UAV-Facilitated 3D Signal Mapping via Temporal-Spatial Features and Neural Networks

Fin Mead*, Yi Ren*, Ruoting Xiong*, Fang-Jing Wu†, Taoyang Wu*
*School of Computer Science, University of East Anglia, Norwich, UK
†Department of Computer Science and Information Engineering, National Taiwan University, Taiwan
{f.mead, e.ren, ruoting.xiong, taoyang.wu}@uea.ac.uk, fangjing@csie.ntu.edu.tw

***Abstract*—The rise of complex robotics systems for 6G infrastructure is generating demand for greater environmental mapping and monitoring, with increasing scale, complexity, and precision. A key area of development is within 3D signal estimation for cellular and robotic infrastructure. We present a method for predicting cellular network properties in 3D by combining sparse real-world collection methods with feature engineering and neural networks. We show this model applied to our experimental drone-based dataset, consisting of network, environment, time-series and positional fields. Through this approach, we show 3D predictions for cellular network conditions across a range of test sizes and formats including vertical and non-uniform subsets. Our approach shows promise in applications such as cellular map generation, drone networks, fleet management, mobile 6G infrastructure and comprehensive environmental mapping.**

***Index Terms*—signal prediction, drone mapping, sensor fusion, machine learning, 6G**

I. Introduction

With the adoption of drone delivery services in urban and rural environments, ensuring seamless and reliable connectivity is crucial. In the future, drones will likely rely on cellular communications for navigation, telemetry, and real-time data exchange. However, existing cellular networks are optimised for ground-level users, and their performance in the vertical dimension is not clearly defined. Traditional 2D coverage maps do not account for signal variations at different altitudes. Buildings, trees, topographical features as well as tropospheric conditions, can create dead zones, which could lead to disconnections, mission failures, or safety hazards. Consequently, 3D signal strength maps are key to helping identify and mitigate these risks.

In large-scale environments, generating 3D cellular signal strength maps has the following challenges: a) Granular data collection is costly and time-consuming for generating high-resolution maps; b) Direct measurements can be impractical due to aerial restrictions; and c) Signal conditions are subject to change. Maps require regular updates and maintenance, which are further complicated by a) and b).

A key question arises: *How can we efficiently predict signal strengths given limited measured data?* To address this, researchers have explored a variety of techniques, such as interpolation methods [1], [2], machine learning approaches [3]–[5], nd deep learning models [6], [7]. However, these approaches face scalability and large dataset issues in sparse or vertical signal environments. Transformer architectures and Gaussian splatting [8]–[13] have shown promise in modelling complex spatial dependencies and reducing data requirements. However, both methods require considerably large datasets for dependable predictions. Beyond telecomms, these approaches hold potential in environmental monitoring and disaster response, where extracting maximal insight from minimal measurements is essential.

Our work addresses the challenge of scalable and accurate 3D signal mapping for large-scale environments. First, we extend traditional 2D signal models by incorporating altitude-based predictions, enabling more precise coverage estimation critical for cellular infrastructure. Second, we optimise data collection by identifying key signal features and determining the most informative sampling locations, reducing the need for exhaustive measurements. Finally, we propose a lightweight artificial neural network (ANN) framework that enables accurate 3D signal prediction from sparse UAV measurements, leveraging tower and spatial features to balance accuracy and computational efficiency. Together, these contributions advance the field toward practical, scalable solutions for next-generation network planning and beyond. Our approach contributes to the realisation of AI-enabled, UAV-assisted 6G infrastructure capable of self-sustaining, data-driven wireless optimisation.

The paper is structured as follows: we survey related work in Sec. II, and propose the system model in Sec. III. Sec. IV discusses the dataset. Sec. V outlines the proposed scheme and we present an evaluation in Sec. VI and conclude our findings in Sec. VII.

II. Related Work

A. Geospatial Interpolation

Recent works [1], [2], [14], [15] have proposed geospatial interpolation strategies for signal prediction. Methods as inverse distance weighting (IDW) [1] and Kriging [2], have been widely used for environmental, behavioural and signal prediction due to their efficiency. However, these methods struggle to model complex spatial relationships and propagation effects in large-scale deployments. With the rise of 6G infrastructure, predictive models [14], [15] account for highly dynamic environments with expanding network demands

across various industries, including robotic automation, UAV communication, and smart cities.

B. Machine Learning

Another branch of works [3]–[5] attempts to predict signal strength using machine learning methods. UAV-based data collection and machine learning have significantly improved the accuracy of signal strength estimation. UAVs offer a promising tool for on-demand data collection and radio environment mapping (REM) as demonstrated by [3], which leveraged UAV vision and deep learning for aerial networks. The work [4] proposed a signal hunting solution where drone mobility plays a key role in localising signal sources. The authors in [5] explored path-planning optimization for maximizing coverage efficiency. While these methods are effective in structured settings, they face challenges in unstructured, dynamic environments.

C. Deep Learning

Leveraging Ensemble learning methods (Random Forest, XGBoost, LightGBM) has enabled better spatial modelling but remains computationally intensive [6]. Deep learning approaches, such as U-Net-based 3D signal prediction [7], have demonstrated strong performance in urban environments. However, high data requirements and computational costs hinder their scalability in large-scale or real-time scenarios.

