

Deep Learning-Driven Optimal Beam Prediction for Drone Connectivity via Flying-RIS and Base Station

Muhammad Abul Hassan

*Department of Information Engineering and
Computer Engineering
University of Trento, Italy
Email: muhammadabul.hassan@unitn.it*

Fabrizio Granelli

*Department of Information Engineering and
Computer Engineering
University of Trento, Italy
Email: fabrizio.granelli@unitn.it*

Abstract—A deep learning model is presented for the cutting-edge wireless communication framework to ensure efficient channel estimation for a drone operating at sub-terahertz frequencies by leveraging Flying-Reconfigurable Intelligent Surfaces (Flying-RIS) in conjunction with a ground Base Station. Terahertz (THz) frequencies are envisioned as key enablers for different 6G applications such as ultra-broadband, but this technology still has its own limitations, for instance, high path loss and limited signal wavelength. Channel estimation is an essential component at such high frequencies to enhance overall system performance. To address this problem, we have employed multi-layer Long Short-Term Memory (LSTM) to handle temporal dependencies and OFDM channel complexities. The performance of our proposed approach is tested on different evaluation parameters in comparison with deep learning models including gated recurrent unit (GRU), convolutional neural networks (CNN), and fully connected networks (FCN). The findings for our proposed multi-layer LSTM are better than those of all other deep learning models across all tested parameters, proving its effectiveness in dynamic wireless communications operating at high frequencies.

Index Terms—Deep Learning (DL), Orthogonal Frequency Division Multiplexing (OFDM), DeepMIMO, Terahertz (THz), Reconfigurable Intelligent Surfaces (RIS), Wireless Communication

1. Introduction

With the arrival of 6G, which is still in the testing phase [1], many advanced devices—such as smart wearables, autonomous transportation [2], [3]—and applications like augmented reality and digital twins [4], [5] are expected to become seamlessly connected, a need not fully met by previous generations [6]. To achieve this, we must reshape existing wireless designs and move to higher frequencies, which can deliver high data rates to many devices and applications simultaneously.

Terahertz (THz) frequencies replace the conventional frequency bands (sub-6 GHz). The THz band ranges from 0.1 to 10 THz [7]. Despite their potential for robust data rates, THz signals suffer from severe path loss and environmental attenuation. To mitigate these issues, especially in

drone applications, reconfigurable intelligent surfaces (RIS) [8] mounted on flying drones are a suitable option to intelligently reflect wireless signals to the desired geographical locations. Moreover, multipath-induced inter-symbol interference is yet another major issue in THz, which further degrades overall communication. In our case, the Orthogonal Frequency-Division Multiplexing (OFDM) technique is utilized to split the channel into subcarriers. The effectiveness of these high frequencies critically depends on accurate channel estimation.

Over the last decade, Deep Learning (DL) models have been deployed for wireless channel estimation [9]. Following that, many authors have applied DL to extract spatial properties of wireless signals to enhance overall performance—for instance, Direction-of-Arrival (DoA) [10], Direction-of-Departure (DoD) [11], Angle-of-Arrival (AoA) [12], and path loss [13]. However, these studies have focused only on few spatial feature among the many spatial properties of wireless signals, which in turn limits the exploration of all key properties and their possible effects on channel estimation and overall performance. To explore the full range of spatial features, the authors in [14] applied a gated recurrent unit (GRU) model. However, GRUs suffer from vanishing gradients and reduced effectiveness with long sequences, making them less suitable for dynamic sub-terahertz drone communications.

Motivated by these challenges, we explore alternative solutions for a dynamic drone scenario. Our contributions are as follows:

- We propose a joint deployment of ground base stations and Flying-RIS to establish reliable communication links for drones in the sub-terahertz band, regardless of whether they are in line-of-sight (LoS), non-line-of-sight (NLoS), or fully blocked conditions.
- We apply a multi-layer long short-term memory (LSTM) network to all spatial properties of wireless signals to proactively predict the most suitable channel for the drone. This approach also addresses the vanishing-gradient and long-sequence challenges inherent to traditional recurrent models.

