforcement Learning (RL) [9] learns scheduling policies through environment interaction but is not adopted here due to the need for online exploration and the risk of instability under non-stationary satellite conditions; its motivation informs our exploration of the decision space. Recurrent neural networks combined with attention mechanisms have been shown to enhance traffic flow management [10, 11]. Short-Term Memory (LSTM)-based solutions have been applied in satellite frequency-hopping scenarios [12] and in spatio-temporal traffic prediction when combined with Convolutional Neural Networks (CNNs) [13]. Still, they may struggle with long-range dependencies [14, 15].

Transformer models rely on self-attention to handle long sequences [16, 17] and have been applied to resource allocation in terrestrial wireless networks [18]. In the satellite domain, recent work has begun exploring transformer-based architectures for traffic prediction in LEO systems to support dynamic BH [19]. However, these approaches focus on traffic forecasting only and do not address the joint optimization of beam scheduling and resource allocation. However, to the best of the authors' knowledge, the use of a hybrid LSTM-transformer architecture with a stacked multi-LSTM configuration for BH optimization in satellite systems has not yet been explored. Moreover, whether such a hybrid architecture can outperform a pure transformer baseline in this context, and at what computational cost, remains an open question. This gap is addressed in this paper.

Different from existing works, the main contributions of this paper are as follows.

- We propose a hybrid bidirectional LSTM (Bi-LSTM)–transformer model for BH optimization based on a three-layer stacked Bi-LSTM front end, in which the Bi-LSTM layers are arranged sequentially and the output of each layer is fed to the next layer before being processed by the transformer encoder. The model uses traffic demand, Channel State Information (CSI) and historical allocation data as inputs to generate a BH allocation pattern and corresponding Key Performance Indicators (KPIs).
- We benchmark the proposed method against the DBO baseline of [1] and against a pure transformer model trained under identical conditions, providing a fair comparison that isolates the contribution of the recurrent front end.

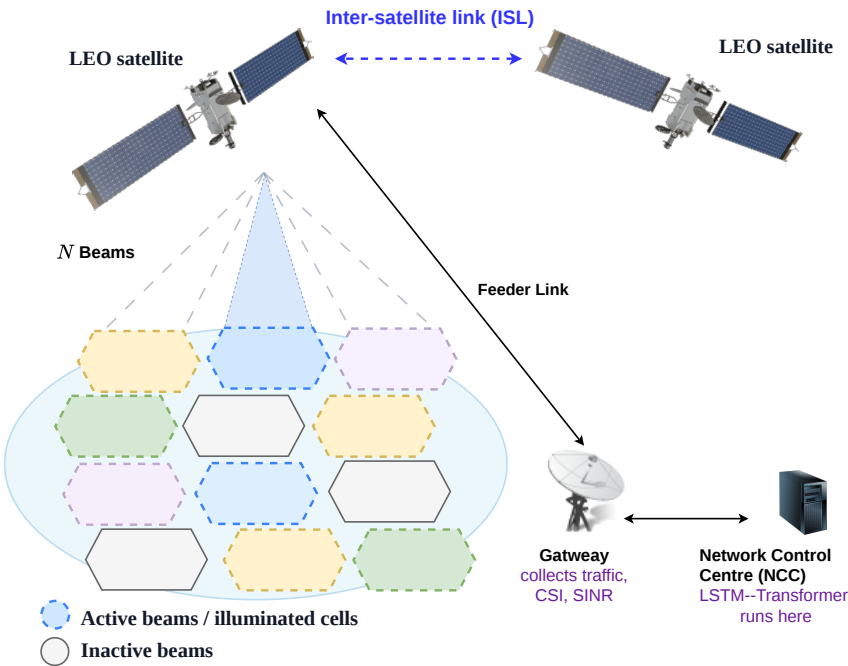


Figure 1 – Proposed system model for a multi-beam LEO satellite with beam hopping.

- We conduct Monte Carlo simulations under stochastic channel and traffic conditions to evaluate the proposed method. The results show improved performance in terms of SE, fairness, Quality of Service (QoS) compliance, power utilization and Signal-to-Interference-plus Noise Ratio (SINR) compared with representative benchmark methods. In addition, the sensitivity analysis characterizes the SE–fairness trade-off as a function of the loss-weighting parameter α .
- We evaluate the scalability of the proposed model with respect to the number of beams, showing that the Bi-LSTM-transformer architecture maintains higher SE and degrades more slowly than the pure transformer and DBO baselines in higher multibeam scenarios.

The rest of this paper is organized as follows. Section 2 describes the system model. Section 3 presents the simulation setup, Monte Carlo results with Confidence Intervals (CIs) and ablation and sensitivity analyses. Section 4 discusses practical applicability. Section 5 concludes the paper. Finally, the list of abbreviations is given in Table 1.

2. SYSTEM MODEL AND PROPOSED ARCHITECTURE

2.1 Overview and input/output description

As depicted in Fig. 1, we consider an LEO satellite that serves multiple ground regions using the BH technique. Following the considered system model, each beam is

Table 1 – List of abbreviations

Abbreviation	Definition
ADAM	Adaptive Moment estimation
BH	Beam Hopping
LSTM	long Short-Term Memory
Bi-LSTM	Bidirectional Long Short-Term Memory
CI	Confidence Interval
CPLEX	C Programming Language Extension
CSI	Channel State Information
DBO	Deep learning-Based Optimization
DL	Deep Learning
DVB-S2X	Digital Video Broadcasting - Satellite
FFN	Feed-Forward Network
HTS	High Throughput Satellite
KPI	Key Performance Indicator
LEO	Low Earth Orbit
LSTM	Long Short-Term Memory
NCC	Network Control Centre
QoS	Quality of Service
RL	Reinforcement Learning
SCIP	Solving Constraint Integer Programs
SE	Spectral Efficiency
SINR	Signal-to-Interference-plus-Noise Ratio.

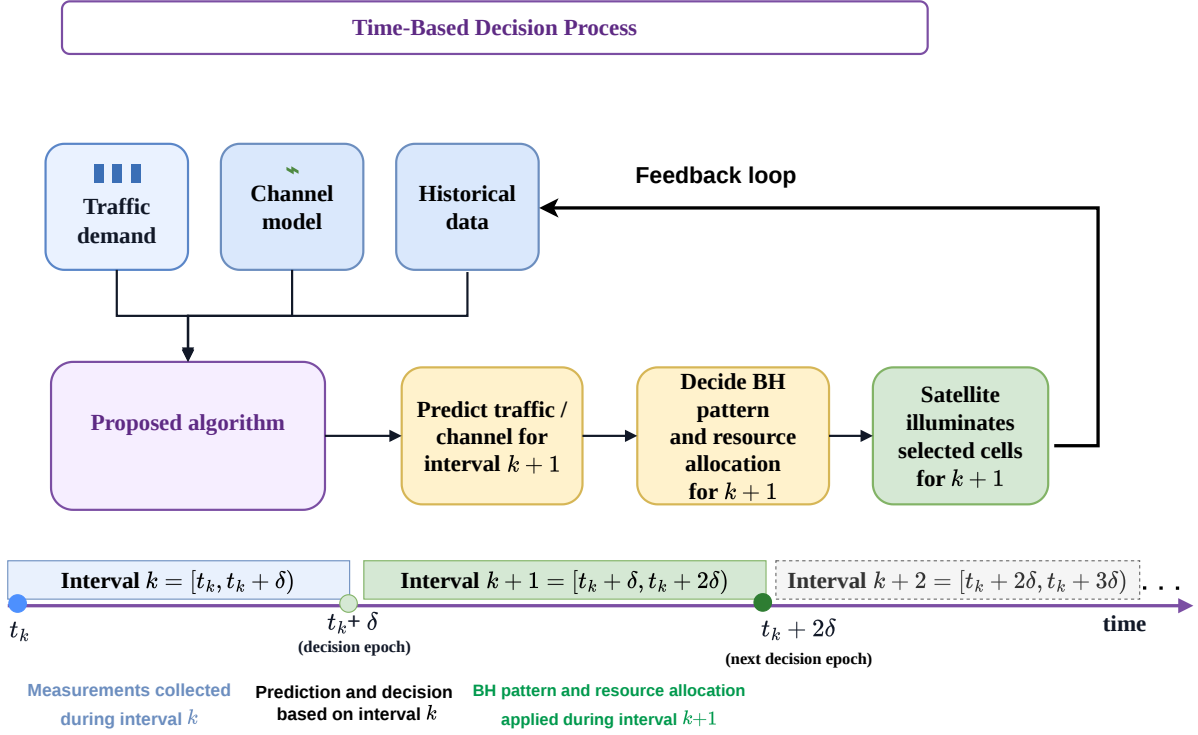


Figure 2 – Time-based decision process for beam hopping and resource allocation.

associated with a corresponding ground cell/region; therefore, the terms beam and cell are used interchangeably in the following. At each interval, the satellite illuminates only a subset of beams, defining the BH pattern, while the remaining beams stay inactive. The objective is to select the beams to be illuminated and determine their activation pattern and dwell times according to the expected traffic and channel conditions. In the current implementation, the per-beam transmit power and bandwidth are treated as fixed system parameters rather than dynamically optimized variables.