D. Transformer Models

Transformer models and other emerging techniques have recently gained attention for their ability to model complex spatial relationships through self-attention mechanisms [8], [9]. Originally developed for NLP, transformers have been successfully applied to spatial field prediction tasks [10], thanks to their ability to capture long-range dependencies and contextual interactions—key for modeling multi-source signal propagation. Another emerging method, Gaussian splatting, has demonstrated state-of-the-art performance in visual rendering [11], [12] and is now being explored for signal propagation modeling [13]. By approximating continuous fields with anisotropic Gaussians, it offers a potentially data-efficient alternative to traditional deep learning, well-suited for sparse drone-collected data. However, many existing methods still face challenges in scalability, efficiency, and domain adaptation.

This work aims to address these gaps by integrating a specialised dataset with feature engineering with neural network-based spatial and temporal modelling. Leveraging efficient collection strategies and domain/scenario-specific feature engineering, we can achieve scalable, accurate, and computationally feasible signal prediction.

III. System Model

A. Problem Definition

As shown in Fig. 1, we consider a three-dimensional region of interest (RoI), denoted as $\mathcal{D} = \{(x_i, y_i, z_i)\}_{1 \le i \le n}$, where each tuple (x_i, y_i, z_i) represents a physical location

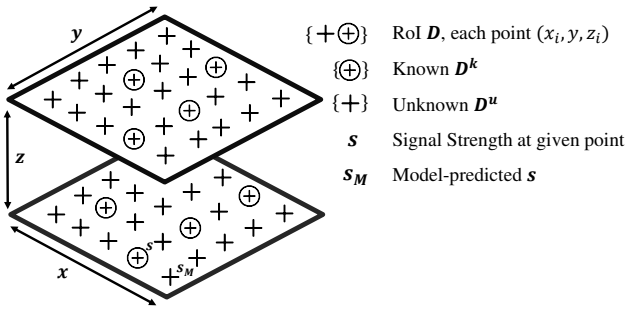


Fig. 1. Overview of system parameters and problem.

with spatial coordinates x_i, y_i, z_i, and n is the total number of points in the RoI. The received signal strength at a given location is represented as $s(x_i, y_i, z_i)$. We define $\mathcal{D}^{\mathrm{k}} \subset \mathcal{D}$ as the set of locations where signal strength measurements are known, and $\mathcal{D}^{\mathrm{u}} = \mathcal{D} \setminus \mathcal{D}^{\mathrm{k}}$ as the unknown set, where signal strength values must be predicted. The objective is to estimate $s(x_i, y_i, z_i)$ at all points in $\mathcal{D}^{\mathrm{u}}$, leveraging known measurements from $\mathcal{D}^{\mathrm{k}}$. The prediction task involves estimating the signal strength at each unknown location $(x_i, y_i, z_i) \in \mathcal{D}^{\mathrm{u}}$, where the prediction model M computes the estimated value $s_M(x_i, y_i, z_i)$. The accuracy of the prediction model is evaluated using the Mean Absolute Error (MAE), defined as:

$$\theta(M) = \frac{1}{|\mathcal{D}^{\mathrm{u}}|} \sum_{(x,y,z) \in \mathcal{D}^{\mathrm{u}}} \left| s(x_i, y_i, z_i) - s_M(x_i, y_i, z_i) \right|.$$

B. Baseline Comparison

1) Inverse Distance Weighting (IDW): [1] is a deterministic spatial interpolation method that estimates s at unknown locations $(x_i, y_i, z_i) \in \mathcal{D}^{\mathrm{u}}$ by computing a weighted average of values from nearby known points. The weights are inversely proportional to the distance between the known and unknown locations, with a power parameter controlling the influence of proximity. Given a set of known signal measurements $\mathcal{D}^{\mathrm{k}}$, the prediction model M assigns higher influence to closer neighbors in the spatial domain. In our implementation, all known points contribute to the estimate, resulting in a globally smoothed interpolation surface.

2) Kriging (Ordinary Kriging): [2] is a geostatistical interpolation method that models spatial correlation using a variogram. Unlike IDW, Kriging not only considers distance-based weights but also optimises these weights based on spatial autocorrelation. This allows it to provide more accurate estimates, particularly in heterogeneous environments where signal strength varies due to obstacles and reflections.