2. Wireless Channels and System Modeling

We have adopted an outdoor communication system operating at a sub-THz frequency (0.2 THz) to serve the mobile drone, as shown in Figure 1. Both the Base Station (BS) and the Flying-RIS are equipped with a single antenna, forming beamforming vectors $\mathbf{f} \in C^{M \times 1}$ and $\mathbf{f} \in C^{N \times 1}$, respectively. These vectors are selected from a predefined beam codebook to generate different channels, either directly or through the Flying-RIS. In this context, the received subcarrier signal is expressed as:

$$y_r = \mathbf{h}_{R,k}^T \Psi \mathbf{h}_{T,k} s + v \quad (1a)$$

$$= (\mathbf{h}_{R,k}^T \odot \mathbf{h}_{T,k})^T \psi s + v \quad (1b)$$

$$\text{s.t. } \Psi = \text{diag}(\Psi); \Psi \in C^{I \times I} \quad (1c)$$

$$|s|^2 \leq P_{\max} \quad (1d)$$

The uplink and downlink channels between the transmitter and receiver are expressed in equations (1a) and (1b), where $\mathbf{h}_{T,k}$ and $\mathbf{h}_{R,k}$ represent the respective channel vectors, and v denotes the additive white Gaussian noise (AWGN). Additionally, the Flying-RIS's internal circuit model is completely passive, with no external power supply, acting solely as a phase shift component ($[\psi]_i = e^{j\psi_i}$), as expressed in equation (1c). To make the system more realistic, a power constraint is applied to the data symbol (s) such that its power does not exceed the total power flux (P_{\max}). Furthermore, to provide a realistic representation of high-frequency, wideband geometric THz channel model is adopted [15]. These channels are further divided into clusters $c, c = \{1, 2, \dots, C\}$ of multipath components (MPCs), each with different characteristics, such as time delay, angles of arrival (AoA), both azimuth (ϕ_c) and elevation (θ_c), and complex path gains (attenuation, phase shift). We assume that one path contributes to the generation of different channels. To calculate the overall channel status between the transmitter and receiver, equation 2 is utilized. After clipping equation 2 into equation 3, we can calculate the frequency domain channel vector at subcarrier k . Additionally, we assume that the block-fading channel model remains unchanged during the coherence time of the channels.

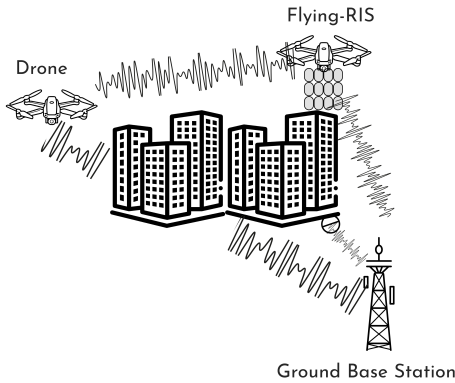


Figure 1. Outdoor Scenario with Base Station and Flying-RIS and Drone

$$\mathbf{h}_d = \sum_{c=1}^C \rho \alpha_c p_{rc} (dT_s - \tau_c) a_{rv}(\theta_c, \phi_c) \quad (2)$$

$$\mathbf{h}_k = \sum_{d=0}^{D-1} \mathbf{h}_d e^{-j \frac{2\pi k}{K} d} \quad (3)$$

3. Data Generation

Deep learning models require data for training, and for this purpose, we utilized the DeepMIMO [16] module, specifically '01 Drone' Scenario [17]. The channels in this scenario are designed using a ray-tracing simulator called Remcom Wireless InSite [18], which is widely adopted by wireless communication professionals in both industry and academia [19], [20] due to its accurate and comprehensive representation of real-world material properties and geometries. For example, it accounts for smooth or rough surfaces (shapes) along with their respective scattering properties. Additionally, the 3D positions of both the transmitter(s) and receiver(s) are crucial for any wireless design engineer/DL/THz-applications. The overall drone communication channels generation scenario are illustrated in Figure 2, where the '01 drone' ray-tracing scenario, along with its ray-tracing parameters, is input into the DeepMIMO dataset generation module. Relevant channel parameters, as shown in Table 1, are configured, and different OFDM channels are subsequently generated (dataset).