The proposed framework adopts a time-based decision process, as illustrated in Fig. 2. During each interval $k = [t_k, t_{k+1}]$, the satellite gateway collects traffic demand, CSI, and historical allocation data. In practice, CSI can be obtained using conventional channel estimation techniques, such as pilot-based estimation and channel sounding [20]. In this work, we assume that the most recent CSI estimates are available at the NCC at each scheduling decision interval. However, in practical LEO systems, channel estimation overhead and feedback delay may introduce CSI aging and affect the timeliness of the scheduling decisions. These effects are not explicitly modelled in the current framework and are discussed as an implementation limitation in Section 4. Periodic model retraining or online fine-tuning may also be considered to account for changes in the channel and traffic dynamics over longer time scales.

These measurements are processed by the proposed hybrid LSTM-transformer model, which predicts the traffic and channel conditions for the next interval and generates beam activation probabilities. The resulting probabilities are used to construct the BH pattern, from which the active beams and their dwell times are determined. The NCC communicates the computed BH schedule to the satellite through the feeder link, while the transmit power and bandwidth remain fixed according to the considered system configuration. The satellite then illuminates the selected beams, and the measured outcomes, including SINR, throughput, and QoS indicators, are fed back to update the historical state and support the next decision cycle.

This recurrent process, involving prediction, scheduling, illumination, measurement and feedback, motivates the use of a hybrid architecture that extracts hierarchical temporal features and long-range dependencies from the joint representation of the beam states. The Bi-LSTM component extracts temporal patterns from the observation window, while the transformer further models dependencies across this temporal representation. Since the input at each time step contains information from all beams, inter-beam information is implicitly included in the learned representation. The active and inactive beam states therefore evolve from one scheduling interval to the next according to the learned activation probabilities, which depend on the observed traffic and channel conditions.

Table 2 – Main notation used in the proposed framework.

Symbol	Definition
N	Total number of satellite beams.
i, j	Beam indices; i denotes the considered beam and j an interfering beam.
t	Current discrete scheduling time index.
τ	Future scheduling-step index within the prediction horizon.
T_{obs}	Length of the observation window used as input to the learning model.
T_{hor}	Prediction/scheduling horizon length.
$\mathbf{D}_t$	Traffic demand vector across all beams at time t .
$\mathbf{G}_t$	Channel gain matrix at time t .
$\mathbf{S}_{t-1}$	Historical system state containing previous scheduling, SINR, and QoS information.
d_S	Number of features in the historical state $\mathbf{S}_{t-1}$.
$\mathbf{X}_t$	Raw input vector combining traffic demand, channel gains, and historical state.
$\mathbf{Z}_t$	Input sequence formed over the observation window T_{obs} .
d_h	Hidden-state dimension of each unidirectional LSTM.
d_{model}	Feature dimension used by the transformer encoder.
$\mathbf{A}_t$	Attention-enhanced representation produced by the transformer encoder.
$\mathbf{p}(t)$	Continuous beam-activation probability vector generated by the allocation head.
$\mathbf{B}_{t+1}$	Binary BH schedule generated for the next scheduling interval.
ϑ	Number of beams selected by the Top_{ϑ} operation.
$T_{\text{dwell}}^{(i,m)}$	Dwell time of beam i during its m -th consecutive activation interval.
$\mathcal{L}_{\text{MO}}$	Multi-objective loss function used to train the proposed model.
$\alpha, \beta, \gamma, \delta$	Weights of the SE, fairness, QoS and power-related loss terms, respectively.
η	Target power-utilization parameter used in the power-related loss term.

Let N denote the total number of satellite beams covering the service area. The system is modelled in discrete time, where each index t represents the smallest scheduling unit at which BH and resource allocation decisions are updated. This abstracted time representation is consistent with practical frame-based BH systems, where a physical frame can be divided into slots or sub-slots. In the following, $i \in \{1, \dots, N\}$ indicates the beam index, t is the current decision epoch and τ denotes a future scheduling step within the considered horizon. The parameters T_{obs} and T_{hor} denote the observation window length and the scheduling horizon length, respectively. More specifically, at each discrete time interval t , the NCC receives three categories of inputs given as

1. Traffic demand $\mathbf{D}_t = [d_t^{(1)}, \dots, d_t^{(N)}]$: aggregated demand per beam, expressed in bits per second (bps). Each component $d_t^{(i)}$ represents the total traffic load requested by users within beam i at time t , as estimated from gateway measurements. The demand is modelled as a combination of heterogeneous traffic types, including Poisson voice traffic, on/off video traffic and log-normal data traffic.
2. CSI $\mathbf{G}_t$: forward-link channel-state matrix describing the channel gains between the satellite beams and the corresponding ground regions. The per-beam channel gains used as learning features are extracted from $\text{diag}(\mathbf{G}_t)$. Inter-beam interference and thermal noise are considered separately in the SINR calculation rather than as components of the channel gain itself.

3. Historical state $\mathbf{S}_{t-1} \in \mathbb{R}^{d_S}$: aggregated information from the previous scheduling interval, consisting of three scalar features: a summary of the previous BH decisions, the average SINR, and the average QoS indicator. In our implementation, $d_S = 3$.

Based on these inputs, the system produces the following outputs:

- (a) A BH schedule $\mathbf{B}_{t+1} \in \{0, 1\}^{T_{\text{hor}} \times N}$, where entry (τ, i) equals 1 if beam i is active at future scheduling step τ . The transmit power and bandwidth are not output variables of the learning algorithm and remain fixed system parameters.
- (b) A set of KPIs computed for that schedule: SE (b/s/Hz), Jain's fairness index, QoS compliance rate, power utilization ratio, and average SINR.

These KPIs are reinjected into the feedback loop as part of the historical state $\mathbf{S}_t$, closing the adaptation loop. The overall architecture of the proposed BH system is illustrated in Fig. 3. It processes temporal sequences containing the joint state of all beams and produces the activation decisions used to construct the BH schedule.

The main notation used throughout the proposed framework is summarized in Table 2.

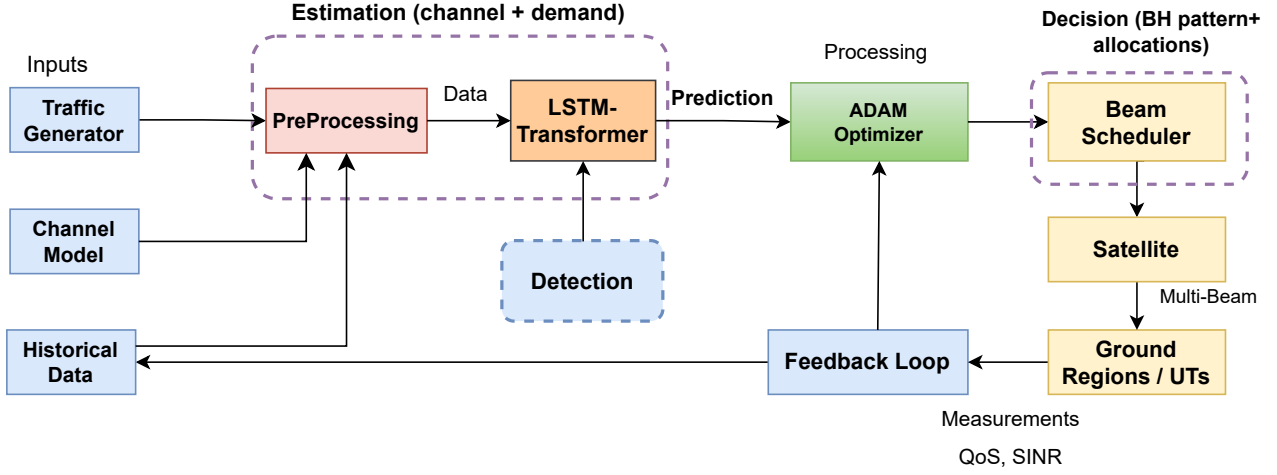


Figure 3 – Functional architecture of the proposed BH optimization system. Inputs: traffic demand, CSI, historical state. Output: BH schedule and KPIs.

2.2 Preprocessing module

The preprocessing module constructs the input features for the learning-based model from the previously defined system variables. At each discrete time step t corresponding to a subslot, the raw input vector is defined as

$$\mathbf{X}_t = [\mathbf{D}_t, \text{diag}(\mathbf{G}_t), \mathbf{S}_{t-1}] \in \mathbb{R}^{(2N+d_s)}. \quad (1)$$

This construction combines traffic demand, per-beam channel information, and historical feedback into a common system-state representation. The raw features are normalized feature-wise using scaling parameters computed only from the training subset. In the current implementation, each feature is scaled using its maximum value observed in the corresponding training data. The same scaling parameters are then applied to the validation data, avoiding information leakage between the training and validation subsets. The normalized input vector is therefore expressed as

$$\tilde{\mathbf{X}}_t = \mathcal{N}(\mathbf{X}_t), \quad (2)$$

where $\mathcal{N}(\cdot)$ denotes the feature-wise normalization operator whose parameters are estimated exclusively from the training data. To capture temporal dependencies, the normalized feature vectors are assembled into a sliding observation window of length T_{obs} :

$$\mathbf{Z}_t = [\tilde{\mathbf{X}}_{t-T_{\text{obs}}+1}, \dots, \tilde{\mathbf{X}}_t] \in \mathbb{R}^{T_{\text{obs}} \times (2N+d_s)}. \quad (3)$$

This sequence represents the joint temporal evolution of traffic demand, channel conditions, and system feedback over the last T_{obs} sub-slots, and serves as input to the subsequent learning module. The inclusion of $\mathbf{S}_{t-1}$ enables the model to account for past BH scheduling decisions and their observed impact on the system KPIs,

improving the representation of the closed-loop dynamics. This preprocessing module, shown in Fig. 3, ensures a structured and consistent representation of the system state for the learning-based estimation stage.