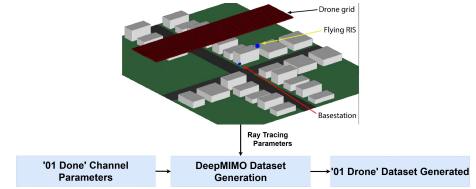


Figure 2. Dataset Generation Process using DeepMIMO module [21]

TABLE 1. COMPARISON OF PARAMETERS FOR BASE STATION AND FLYING-RIS

Parameter	Base Station	Flying-RIS
Centre Frequency	200 GHz	200 GHz
Total OFDM Channels	512	512
OFDM Sampling Factor	1	1
OFDM Limit	1	1
Total Number of Paths	1	1
Location (3D-plane)	(237.504, 580, 6)	(237.504, 650, 80)
Height	6m	80m
Antenna Rotation (3D-plane)	(64, 1, 1)	(256, 1, 1)
Antenna Radiation Pattern	Isotropic	Isotropic
Antenna Spacing	0.5	0.5
Doppler Effect	True	True
BS-Flying RIS Communication Link	True	True

4. Problem Formulation

So far, we have discussed the generation of channels between the drone and different serving stations, either a Flying-RIS or a Base Station. The communication links are updated at each time instant depending on the geographical position of the drone, which could be in Line of Sight (LoS), Non-Line of Sight (NLoS), or in complete blockage

from both the Base Station and Flying-RIS perspectives. To translate the above problem into a deep learning framework, we design the input mapping to its target at a given time instant, as shown in Equation 4.

$$\{Pos^t, CH^t\} \rightarrow \{Pos^{t+1}, CH^{t+1}\} \quad (4)$$

where: Pos^t Drone Position,

CH^t Base Station and Flying-RIS Beam Sequences

5. Deep Learning based Channel Prediction

In this section, we present our proposed deep learning approach, named Multi-Layer Long Short-Term Memory (LSTM) [22], [23], [24]. The internal structure of the multi-layer LSTM is designed to handle input streams (sequences of data) that map previous states to future ones over a period of time, enabling the network to predict future sequences or data. To achieve this, different gates are incorporated, each with a specific functionality. For instance:

(a) Forget gate: This gate classifies streams of data as relevant or irrelevant, operating in conjunction with the candidate cell based on equation 5. The sigmoid activation function is applied to the collective streams of different states (input, hidden, and bias) to scale the values, which are then multiplied (equation 6) with the candidate cell. This operation determines how much data should be retained in the long-term memory (candidate cell).

$$f_t = \sigma(U_f x_t + V_f h_{t-1} + b_f) \quad (5)$$

$$C'_t = f_t \cdot C_{t-1} \quad (6)$$

(b) Store and Update Gate: In this gate, both the input and hidden data streams are combined and utilized alongside the candidate and partial candidate, as shown in equations 7, 8, and 9, also demonstrated in figure 3.

$$i_t = \sigma(U_i x_t + V_i h_{t-1} + b_i) \quad (7)$$

$$C_t^+ = \tanh(U_c x_t + V_c h_{t-1} + b_c) \quad (8)$$

$$C_t = C'_t + i_t \cdot C_t^+ \quad (9)$$

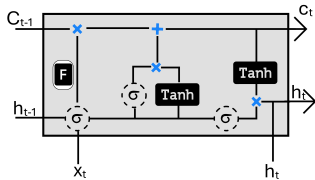


Figure 3. LSTMs internal gates structure

(c) The Output Gate: After getting data streams from above gates, The internal cells data are then passed through tanh as a final output to the next cell, as shown in equations 10, 11.

$$o_t = \sigma(U_o x_t + V_o h_{t-1} + b_o) \quad (10)$$

$$h_t = o_t \cdot \tanh(C_t) \quad (11)$$

5.1. Data Preparation

Raw data by itself isn't enough to train deep learning models. So, we need to prepare the data to make it useful and help the model learn better. First, we separate the data into input features and target classes. Then after, we apply exploratory data analysis (EDA) to find and fix any missing or duplicate data. After that, we use a standard scaler to adjust the input features and target classes so they are on the same scale, which helps the model to learn faster. To tackle the linear relationship between input and target, we transform them into power of originals for instance the interactions of the key features $\phi_d(InputFeatures) = [1, x_1, x_2, \dots, x_n, x_1^2, x_1 x_2, x_1 x_3, \dots, x_n^2, x_1^3, x_1^2 x_2, \dots, x_n^d]$. To check how well our model works on new data and avoid it from learning too much from the training data, we use a method called k-fold cross-validation with k=5. This means we test the model on different parts of the data. We evaluate the cross-validation using equation 12 that helps us calculate how well the model performs across different evaluation parameters, discussed in section 6. Apart from all these the high level view is demonstrated in figure 4, and table 2 provides further details about the different parameters with its respective values utilized during this research study.

TABLE 2. MODEL TRAINING WITH MULTIPLE PARAMETERS

Parameter	Value
Layer	2
Hidden layers	64
Batch Size	64
K-fold cross validation	5
Loss Function	MSE
Optimizer	Adam
Activation Function	Sigmoid and Tanh
Normalization	L2
Initial Learning rate	0.001
Scheduling technique	ReduceLROnPlateau
Factor	0.5
Mode	Minimum
Patience	5
Features Scaler	Standard Scaler

$$\text{Validation Error} = \frac{1}{K} \sum_{i=1}^K E_i \quad (12)$$

where: k s the number of folds,
 E_i is the error (MSE) on the i -th fold.

6. Results and Discussion

In this section, we present an experimental analysis that provides in-depth insights into our proposed approach and other baseline deep learning algorithms, such as Convolutional Neural Networks (CNN), Gated Recurrent Units (GRU), and Fully Connected Feedforward Neural Networks (FCN), across different evaluation metrics. Both the Base Station and Flying-RIS estimate the optimal channel to serve the drone located in a specific location. Sometimes, the direct line of sight was blocked from one serving Base Station viewpoint but clear to another one, which provides continuous service to the serving drone, as shown in Figure 5. In addition to that, pathloss scale is varying from different

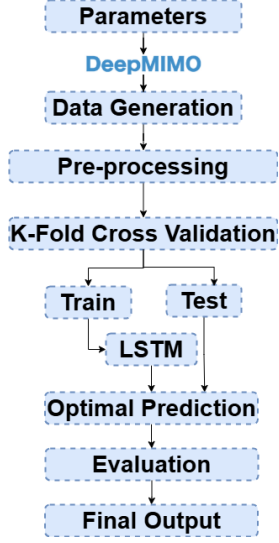


Figure 4. From Top-to-bottom flow of our proposed approach viewpoints, as shown in figure 6. For smooth training, we calculate the magnitude of all OFDM channels based on Equation 13. A sample distribution of the highest beam (only beam 1) is demonstrated in Figure 7.