2.3 Hybrid LSTM-transformer estimation module

2.3.1 Three-layer Bi-LSTM front end

The model employs a three-layer Bi-LSTM stack to capture temporal dynamics at multiple scales. Given the input sequence $\mathbf{Z}_t \in \mathbb{R}^{T_{\text{obs}} \times (2N+d_s)}$, which encodes the temporal evolution of traffic demand $\mathbf{D}_t$, per-beam channel gains $\text{diag}(\mathbf{G}_t)$ and historical system state $\mathbf{S}_{t-1}$ over the last T_{obs} sub-slots, the Bi-LSTM layers extract hierarchical temporal representations.

Rather than assigning a predefined temporal scale to each layer, the stacked Bi-LSTM architecture progressively transforms the input sequence into higher-level temporal representations, allowing the network to capture traffic fluctuations and dependencies over different time scales. This hierarchical behaviour emerges from the learning process and is consistent with previous studies on deep recurrent networks [14, 15]. To further evaluate the benefit of the three-layer architecture and verify that the additional layers do not merely introduce redundant information, we conducted an ablation study with one, two, and three Bi-LSTM layers. The corresponding results are presented in Section 3.

For the stacked architecture, the input to the first Bi-LSTM layer is the preprocessed feature vector, while each subsequent layer receives the output of the previous layer. We therefore define

$$\mathbf{v}_t^{(\ell)} = \begin{cases} \mathbf{z}_t, & \ell = 1, \\ \mathbf{u}_t^{(\ell-1)}, & \ell = 2, 3. \end{cases} \quad (4)$$

For layer $\ell \in \{1, 2, 3\}$, the forward and backward hidden states are then computed as

$$\vec{\mathbf{u}}_t^{(\ell)} = \text{LSTM}_{\text{fwd}}^{(\ell)}(\mathbf{v}_t^{(\ell)}, \vec{\mathbf{u}}_{t-1}^{(\ell)}), \quad \overleftarrow{\mathbf{u}}_t^{(\ell)} = \text{LSTM}_{\text{bwd}}^{(\ell)}(\mathbf{v}_t^{(\ell)}, \overleftarrow{\mathbf{u}}_{t+1}^{(\ell)}). \quad (5)$$

The output of the ℓ -th Bi-LSTM layer is obtained by concatenating the forward and backward states:

$$\mathbf{u}_t^{(\ell)} = [\vec{\mathbf{u}}_t^{(\ell)} \parallel \overleftarrow{\mathbf{u}}_t^{(\ell)}]. \quad (6)$$

This formulation explicitly represents the sequential stacking of the three Bi-LSTM layers, where each layer progressively refines the temporal representation produced by the preceding layer. The final representation produced by the Bi-LSTM block at time t is given by:

$$\mathbf{u}_t = \mathbf{u}_t^{(3)}. \quad (7)$$

The sequence of Bi-LSTM hidden states over the observation window is denoted by

$$\mathbf{U}_t = [\mathbf{u}_{t-T_{\text{obs}}+1}, \dots, \mathbf{u}_t] = \text{Bi-LSTM}(\mathbf{Z}_t), \quad (8)$$

where $\mathbf{U}_t \in \mathbb{R}^{T_{\text{obs}} \times 2d_h}$ and d_h is the hidden size of each unidirectional LSTM.

The internal forward and backward LSTM units follow the standard gating formulation introduced in [21], while the bidirectional processing follows [22].

To clarify the internal operation of each Bi-LSTM layer, we consider the standard LSTM gating mechanism in both the forward and backward directions. For compactness, let $r \in \{\rightarrow, \leftarrow\}$ denote the recurrent direction, and the corresponding neighboring time index is defined as

$$t_r = \begin{cases} t-1, & r = \rightarrow, \\ t+1, & r = \leftarrow. \end{cases} \quad (9)$$

The LSTM gating operations for layer ℓ and direction r are then given by

$$\mathbf{f}_t^{(\ell,r)} = \sigma(W_f^{(\ell,r)}[\mathbf{u}_{t_r}^{(\ell,r)}, \mathbf{v}_t^{(\ell)}] + b_f^{(\ell,r)}), \quad (10)$$

$$\mathbf{i}_t^{(\ell,r)} = \sigma(W_i^{(\ell,r)}[\mathbf{u}_{t_r}^{(\ell,r)}, \mathbf{v}_t^{(\ell)}] + b_i^{(\ell,r)}), \quad (11)$$

$$\tilde{\mathbf{c}}_t^{(\ell,r)} = \tanh(W_c^{(\ell,r)}[\mathbf{u}_{t_r}^{(\ell,r)}, \mathbf{v}_t^{(\ell)}] + b_c^{(\ell,r)}), \quad (12)$$

$$\mathbf{c}_t^{(\ell,r)} = \mathbf{f}_t^{(\ell,r)} \odot \mathbf{c}_{t_r}^{(\ell,r)} + \mathbf{i}_t^{(\ell,r)} \odot \tilde{\mathbf{c}}_t^{(\ell,r)}, \quad (13)$$

$$\mathbf{o}_t^{(\ell,r)} = \sigma(W_o^{(\ell,r)}[\mathbf{u}_{t_r}^{(\ell,r)}, \mathbf{v}_t^{(\ell)}] + b_o^{(\ell,r)}), \quad (14)$$

$$\mathbf{u}_t^{(\ell,r)} = \mathbf{o}_t^{(\ell,r)} \odot \tanh(\mathbf{c}_t^{(\ell,r)}). \quad (15)$$

Here, $\mathbf{f}_t^{(\ell,r)}$, $\mathbf{i}_t^{(\ell,r)}$ and $\mathbf{o}_t^{(\ell,r)}$ denote the forget, input and output gates, respectively, while $\mathbf{c}_t^{(\ell,r)}$ represents the internal

cell state and $\tilde{\mathbf{c}}_t^{(\ell,r)}$ denotes the candidate cell state. The matrices $W_{\{f,i,c,o\}}^{(\ell,r)}$ and vectors $b_{\{f,i,c,o\}}^{(\ell,r)}$ are the corresponding trainable weights and biases for layer ℓ and direction r . The cell state is a learned latent memory and should not be confused with the physical system state $\mathbf{S}_{t-1}$ defined in the preprocessing module.

2.3.2 Transformer encoder and dimensionality adaptation

The transformer encoder processes the temporal representation produced by the Bi-LSTM through multi-head self-attention. In the proposed architecture, self-attention operates along the T_{obs} observation-window dimension and therefore directly models dependencies between different time steps. Since the feature representation at every time step is obtained from the joint state of all beams, inter-beam information is implicitly embedded in each temporal representation. Thus, the transformer does not perform attention directly over the beam dimension; rather, it models long-range temporal dependencies using representations that already contain joint beam information.

To feed the Bi-LSTM output into the transformer, a linear adaptation layer is introduced. The adaptation layer projects the feature vector at each time step to a dimension d_{model} compatible with the multi-head attention mechanism. Specifically, we set $d_{\text{model}} = n_{\text{heads}} \times \lceil (2d_h)/n_{\text{heads}} \rceil$, where n_{heads} denotes the number of attention heads. This ensures that d_{model} is divisible by n_{heads} , allowing the feature dimension to be evenly divided among the attention heads. The resulting attention-enhanced representation is given by:

$$\mathbf{A}_t = \text{Attention}(\text{Adapt}(\mathbf{U}_t)), \quad (16)$$

where $\text{Adapt}(\cdot)$ denotes the linear projection.

In our architecture, we distinguish two functional roles for the learned mapping: a prediction function f_{θ}^{pred} that outputs the predicted traffic and CSI for the next time slot, and an allocation function $f_{\theta}^{\text{alloc}}$ that outputs the activation probabilities for the beam hopping schedule. Both functions share the same underlying LSTM-transformer backbone but use separate output heads, allowing the model to jointly learn forecasting and decision-making. These two output functions are expressed as

$$(\hat{\mathbf{D}}_{t+1}, \hat{\mathbf{G}}_{t+1}) = f_{\theta}^{\text{pred}}(\mathbf{A}_t), \quad (17)$$

and

$$\mathbf{p}(t) = f_{\theta}^{\text{alloc}}(\mathbf{A}_t). \quad (18)$$

2.3.3 Pure transformer baseline

To isolate the contribution of the recurrent front end, a pure transformer baseline is also implemented, in which the three Bi-LSTM layers are replaced by a positional encoding layer. This baseline uses the same number of encoder blocks ($L = 6$), attention heads ($N_{\text{head}} = 8$), FFN dimension (512), Adaptive Moment (ADAM) estimation optimizer, multi-objective loss, and early stopping, differing only in the absence of recurrent inductive bias.

2.4 Optimization and decision module

During training, the allocation head directly produces continuous beam-activation probabilities, which are used to evaluate the multi-objective loss. Therefore, the proposed framework does not solve a separate optimization problem over the binary matrix $\mathbf{B}$ during inference. Instead, the network parameters are learned by minimizing the multi-objective training loss, while the final binary BH schedule is obtained from the trained activation probabilities using the Top- ϑ operation described in Algorithm 1.