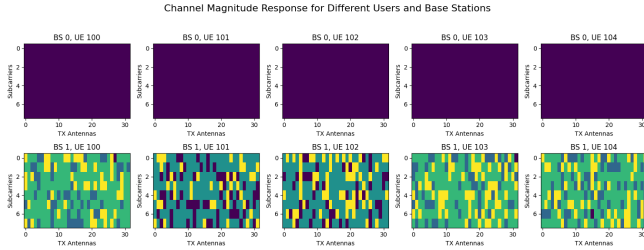


Figure 5. Channel Magnitude Response for RIS and Base Station

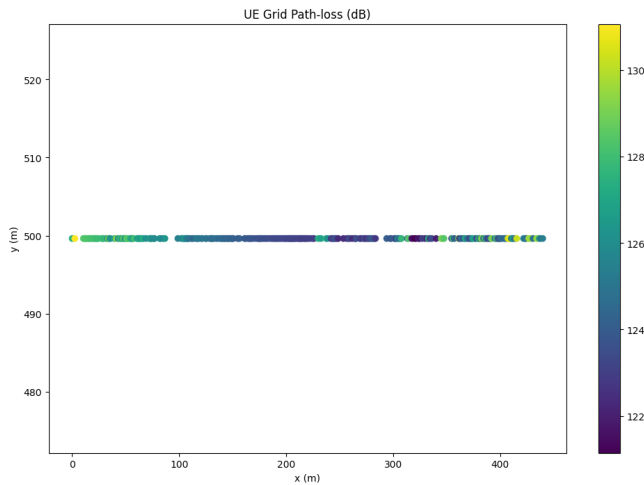


Figure 6. Demonstration of path-loss dynamics from a Flying-RIS to a serving drone

$$|a + bi| = \sqrt{a^2 + b^2} \quad (13)$$

The deep learning model we've developed has shown excellent performance in predicting the best communication

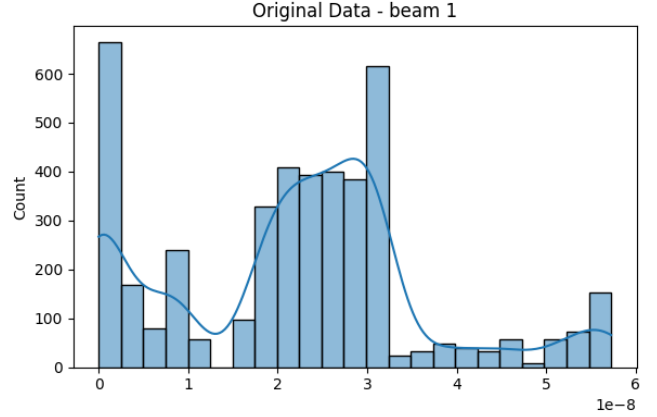


Figure 7. Illustrative representation of the beam distribution pattern channels for the '01 Drone' scenario, especially when tested with different K-fold cross-validation methods. By using detailed information about various OFDM channels, such as signal loss, power levels, user and Base Station locations, and other channel conditions, our model has successfully identified the underlying patterns and complexities of the '01 Drone' wireless environment. As seen in Figure 8, the comparison between the predicted and actual channel strengths shows a strong match, with most data points closely following the ideal diagonal line. This close alignment indicates that the model is reliable in dealing with various link conditions, including line-of-sight (LOS), non-line-of-sight (NLOS), and complete blockage, providing accurate predictions even in difficult situations.

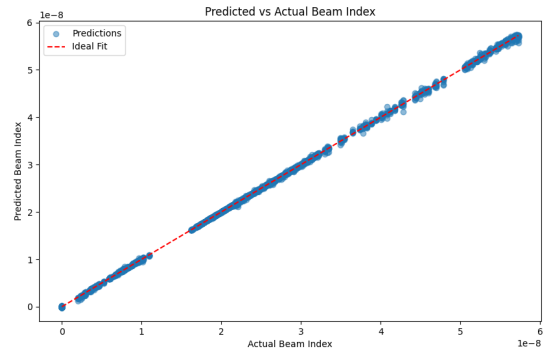


Figure 8. Predicted vs. actual values achieved with our proposed approach

In Figure 9, the average training and validation loss values across epochs show that the model's learning is stable. This confirms a steady decrease and convergence to a low value as training progresses and suggests that the model effectively adjusted its parameters to reduce prediction errors. Additionally, the validation loss consistently matches the training loss, which shows that the model's ability to generalize well prevents overfitting.

The coefficient of determination (R^2) gives us more information about how well the model can predict outcomes, and it's calculated using equation 14. As shown in table 3 and figure 10, our suggested model can find and use complicated patterns in complex wireless channels better

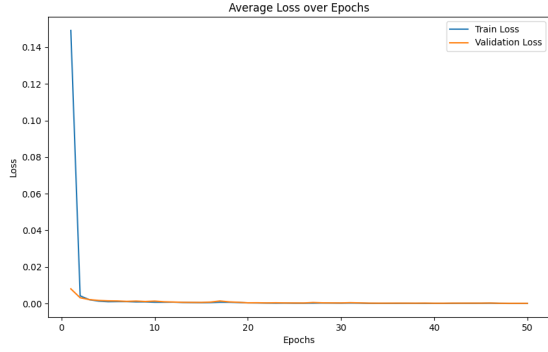


Figure 9. Training vs. validation loss curve of the proposed approach across epochs
than traditional methods, making it better at choosing the best channels.

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}, \quad (14)$$

y_i is the observed value,
where: \hat{y}_i is the predicted value,
 \bar{y} is the mean of observed values,
 n is the number of observations.

TABLE 3. PERFORMANCE METRICS FOR DIFFERENT MODELS

Parameter	FCN	CNN	GRU	LSTM
R^2	0.998832	0.999290	0.999749	0.999881
MAPE	1.729409%	0.797078%	0.691961%	0.290092%

The mean absolute percentage error (MAPE) is yet another important evaluation parameter to analyze our model performance and how accurate predictions are made during learning process, calculated using equation 15. From table 3 we can analyze that our proposed approach did much better than other base line methods. Figure 11 shows that our model is very good at reducing errors as the training progresses, either dealing with very weak or very strong signals.

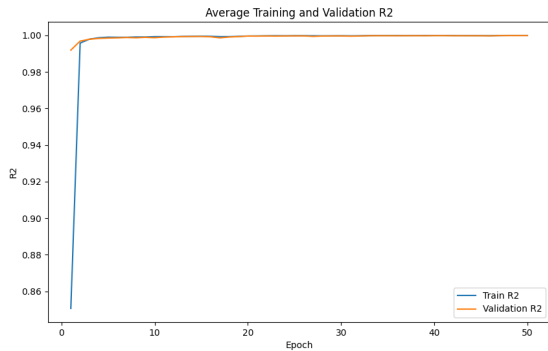


Figure 10. Visualization of the coefficient of determination (R^2) curve of the proposed approach across epochs

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100 \quad (15)$$

Besides MAPE and R^2 , other metrics like mean absolute error (MAE) and root mean square error (RMSE) were also utilized to assess the model's performance, based on equations 16 and 17 and by analyzing Figures 12 and 13, we observe that our proposed model has lower overall prediction errors and handle outliers and noise in the data effectively. By balancing accuracy and reducing errors, the suggested model consistently performs better across different evaluation standards, making it a strong and efficient choice for accurate channel prediction.

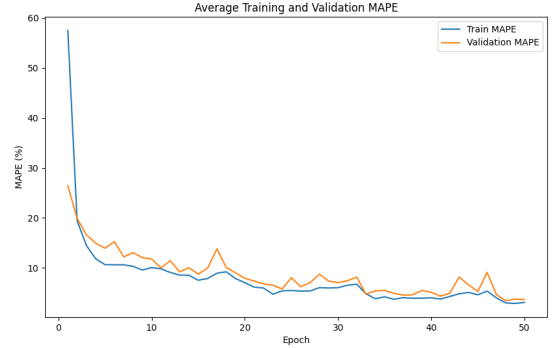


Figure 11. MAPE curve showcasing the performance of the proposed approach

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (16)$$

$$RMAE = \sqrt{\frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|} \quad (17)$$