The multi-objective loss is

$$\mathcal{L}_{\text{MO}} = \alpha \mathcal{L}_{\text{SE}} + \beta \mathcal{L}_{\text{F}} + \gamma \mathcal{L}_{\text{QoS}} + \delta \mathcal{L}_{\text{P}}, \quad \alpha + \beta + \gamma + \delta = 1, \quad (19)\text{a}$$

$$\mathcal{L}_{\text{SE}} = \frac{1}{T_{\text{hor}} N} \sum_{\tau, i} (y_{i,\tau} - y_{i,\tau}^{\text{ref}})^2, \quad (19)\text{b}$$

$$\mathcal{L}_{\text{F}} = 1 - \frac{(\sum_i R_i)^2}{N \sum_i R_i^2}, \quad (19)\text{c}$$

$$\mathcal{L}_{\text{QoS}} = \frac{1}{N} \sum_i \max(0, \text{SINR}_{\min} - \text{SINR}_i), \quad (19)\text{d}$$

$$\mathcal{L}_{\text{P}} = \left(\frac{\sum_{\tau, i} P_{i,\tau}}{P_{\max}} - \eta \right)^2. \quad (19)\text{e}$$

Here, $\mathcal{L}_{\text{SE}}$ measures the deviation between the continuous activation output $y_{i,\tau}$ and the demand-guided reference $y_{i,\tau}^{\text{ref}}$. The quantity $y_{i,\tau}^{\text{ref}}$ is obtained by normalizing the traffic demand for the next scheduling step and is used as a surrogate for the desired beam activation. It is not obtained from an exact optimization solver. Therefore, the proposed model should not be interpreted as learning the solution of an exact optimization algorithm; rather, it learns a demand-guided scheduling policy shaped by the overall multi-objective training loss.

The term $\mathcal{L}_{\text{F}}$ is based on Jain's fairness index with R_i the average throughput of beam i , and $\mathcal{L}_{\text{QoS}}$ penalizes QoS violations below the minimum threshold $\text{SINR}_{\min}$. The term $\mathcal{L}_{\text{P}}$ evaluates compliance with the considered power-utilization target under the fixed per-active-beam power model. The transmit power of an active beam

and the bandwidth are fixed; these quantities are not independently optimized by the learning algorithm. The coefficients $\alpha, \beta, \gamma, \delta$ reflect the relative priority of each objective. Training uses the ADAM optimizer with learning rate 10^{-3} , batch size 32, and early stopping with patience equal to 20 epochs.

2.5 Channel model

The channel gain $|g_{i,q}(t)|$ for the user q in the beam i at time t is generated by a Rician fading process as

$$\mathbb{E}[|g_{i,q}|^2] = \frac{10^{G_{\text{sat}}/10} \cdot 10^{G_{\text{usr}}/10}}{(4\pi d_{i,q}/\lambda)^2}, \quad (20)$$

with $G_{\text{sat}} = 52$ dBi, $G_{\text{usr}} = 45$ dBi, $d_{i,q} \in [600, 800]$ km, $\lambda = c/f_c$, and carrier frequency $f_c = 19.5$ GHz in the Ka-band. The inter-beam interference matrix is $I_{j,i,q} = 0.05 \exp(-|i - j|/2)$ for $i \neq j$, where $I_{j,i,q}$ denotes the interference contribution from beam j to user q served by beam i . With the considered simulation parameters, this model results in SINR values mainly within the 8–30 dB range. In addition, the normalized satellite antenna radiation pattern is modelled as

$$\bar{G}_{\text{ant}}(\psi) = \left[\frac{J_1(u)}{2u} + 36 \frac{J_3(u)}{u^3} \right]^2, \quad u = 2.07123 \frac{\sin(\psi)}{\sin(\psi_{3\text{dB}})}, \quad (21)$$

where ψ is the off-axis angle from the beam centre, $\psi_{3\text{dB}}$ denotes the one-sided half-power beamwidth, and $J_1(\cdot)$ and $J_3(\cdot)$ are the first and third-order Bessel functions of the first kind, respectively.

2.6 Beam scheduler and feedback loop

The beam scheduler translates the binary BH schedule $\mathbf{B}_{t+1}$ into executable satellite commands specifying the active beam set and the corresponding dwell times. Since the transmission power and bandwidth are fixed here in the system configuration, they are not dynamically generated by the scheduler. After transmission, the measured outputs are given by

$$\mathbf{Y}_{t+1} = \{\text{SINR}_{t+1}, \text{QoS}_{t+1}, \text{Throughput}_{t+1}\}. \quad (22)$$

These measurements and the executed BH decision are reintegrated into the historical state as

$$\mathbf{S}_{t+1} = \mathcal{U}(\mathbf{S}_t, \mathbf{B}_{t+1}, \mathbf{Y}_{t+1}), \quad (23)$$

where $\mathcal{U}(\cdot)$ denotes the feedback state update function. This update incorporates the executed BH decision together with the measured SINR, QoS and throughput indicators to construct the historical state used at the next decision epoch. The feedback loop adapts subsequent scheduling decisions through the updated input state

Algorithm 1: Generation of the binary BH schedule

Input: Trained hybrid model f_θ , input sequence $\mathbf{Z}_t$, number of beams N , prediction horizon T_{hor} , number of active beams ϑ

Output: Binary BH schedule matrix $\mathbf{B}_{\text{opt}} \in \{0, 1\}^{T_{\text{hor}} \times N}$

```

1  $\mathbf{U}_t \leftarrow \text{BiLSTM}(\mathbf{Z}_t)$ ;
2  $\mathbf{A}_t \leftarrow \text{TransformerEncoder}(\text{Adapt}(\mathbf{U}_t))$ ;
3  $\mathbf{P} \leftarrow f_\theta^{\text{alloc}}(\mathbf{A}_t)$ ;
4 Initialize  $\mathbf{B}_{\text{opt}} \leftarrow \mathbf{0}_{T_{\text{hor}} \times N}$ ;
5 for  $\tau \leftarrow 1$  to  $T_{\text{hor}}$  do
6   Extract probability vector  $\mathbf{p}(\tau)$  from  $\mathbf{P}$ ;
7    $\mathcal{I}(\tau) \leftarrow \text{Top}_\vartheta(\mathbf{p}(\tau), \vartheta)$ ;
8   foreach  $i \in \mathcal{I}(\tau)$  do
9      $B_{\tau,i} \leftarrow 1$ ;
10  end
11 end
12 return  $\mathbf{B}_{\text{opt}}$ 
    
```

without implying that the neural network parameters are retrained after every scheduling slot. By tightly aligning learning, scheduling and feedback with the functional blocks shown in Fig. 3, the proposed model dynamically adapts BH decisions to traffic and channel variations.

2.7 Beam hopping schedule generation and dwell time computation

Algorithm 1 outlines the process for creating the BH scheduling matrix $\mathbf{B}_{\text{opt}}$. This uses the outputs from the Bi-LSTM-transformer hybrid model. At each time step t , the trained model generates a vector of activation probabilities expressed as

$$\mathbf{p}(t) = (p_1(t), p_2(t), \dots, p_N(t)). \quad (24)$$

Here, $p_i(t)$ represents the probability of activating beam i during time slot t .

The maximum number of simultaneously active beams is denoted by ϑ . In the considered simulations, it is set to

$$\vartheta = \left\lfloor \frac{N}{4} \right\rfloor. \quad (25)$$

This parameter represents the simultaneous activation constraint adopted in the simulation setup. For each scheduling interval, the indices associated with the ϑ highest activation probabilities are selected and converted into a binary allocation vector.

In the experiments considered in this work, $T_{\text{hor}} = 1$. Therefore, the dwell time is not obtained from multiple future slots generated in a single model evaluation. Instead, the one-step binary decisions produced at consecutive scheduling epochs are concatenated to form the

Algorithm 2: Extraction of beam dwell times

Input: Binary scheduling matrix $\mathbf{B} \in \{0, 1\}^{T \times N}$

Output: Set of dwell times $\mathcal{D}$

```

1 Initialize  $\mathcal{D} \leftarrow []$ ;
2 for  $i \leftarrow 1$  to  $N$  do
3    $c \leftarrow 0$ ;
4   for  $t \leftarrow 1$  to  $T$  do
5     if  $B_{t,i} = 1$  then
6        $c \leftarrow c + 1$ ;
7     else
8       if  $c > 0$  then
9         Append  $c$  to  $\mathcal{D}$ ;
10         $c \leftarrow 0$ ;
11      end
12    end
13  end
14  if  $c > 0$  then
15    Append  $c$  to  $\mathcal{D}$ ;
16  end
17 end
18 return  $\mathcal{D}$ 
    
```

BH sequence over the simulation period. The dwell time of a beam is then calculated from consecutive active slots in this accumulated sequence.

The dwell time for a beam i during an activation interval can be expressed as

$$T_{\text{dwell}}^{(i)} = n_i \Delta t, \quad (26)$$

where n_i is the number of consecutive scheduling slots in which the beam i remains active and Δt is the duration of one scheduling slot. More generally, the dwell time for beam i during the m -th activation period can be written as

$$T_{\text{dwell}}^{(i,m)} = \left(\sum_{t=t_s}^{t_e} B_{t,i} \right) \Delta t. \quad (27)$$

Here, t_s and t_e are the start and end indices of a consecutive activation interval, while $B_{t,i} \in \{0, 1\}$ shows whether beam i is active at slot t .

The dwell time is therefore directly determined by the number of consecutive active scheduling slots assigned to each beam. Depending on the learned activation probabilities, beams with higher traffic demand or more restrictive QoS conditions may be selected over consecutive scheduling slots and consequently obtain longer dwell intervals. To enable end-to-end training of these beam-activation decisions, the non-differentiable Top_ϑ operation is applied only during inference. During training, the network outputs a continuous vector of beam-activation probabilities $\mathbf{p}(t) \in [0, 1]^N$, which is used to evaluate the multi-objective loss function $\mathcal{L}_{\text{MO}}$, including the SINR and throughput-related terms. This allows gradients to propagate through the network during training.

After training, the Top_s operation converts the continuous activation probabilities into a binary BH schedule according to the adopted simultaneous-beam activation constraint.

2.8 Realistic training data

The model is trained and evaluated using synthetic traffic composed of Poisson voice traffic, on/off video traffic and log-normal data traffic. These models represent heterogeneous traffic characteristics, including random arrivals, intermittent activity, and broadband traffic variability. No real operational traffic traces are used in the current evaluation. Therefore, the reported results characterize the performance of the proposed architecture under the traffic and channel distributions considered in the simulations and should not be interpreted as evidence of generalization to arbitrary operational traffic conditions. Evaluation using real traces and out-of-distribution traffic models is left for future work.

For each Monte Carlo realization, we used a total of 128 time slots, with an observation window of $T_{\text{obs}} = 50$ and a prediction horizon of $T_{\text{hor}} = 1$ slot, yielding 78 input-target sequences. These sequences were split into 80% for training and 20% for validation. No separate test subset was extracted from the same realization. Instead, the complete training and validation procedure was repeated over 30 independently generated Monte Carlo realizations drawn from the same statistical traffic and channel models. Data was normalized using scaling parameters obtained exclusively from the training subset.

A base random seed of 42 was used for reproducibility, with a different realization generated for each Monte Carlo run. The model was retrained from scratch for every run. The resulting confidence intervals therefore characterize performance variability across different realizations of the considered distributions rather than out-of-distribution generalization.

2.9 Scalability to large multibeam satellite systems

The simulations considered in this work mainly focus on a configuration with $N = 16$ beams. To discuss the extension of the proposed hybrid Bi-LSTM-transformer architecture to larger multibeam systems without relying on a simple linear extrapolation of the total training time, we analyze separately the computational components that depend on the observation-window length T_{obs} and the number of beams N .

2.9.1 Analysis of computational complexity

The computational scalability of the proposed hybrid architecture depends mainly on the observation-window length T_{obs} and the number of beams N . For the first Bi-LSTM layer, the input dimension is

$$d_{\text{in}} = 2N + d_s, \quad (28)$$

and its computational cost can be expressed, up to constant factors associated with the LSTM gates and the two directions, as

$$\mathcal{O}\left[T_{\text{obs}}(d_{\text{in}}d_h + d_h^2)\right]. \quad (29)$$

The upper Bi-LSTM layers operate on fixed-dimensional hidden representations and therefore add terms that depend on T_{obs} and d_h but do not directly increase with N when d_h is fixed.

For a transformer encoder block, the self-attention score computation scales as

$$\mathcal{O}(T_{\text{obs}}^2 d_{\text{model}}), \quad (30)$$

while the linear projections and feed-forward operations introduce additional terms of order

$$\mathcal{O}(T_{\text{obs}} d_{\text{model}}^2). \quad (31)$$

The final allocation head produces one activation output per beam, resulting in an additional beam-dependent term of order

$$\mathcal{O}(Nd_{\text{model}}). \quad (32)$$

For a fixed number of transformer blocks and fixed internal dimensions, the main sequence and beam-dependent components of the hybrid architecture can therefore be summarized as

$$C_{\text{Hybrid}} = \mathcal{O}\left(T_{\text{obs}}Nd_h + T_{\text{obs}}d_h^2 + T_{\text{obs}}^2d_{\text{model}} + T_{\text{obs}}d_{\text{model}}^2 + Nd_{\text{model}}\right). \quad (33)$$

Consequently, when T_{obs} , d_h , and d_{model} are fixed, the beam-dependent part increases approximately linearly with N . In contrast, when N is fixed, the self-attention component grows quadratically with the observation-window length T_{obs} .

With the baseline configuration of $N = 16$ beams, the measured average training time is approximately 30 s on an NVIDIA V100 GPU. Considering only the terms that depend linearly on N , increasing the number of beams from 16 to 128 corresponds to an eight-fold increase in those beam-dependent operations. This factor should not be interpreted as an eight-fold prediction of the total

training time, since the complete runtime also depends on the sequence-processing operations, transformer computations, mini-batch size, number of optimization steps, software implementation, and hardware utilization.

Because d_{model} is kept fixed after the Bi-LSTM adaptation layer, increasing N does not directly increase the feature dimension processed by the Transformer. The main direct dependence on N therefore appears in the input processing of the first Bi-LSTM layer and in the final beam-allocation head.

Inference is expected to require less computation than training because no gradient calculation or parameter update is performed. Once the model is trained, each scheduling decision requires only a forward pass through the Bi-LSTM, adaptation layer, transformer encoder and allocation head. The architecture is therefore potentially suitable for time-constrained scheduling applications; however, compatibility with a specific operational scheduling interval should be established using dedicated inference time measurements on the target hardware.

3. SIMULATION SETUP AND RESULTS

In this section, we present the experimental setup used to evaluate the proposed model. The simulations are configured according to the parameters listed in tables 3 and 4. For consistency with the DBO benchmark and to ensure a fair comparison, these parameters follow the Ka-band scenario in [1], with 16 beams, $f_c = 19.5$ GHz, a bandwidth of 500 MHz per beam, $P_{\text{tx}} = 20$ dBW, and a noise power of -120.64 dBW. The synthetic traffic combines Poisson voice traffic, on/off video traffic and log-normal data traffic to represent heterogeneous characteristics such as random arrivals, burstiness and broadband traffic variability. The evaluation is performed over $N_{\text{MC}} = 30$ Monte Carlo realizations. For each realization, the channel matrix $\mathbf{G}$ and traffic sequence $\mathbf{D}$ are regenerated using different random seeds. The reported confidence

Table 3 – Main system and simulation parameters

Parameter	Value
Number of beams N	16
BH cycle duration T_H	256 ms
Slot duration T_{slot}	1 ms
Carrier frequency	19.5 GHz (Ka-band)
Bandwidth per beam	500 MHz
Transmit power per active beam	20 dBW
Noise power	-120.64 dBW
Satellite antenna gain	52 dBi
User terminal gain	45 dBi
Rician K -factor	10 dB
Propagation distance	600–800 km
SINR QoS threshold	3 dB

Table 4 – Hybrid model and training configuration

Parameter	Value
Observation window T_{obs}	50 slots
Prediction horizon T_{hor}	1 slot
Bi-LSTM hidden units	128 per direction
Transformer attention heads	8
Transformer encoder blocks	6
FFN hidden dimension	512
Dropout	0.1
Learning rate	10^{-3}
Batch size	32
Maximum epochs	50
Early stopping patience	20
Total time slots per MC run	128
Training/validation split	80% / 20%
Normalization	Training set maximum
Base random seed	42
Monte Carlo runs	30
Model initialization	Retrained for each MC run
Default $(\alpha, \beta, \gamma, \delta)$	$(0.4, 0.3, 0.2, 0.1)$

intervals therefore account for the stochastic variations of both the propagation channel and traffic process.

To validate the benefit of using three layers, we performed an ablation study where we varied the number of Bi-LSTM layers (1, 2, and 3). As shown in Table 5, the SE gradually improves as the number of layers increases, with the three-layer configuration achieving the highest mean value. The gain from two to three layers is modest but consistent.

Table 6 reports the KPIs for the three models over $N_{\text{MC}} = 30$ runs. Each entry gives the sample mean $\pm$ the 95% half-width of the t -confidence interval $t_{0.975,29} \cdot \hat{\sigma} / \sqrt{30}$.

The non-overlapping confidence intervals on SE confirm that all three pairwise differences (DBO vs pure transformer, DBO vs hybrid and pure transformer vs hybrid) are statistically significant at the 5% level. Specifically, the proposed hybrid achieves:

A 44.1% improvement in average SE over DBO ($4.13 \rightarrow 5.95$ b/s/Hz), well outside the respective CIs; a 16.9% improvement over the pure transformer baseline ($5.09 \rightarrow 5.95$ b/s/Hz), confirming the value of the recurrent front-end;

A 32.9% reduction in training time compared with the pure transformer ($45.6 \rightarrow 30.6$ s), because the Bi-LSTM layers extract informative temporal features before the attention stage, reducing the learning burden of the transformer and accelerating convergence;

The lowest power utilization ($4.3 \pm 0.3\%$), closest to the design target $\eta = 4.3\%$, reflecting accurate power-budget regulation by the δ weighted loss term.