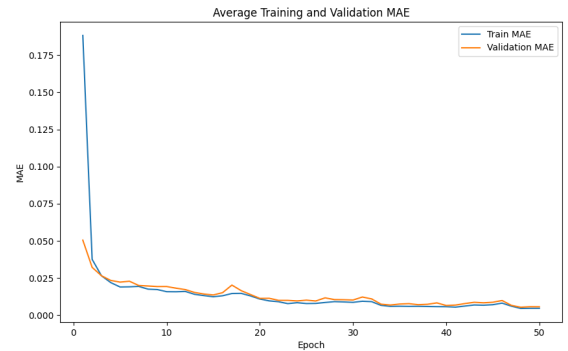


Figure 12. MAE curve showcasing the performance of the proposed approach

7. Conclusion

In this research, we performed experiments on channel estimation in a drone scenario called “01 drone” operating at sub-THz frequencies (0.2 THz) using a ground Base Station and a Flying-RIS [17]. The main goal was to provide continuous beams to serve the drone at specific locations—either from the ground Base Station or the Flying-RIS—under Line-of-sight (LoS), Non-line-of-sight (NLoS), or complete blockage conditions. To achieve this, we employed a multi-layer long short-term memory (LSTM) deep-learning model

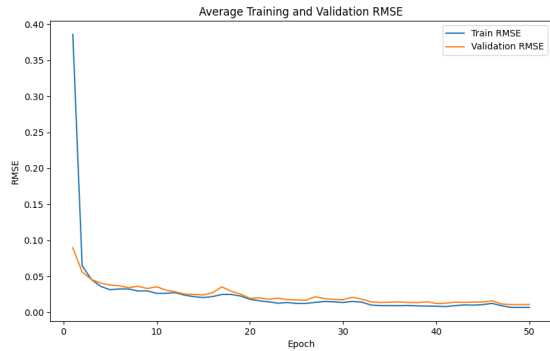


Figure 13. RMSE curve showcasing the performance of the proposed approach

[22], which can track the drone's three-dimensional positions and capture long-term dependencies in sequential data more effectively than a gated recurrent unit (GRU) model [14], thanks to its input, forget, and output gates and its internal memory that provide finer control over information flow.

Furthermore, to give competing models (CNN, GRU, FCN) an additional edge, we used the Huber loss function during training, which is less sensitive to outliers. Despite this, these models still struggled to outperform our proposed approach across various evaluation metrics, as discussed in Section V. In future work, we will conduct experiments at different frequencies and scenarios using more advanced and relevant deep-learning models and we will develop Digital Twins (DT) for such scenarios.

8. Acknowledgments

This work was supported by the Italian National Inter-University Consortium for Telecommunications (CNIT) under the European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101096342 (HORSE project).