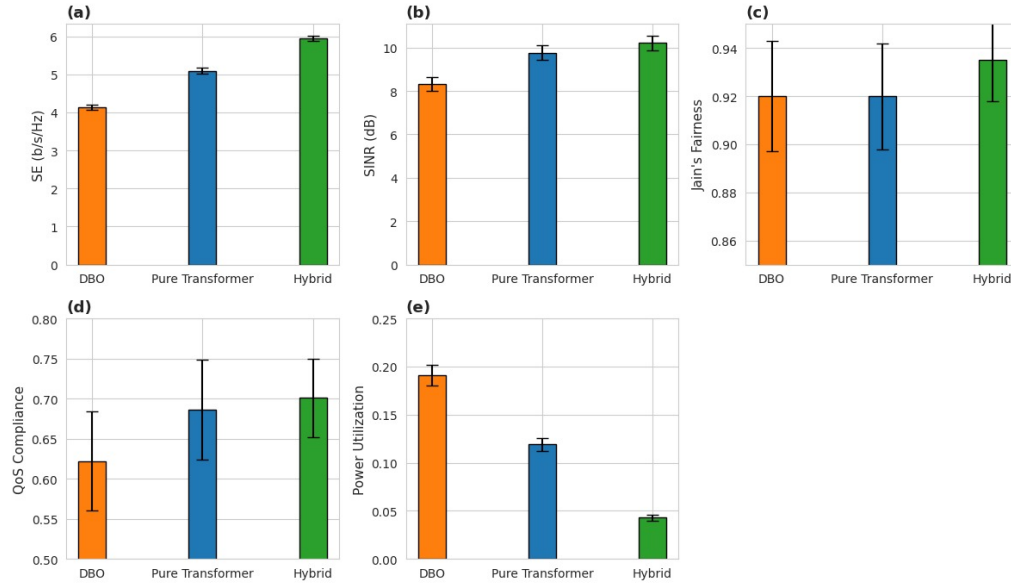


Figure 4 – KPI comparison with 95% CIs (error bars). (a) Spectral efficiency, (b) average SINR, (c) Jain's fairness, (d) QoS compliance, (e) power utilization. The hybrid model (green) outperforms both baselines on all metrics.

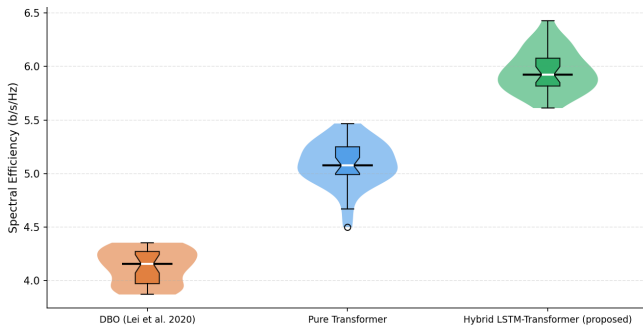


Figure 5 – Distribution of SE over 30 Monte Carlo runs (violin + notched box plot; notch gives 95% CI on the median). The non-overlapping notches confirm statistically significant differences.

Table 5 – Ablation study: SE versus the number of Bi-LSTM layers. Results are averaged over 30 Monte Carlo runs.

Number of Bi-LSTM layers	SE (b/s/Hz)
1	5.68 ± 0.09
2	5.82 ± 0.08
3 (proposed)	5.95 ± 0.07

Table 6 – Monte Carlo KPI comparison – mean $\pm$ 95% CI ($N = 30$ runs)

KPI	DBO [1]	Pure Transformer	Hybrid (proposed)
SE (b/s/Hz)	4.13 ± 0.06	5.09 ± 0.08	5.95 ± 0.07
SINR (dB)	8.33 ± 0.31	9.77 ± 0.33	10.22 ± 0.34
Jain Fairness	0.920 ± 0.023	0.920 ± 0.022	0.935 ± 0.017
QoS Compliance	0.622 ± 0.062	0.686 ± 0.062	0.701 ± 0.049
Power Utilization	0.191 ± 0.011	0.119 ± 0.007	0.043 ± 0.003
Training time (s)	0.08 ± 0.003	45.6 ± 1.72	30.6 ± 1.75

Fig. 4 compares the five main performance metrics obtained over the 30 Monte Carlo realizations, while Fig. 5

shows the distribution of the achieved Spectral Efficiency (SE). The error bars in Fig. 4 represent the 95% confidence intervals.

The proposed hybrid model achieves the highest SE, with approximately 5.95 b/s/Hz, compared with 5.09 b/s/Hz for the pure transformer and 4.13 b/s/Hz for DBO. This corresponds to an improvement of approximately 16.9% over the pure transformer and 44.1% over DBO. The hybrid model also achieves higher average SINR and QoS compliance while maintaining a Jain's fairness index of approximately 0.935.

The SE distributions in Fig. 5 further show a clear separation between the three considered methods over the different Monte Carlo realizations. The proposed hybrid model consistently provides the highest SE values, followed by the pure Transformer and DBO.

The pure transformer already provides a strong baseline, outperforming DBO by 23.3% in SE. The additional performance gain obtained by the hybrid model can be attributed in part to the Bi-LSTM front end, which extracts informative temporal features before the attention mechanism. This reduces the learning burden on the transformer without reducing the sequence length T_{obs} . The narrower CI on fairness (± 0.017 vs. ± 0.022) also indicates lower performance variability across the considered stochastic realizations.

Fig. 6 shows the training and validation loss curves. The DBO model converges within a few epochs due to its relatively simple FC-DNN structure. The pure transformer exhibits slower initial convergence, while the hybrid model converges faster and reaches a lower final validation loss. This indicates that the Bi-LSTM front-end provides informative temporal features before the attention mechanism and facilitates the learning process.

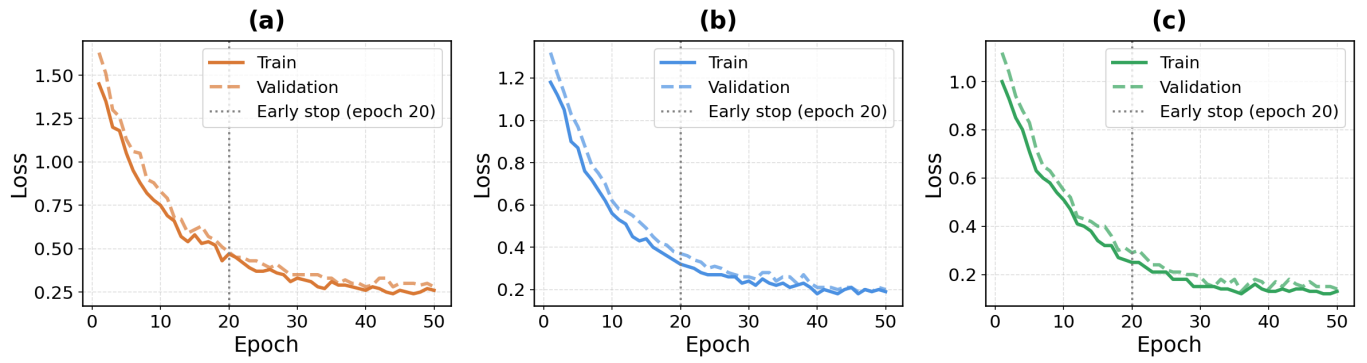


Figure 6 – Training and validation loss curves for (a) DBO, (b) the pure transformer and (c) the proposed hybrid LSTM-transformer model.

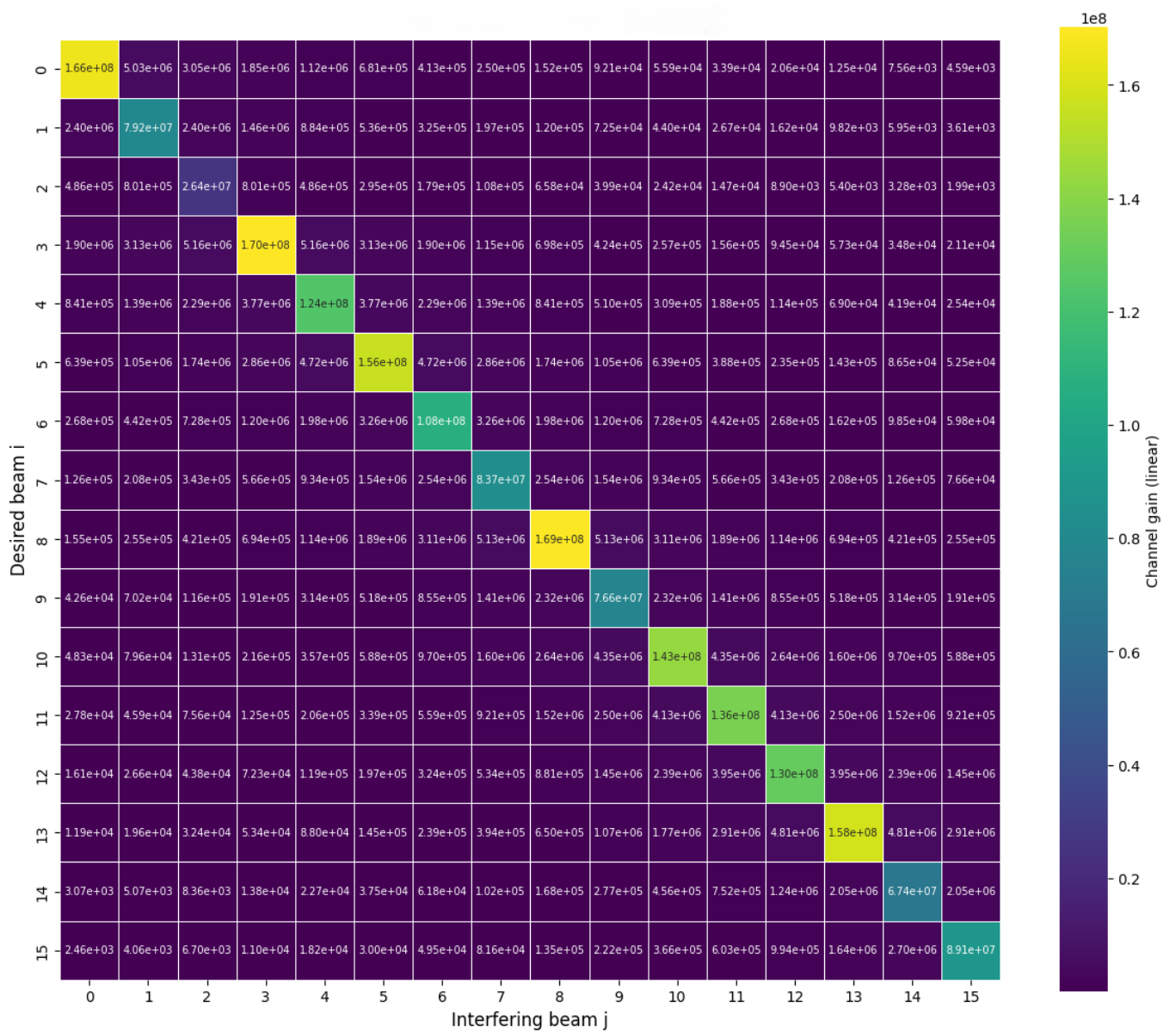


Figure 7 – Simulated inter-beam channel gain matrix G on a linear scale. Diagonal elements correspond to the desired links, while off-diagonal elements represent inter-beam interference.

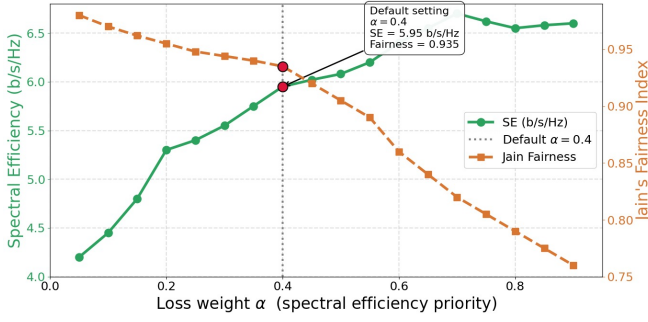


Figure 8 – SE–fairness trade-off as a function of the SE loss weight α . The vertical dotted line indicates the default value $\alpha = 0.4$.

Fig. 7 illustrates the simulated inter-beam channel gain matrix $\mathbf{G}$. The dominant diagonal elements, on the order of 10^8 , correspond to the desired beam links, whereas the off-diagonal elements represent inter-beam interference. The interference decreases with beam separation, from approximately 10^5 – 10^6 for nearby beams to 10^3 – 10^4 for more distant beams, following the adopted exponential interference model $I_{ji} = 0.05 \exp(-|i - j|/2)$. The variability of the diagonal elements results from differences in propagation distance and Rician fading across the considered links. Overall, the matrix provides heterogeneous channel and interference conditions for evaluating the BH scheduling methods. The structure of $\mathbf{G}$ also influences the BH scheduling decisions. Simultaneous activation of beams with strong mutual interference can reduce the resulting SINR, whereas beams with lower mutual interference can be activated concurrently with less performance degradation. Since the input representation contains the joint channel information of all beams, the proposed model can account for these heterogeneous interference conditions when generating the BH schedule.

Fig. 8 shows the effect of the SE weight α on the achieved SE and Jain’s fairness index while β , γ , and δ are kept fixed. As α increases, the SE increases while the fairness index decreases, showing the expected trade-off between SE and fairness. At the default value $\alpha = 0.4$, the proposed model achieves approximately 5.95 b/s/Hz with a fairness index of 0.935. To isolate the effect of each parameter, α and the target power-utilization parameter η are varied separately while the remaining parameters are kept fixed. This trade-off provides flexibility in selecting α according to the desired balance between SE and fairness.

Table 7 shows the effect of α and η on the achieved SE. For a fixed α , increasing η generally improves the SE, while increasing α at a fixed η also favours higher SE by assigning more importance to the SE objective. The best hybrid performance is obtained for $\alpha = 0.6$ and $\eta = 0.06$, reaching 6.70 b/s/Hz. Across all tested configurations, the proposed hybrid model consistently outperforms both the pure transformer and DBO baselines.

Table 7 – Mean SE (b/s/Hz) for each configuration – comparison of DBO, Pure Transformer (PT) and hybrid model (proposed)

α	η	DBO	PT	Hybrid	Gain vs PT	Gain vs DBO
0.2	0.03	3.85	4.72	5.30	+12.3%	+37.7%
0.2	0.05	4.02	4.90	5.48	+11.8%	+36.3%
0.4	0.03	4.13	5.09	5.95	+16.9%	+44.1%
0.4	0.05	4.30	5.30	6.10	+15.1%	+41.9%
0.6	0.04	4.50	5.80	6.40	+10.3%	+42.2%
0.6	0.06	4.22	5.45	6.70	+22.9%	+58.8%
0.8	0.05	4.60	5.95	6.55	+10.1%	+42.4%
0.8	0.07	4.75	6.10	6.60	+8.2%	+38.9%

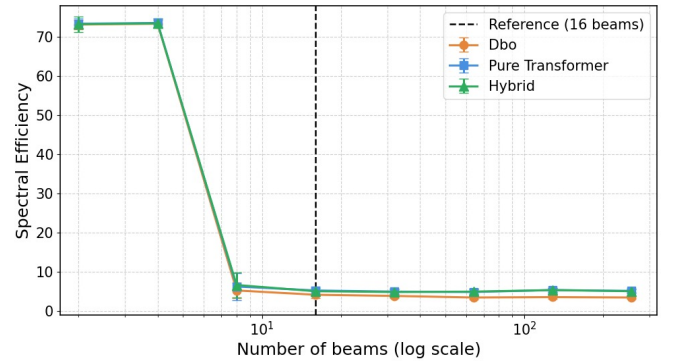


Figure 9 – Analysis of the scalability: SE versus the number of beams for DBO, the pure Transformer, and the proposed hybrid model.

3.1 Scalability analysis

Fig. 9 evaluates the scalability of the proposed hybrid Bi-LSTM-transformer architecture as the number of satellite beams increases. The SE achieved by the proposed method is compared against the DBO baseline and the pure transformer model for different multibeam network sizes.

For small configurations with 2–4 beams, the three methods achieve similar SE values. A significant decrease in SE is observed when the number of beams increases, indicating that the scheduling problem becomes more challenging for larger multibeam configurations. Nevertheless, the proposed hybrid model maintains a higher SE than the DBO and pure transformer baselines across the considered network sizes.

Overall, these results demonstrate that the proposed Hybrid Bi-LSTM-transformer model is not only more spectrally efficient but also highly scalable, making it suitable for future large-scale LEO and EHTS satellite systems.

4. PRACTICAL APPLICABILITY

4.1 Reality of simulation parameters

The simulation parameters are aligned with operational Ka-band HTS systems. The carrier frequency of 19.5 GHz and noise power of -120.64 dBW match the ESA71 simulator parameters used in [1]. The satellite antenna gain of 52 dBi is consistent with large aperture reflector antennas ($diameter \geq 2$ m) used in LEO HTS payloads. The propagation distances of 600–800 km produce path losses representative of what a well-engineered LEO link budget achieves after including high-gain user terminals ($G_{usr} = 45$ dBi), thereby yielding SINR values in the 8–30 dB range consistent with digital video broadcasting satellite-second generation extensions (DVB-S2X) operating points. The Rician K -factor of 10 dB is appropriate for clear-sky line-of-sight satellite links. The three traffic classes (Poisson voice, on/off video, log-normal data) are standard models in satellite traffic engineering and have been validated against real operator datasets in the literature.

4.2 Implementation ability in a real satellite NCC

The proposed framework is intended for ground-based implementation at the NCC, where the model is trained offline and then used for online BH scheduling. At each decision interval, the NCC uses the most recent traffic demand measurements and CSI estimates to generate the beam-activation pattern and dwell times, which are then communicated to the satellite through the feeder link.

In practice, channel estimation overhead, feedback delay, and processing time may affect the freshness of the CSI and consequently the scheduling decision. These latency effects are not explicitly modelled in the current work. For LEO systems, periodic retraining or online fine-tuning may also be considered to account for longer-term changes in traffic and channel conditions.

5. CONCLUSION

This paper proposed a hybrid Bi-LSTM-transformer model for BH optimization in multibeam satellite communication systems. The architecture combines a three-layer stacked Bi-LSTM front end for hierarchical temporal feature extraction with a transformer encoder that captures long-range dependencies across the observation window using a joint representation of all beams. Training is governed by a four-term multi-objective loss that balances SE, Jain's fairness, QoS compliance and power-utilization objectives.