References

- [1] G. Callebaut, L. Liu, T. Eriksson, L. Van der Perre, O. Edfors, and C. Fager, "6g radio testbeds: Requirements, trends, and approaches," *IEEE Microwave Magazine*, vol. 25, no. 4, pp. 14–31, 2024.
- [2] R. Ntassah, G. M. Dell'area, and F. Granelli, "User classification and traffic steering in o-ran," *IEEE Open Journal of the Communications Society*, 2024.
- [3] M. A. Hassan, R. Javed, F. Granelli, X. Gen, M. Rizwan, S. H. Ali, H. Junaid, S. Ullah *et al.*, "Intelligent transportation systems in smart city: a systematic survey," in *2023 International Conference on Robotics and Automation in Industry (ICRAI)*. IEEE, 2023, pp. 1–9.
- [4] J. Sengendo and F. Granelli, "Building network digital twins part i: State synchronization," in *2024 3rd International Conference on 6G Networking (6GNet)*. IEEE, 2024, pp. 182–188.
- [5] R. Fedrizzi, A. Bellin, C. E. Costa, and F. Granelli, "Building the digital twin of a mec node: a data driven approach," in *2023 IEEE 9th International Conference on Network Softwarization (NetSoft)*. IEEE, 2023, pp. 444–449.
- [6] F. Granelli, F. H. Fitzek, T. Doan, and G. Nguyen, "Tutorials: Computing in communication networks," in *2023 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*. IEEE, 2023, pp. x–xv.
- [7] J. Kim, Y. Ahn, S. Kim, and B. Shim, "Deep learning-aided parametric sparse channel estimation for terahertz massive mimo systems," *IEEE Transactions on Cognitive Communications and Networking*, 2024.
- [8] M. Di Renzo, A. Zappone, M. Debbah, M.-S. Alouini, C. Yuen, J. De Rosny, and S. Tretjakov, "Smart radio environments empowered by reconfigurable intelligent surfaces: How it works, state of research, and the road ahead," *IEEE journal on selected areas in communications*, vol. 38, no. 11, pp. 2450–2525, 2020.
- [9] Q. Hu, F. Gao, H. Zhang, S. Jin, and G. Y. Li, "Deep learning for channel estimation: Interpretation, performance, and comparison," *IEEE Transactions on Wireless Communications*, vol. 20, no. 4, pp. 2398–2412, 2020.
- [10] H. Huang, J. Yang, H. Huang, Y. Song, and G. Gui, "Deep learning for super-resolution channel estimation and doa estimation based massive mimo system," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 9, pp. 8549–8560, 2018.
- [11] X. Zhang, L. Xu, L. Xu, and D. Xu, "Direction of departure (dod) and direction of arrival (doa) estimation in mimo radar with reduced-dimension music," *IEEE communications letters*, vol. 14, no. 12, pp. 1161–1163, 2010.
- [12] A. Alteneiji, U. Ahmad, K. Poon, N. Ali, and N. Almoosa, "Angle of arrival estimation in indoor environment using machine learning," in *2021 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*. IEEE, 2021, pp. 1–6.
- [13] S. K. Hinga, O. T. Ajayi, and T. Ogunfunmi, "Deep learning-based path loss prediction model for 5g mmwave," in *2022 IEEE Global Humanitarian Technology Conference (GHTC)*. IEEE, 2022, pp. 114–120.
- [14] N. Abuzainab, M. Alrabeiah, A. Alkhateeb, and Y. E. Sagduyu, "Deep learning for thz drones with flying intelligent surfaces: Beam and handoff prediction," in *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2021, pp. 1–6.
- [15] T. S. Rappaport, Y. Xing, O. Kanhere, S. Ju, A. Madanayake, S. Mandal, A. Alkhateeb, and G. C. Trichopoulos, "Wireless communications and applications above 100 ghz: Opportunities and challenges for 6g and beyond," *IEEE access*, vol. 7, pp. 78 729–78 757, 2019.
- [16] D. Team, "Deepmimo: A generic dataset for mmwave and massive mimo applications," 2024, accessed: 2024-12-07. [Online]. Available: <https://www.deepmimo.net/versions/v2-python/>
- [17] DeepMIMO, "DeepMIMO O1 Drone Scenario," <https://www.deepmimo.net/scenarios/o1-drone-scenario/>, accessed: 2024-12-07.
- [18] R. Inc., "Remcom: Electromagnetic simulation software," 2024, accessed: 2024-12-07. [Online]. Available: <https://www.remcom.com/>
- [19] N. Qasem, "Measurement and simulation for improving indoor wireless communication system performance at 2.4 ghz by modifying the environment," *IEEE Access*, 2024.
- [20] H. Obeidat and G. T. El Sanousi, "Indoor propagation channel simulations for 6g wireless networks," *IEEE Access*, 2024.
- [21] A. Alkhateeb, "Deepmimo: A generic deep learning dataset for millimeter wave and massive mimo applications," *arXiv preprint arXiv:1902.06435*, 2019.
- [22] S. Hochreiter, "Long short-term memory," *Neural Computation MIT-Press*, 1997.
- [23] "torch.nn.lstm — pytorch 2.0.1 documentation," 2024, accessed: 2024-12-10. [Online]. Available: <https://pytorch.org/docs/stable/generated/torch.nn.LSTM.htmltorch.nn.LSTM>
- [24] T. D. Science, "Building a lstm by hand on pytorch," 2023, accessed: 2024-12-10. [Online]. Available: <https://towardsdatascience.com/building-a-lstm-by-hand-on-pytorch-59c02a4ec091>