Monte Carlo simulations in a Ka-band DVB-S2X scenario with $N = 16$ beams showed that the proposed model improves SE compared with the DBO benchmark [1] and a pure transformer baseline, while reducing the training time compared with the pure transformer. The results also show stable performance across different stochastic traffic and channel realizations generated from the considered distributions. However, validation using real operational data or different traffic and channel distributions is still required to assess generalization.

Future work will consider online fine-tuning for time-varying LEO environments, inference time optimization for onboard deployment and joint BH and precoding design.

ACKNOWLEDGEMENT

This work is supported by MDA Space, CRIAQ, MITACS, and NSERC.

REFERENCES

- [1] Lei Lei, Eva Lagunas, Yaxiong Yuan, Mirza Golam Kibria, Symeon Chatzinotas, and Björn Ottersten. "Deep learning for beam hopping in multibeam satellite systems". In: *IEEE Vehicular Technology Conference*. 2020, pp. 1–5.
- [2] Olfa Ben Yahia, Zineb Garroussi, Olivier Bélanger, Brunilde Sansò, Jean-François Frigon, Stéphane Martel, Antoine Lesage-Landry, and Gunes Karabulut Kurt. "Evolution of high-throughput satellite systems: A vision of programmable regenerative payload". In: *IEEE Communications Surveys & Tutorials* 27.3 (2025), pp. 1565–1597.
- [3] Lin Chen, Linlong Wu, Eva Lagunas, Anyue Wang, Lei Lei, Symeon Chatzinotas, and Björn Ottersten. "Joint power allocation and beam scheduling in beam-hopping satellites: A two-stage framework with a probabilistic perspective". In: *IEEE Transactions on Wireless Communications* 23.10 (2024), pp. 14685–14701.
- [4] Zhenyu Xiao, Junyi Yang, Tianqi Mao, Chong Xu, Rui Zhang, Zhu Han, and Xiang-Gen Xia. "LEO satellite access network (LEO-SAN) toward 6G: Challenges and approaches". In: *IEEE Wireless Communications* 31.2 (2024), pp. 89–96.
- [5] Olfa Ben Yahia, Eylem Erdogan, Gunes Karabulut Kurt, Ibrahim Altunbas, and Halim Yanikomeroglu. "HAPS selection for hybrid RF/FSO satellite networks". In: *IEEE Transactions on Aerospace and Electronic Systems* 58.4 (2022), pp. 2855–2867.
- [6] Zhihao Han, Tao Yang, and Rongke Liu. "On beam hopping pattern design for satellite communication systems with hybrid precoding". In: *IEEE Transactions on Vehicular Technology* 73.1 (2024), pp. 1364–1369.
- [7] Mengying Zhang, Xiumei Yang, and Zhiyong Bu. "Resource allocation for beam-hopping-based satellite systems with spectrum sharing". In: *IEEE Access* 12 (2024), pp. 101592–101602.
- [8] Wei Li, Luliang Jia, Yingwu Chen, Quan Chen, Jungang Yan, and Nan Qi. "A game-theoretic approach for satellites beam scheduling and power control in a mega hybrid constellation spectrum sharing scenario". In: *IEEE Internet of Things Journal* 12.12 (2025), pp. 20626–20639.

- [9] Yang Cao, Shao-Yu Lien, Ying-Chang Liang, Dusit Niyato, and Xuemin Sherman Shen. "Collaborative deep reinforcement learning for resource optimization in non-terrestrial networks". In: *Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. 2023, pp. 1–7.
- [10] Zhiguo Liu, Weijie Li, Jianxin Feng, and Jiaojiao Zhang. "Research on satellite network traffic prediction based on improved GRU neural network". In: *Sensors* 22 (2022).
- [11] Bin Han and Bill Howe. "GRU-SPA: Gated recurrent unit with spatial attention for spatio-temporal disaggregation". In: *arXiv* (2024).
- [12] Penghui Duan, Boan Zou, Yuting Hu, Gaojian Liu, Xin Zhang, and Yifeng He. "Optimization of adaptive frequency hopping strategies for satellite communications applying LSTM neural networks". In: *IEEE International Conference on Electrical, Automation and Computer Engineering (ICEACE)*. 2024, pp. 42–47.
- [13] Weiqing Zhuang and Yongbo Cao. "Short-term traffic flow prediction based on CNN-BiLSTM with multicomponent information". In: *Applied Sciences* 12 (2022).
- [14] Zheng Zhao, Weihai Chen, Xingming Wu, Peter C. Y. Chen, and Jingmeng Liu. "LSTM network: A deep learning approach for short-term traffic forecast". In: *IET Intelligent Transport Systems* 11 (2017), pp. 68–75.
- [15] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural Computation* 9.8 (1997), pp. 1735–1780.
- [16] Yuqi Si and Kirk Roberts. "Three-level hierarchical transformer networks for long-sequence and multiple clinical documents classification". In: *arXiv preprint arXiv:2104.08444* (2021).
- [17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need". In: *Advances in Neural Information Processing Systems* 30 (2017).
- [18] X. Wang, Y. Liu, and Z. Ding. "Transformer-based resource allocation for wireless communications". In: *IEEE Wireless Communications Letters* 11.9 (2022), pp. 1900–1904.
- [19] Yekta Demirci, Guillaume Mantelet, Stéphane Martel, Jean-François Frigon, and Gunes Karabulut Kurt. "Forecasting self-similar user traffic demand using transformers in LEO satellite networks". In: *IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)*. 2025, pp. 1–6.
- [20] Ke-Xin Li, Xiqi Gao, and Xiang-Gen Xia. "Channel estimation for LEO satellite massive MIMO OFDM communications". In: *IEEE Transactions on Wireless Communications* 22.11 (2023), pp. 7537–7550. doi: [10.1109/TWC.2023.3252895](https://doi.org/10.1109/TWC.2023.3252895).
- [21] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural Computation* 9.8 (1997), pp. 1735–1780.
- [22] M. Schuster and K.K. Paliwal. "Bidirectional recurrent neural networks". In: *IEEE Transactions on Signal Processing* 45.11 (1997), pp. 2673–2681.

AUTHORS



ISSIAKOU WOROU WOUIGNANSO is a Ph.D. student in Electrical Engineering at Polytechnique Montréal, Canada, where he conducts research under the supervision of Prof. Brunilde Sansò and Prof. Gunes Karabulut-Kurt. He received a Bachelor's degree in Mathematics from the University of Abomey-Calavi, Benin, in 2014, a Master's degree in Computer Science and Information Systems from the Institut de Mathématiques et de Sciences Physiques (IMSP), Benin, in 2020, and a Master's degree in Mathematical Sciences from the African Institute for Mathematical Sciences (AIMS), Senegal, in 2022. His research interests include satellite communications, beam hopping, satellite operations, resource allocation, mathematical optimization, simulation, and machine learning for next-generation satellite networks.



OLFA BEN YAHIA (IEEE Member) received her engineering degree in 2016 and her Ph.D. in telecommunications engineering in 2022. She is currently a Postdoctoral Fellow at Polytechnique Montréal, where her research focuses on resource allocation and optimization for high-throughput satellite systems. Her research interests include satellite communications, optical wireless communications, resource optimization, and physical layer security.



STÉPHANE MARTEL earned his bachelor's degree in electrical engineering from McGill University in 2001. After two decades in broadcast TV software and management, he joined MDA Satellite Systems in 2021 as a Product Manager, specializing in network communications and embedded software for space technologies.



GUILLAUME MANTELET received the equivalent to M.Sc. in telecommunication engineering from ENST Bretagne (now IMT Atlantique) in 2007, and Ph.D in electrical engineering from the ETS, Canada, in 2012. His research interest are next-generation network communica-

tions. Since 2023 he has been a Senior Software Developer at Satellite Systems, MDA, where he contributes to the development of high-performance software for space-based communications.



GUNES KARABULUT KURT (Senior Member, IEEE) received the B.S. degree with high honors in electronics and electrical engineering from Bogazici University, Istanbul, Türkiye, in 2000, and the M.A.Sc. and Ph.D. degrees in electrical engineering from the University of Ottawa, Canada, in

2002 and 2006, respectively. She is currently the Canada Research Chair (Tier 1) at New Frontiers in Space Communications and a Professor at Polytechnique Montréal, QC, Canada. She is the Director of the Poly-Grames Research Center and the Co-Founder and Scientific Director of ASTROLITH, Transdisciplinary Research Unit of Space Resource and Infrastructure Engineering at Polytechnique Montréal. She is also an Adjunct Research Professor at Carleton University, Canada. Prior to her current position, she was a Professor at Istanbul Technical University from 2010 to 2021. She is a Marie Curie Fellow and a Fellow of the Canadian Academy of Engineering.



BRUNILDE SANSÒ is a Full Professor in the Department of Electrical Engineering at Polytechnique Montréal, Canada, and Director of the Laboratory for the Optimization of Communication Networks (LORLAB). For more than three decades, her research has bridged telecommunications and

operations research, with contributions to the design, performance, optimization, sustainability, and resilience of communication networks. Her work addresses these fundamental problems across a broad range of communication infrastructures, including wireless, IP, and optical networks, as well as satellite and non-terrestrial systems, IoT, 5G/6G, and, more recently, digital infrastructures for AI and data-center networks. She has contributed to national and international scientific and engineering organizations and led numerous academic-industry research collaborations, while training and supervising several generations of researchers in communication network design and optimization.