3) Ensemble Learning (Random Forest, LightGBM, XGBoost): Ensemble methods [6] leverage multiple weak learners to improve predictive accuracy. Random forest (RF) aggregates decision trees using bagging, while LightGBM and XGBoost utilise boosting techniques for efficient, high-performance regression. These models can capture complex

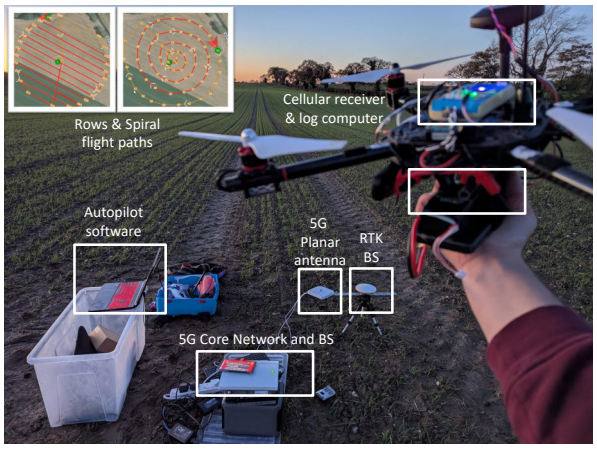


Fig. 2. The setup for our experiment, located on rural agricultural land. Pictured is the hardware setup and drone, as well as flightpaths and antenna position.

non-linear relationships in the data but require sufficient training samples to generalise well.

IV. Dataset

In this section, we introduce how we collect the dataset used for the training and testing of methods.

A. Data collection

As shown in Fig. 2, we collect data at multiple altitudes within a radius of 100 meters around a cellular transmitter, using two flightpaths at 15m and 20m. We refer to the datasets as *Rows* and *Spiral*, as shown on the top left of the figure. The Rows set follows a 2,265m path back and forth laterally. Flown at 15m. Spiral dataset is flown at 20m, starting above the cell tower, and circles outward in a Fibonacci trajectory. Both sets have a horizontal accuracy of 25mm, a vertical accuracy of 40mm and log entries approximately every 3m.

The drone hardware follows a similar architecture to the presented platform [16], where a *core* drone platform controls the drone and communicates with a *companion computer* that logs and processes data from a *sensor array*. We chose to vary flight paths to give flexibility for research into optimal flight path planning and experiment with combining path types. We produced a large dataset with telemetry, environmental and cellular fields attached to timestamps. Before training models and performing work, we process data, syncing timestamps, removing erroneous values and then splitting the datasets into batches of test train splits following randomised or block testing strategies.

Our experiment focuses on spatial signal strength mapping, so we must have a controlled cellular reception and positioning setting. RTK, precise to 25mm is used for our drone pose. We use *free5GC*, an open source 5G core network simulator connected to a 4G *WLTGFC-105 LTE Base Station* which then handles transmission via a 35dBi planar antenna. It is equipped with a planar antenna facing upward. A 5G dongle is attached to the drone log computer with a sim card paired to only *this* cellular station. These hardware choices, combined with this setup design, allowed for total control of the connection properties and antenna configuration. We note that the use of multi-generation equipment means this test system was not truly 6G however the core experiment remains equivalent to the requirements described in Section III, and is certainly useful for future system design.

B. Dataset Overview

The datasets comprise measurements collected during drone flights, capturing environment and network properties at standard increments along the flight path. The key attributes of the dataset are:

- *Geospatial Positioning:* RTK provides latitude, longitude, and altitude measurements(mm above take-off altitude) for precise localisation.
- *Signal Strength Metrics:* Cellular logger captures RSSI, RSRQ, RSRP, and SINR, essential for signal quality estimation.
- *Tower Metadata* such as tower position, altitude and raised height and orientation, are used in calculating engineered features.

All signal values are reported in dBm, with SINR in dB. Distances are in metres and coordinates in WGS-84 decimal degrees. Time is logged in UTC. The dataset includes roughly 700 readings per flight path, and is publicly available [17]. The data was logged in the following format: {Timestamp, RSSI, SINR, Lat, Lon, Altitude, ...} e.g. {2024-11-26 15:14:07, -85.0, -20.0, 52.6081822, 1.5428175, 15200, ...}

C. Processing

We evaluate model performance using spatial and temporal dataset splits. The *Random Subsampling* split provides a well-distributed test baseline, while the *Spatial Block* split simulates real-world generalisation by requiring extrapolation to unobserved regions, akin to dead zones or signal drops, see Fig. 3. These splits are applied to four dataset configurations across both flight paths. Models are tested using within-layer (same dataset) and cross-layer (trained on one, tested on the other) strategies to assess robustness. For each setup, predictions are repeated 10 times, reporting the median MAE while varying test set sizes to evaluate performance under data sparsity. The *Temporal Split* trains on early flight data and tests on future readings, reflecting real-time prediction needs such as signal-based flight planning. Tower-related features—distance, bearing, and elevation angle—are described in Section V-B. After consulting industry professionals, they informed us of a rise in attempts to create RSSI plots from user data; Therefore, we focused on RSSI prediction for this experiment.

V. Neural Network Architecture

A. System Overview and Architecture

Fig. 4 shows an overview of our prediction pipeline, comprised of the following steps:

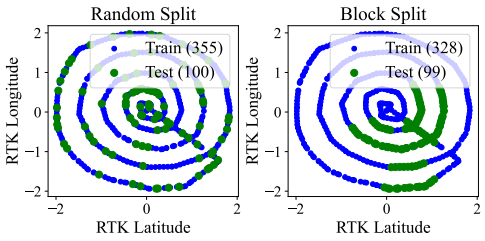


Fig. 3. Example test-train split: green indicates the held-out subset used for testing. In cross-layer prediction, this format is repeated, but the test data comes from a different layer or dataset.

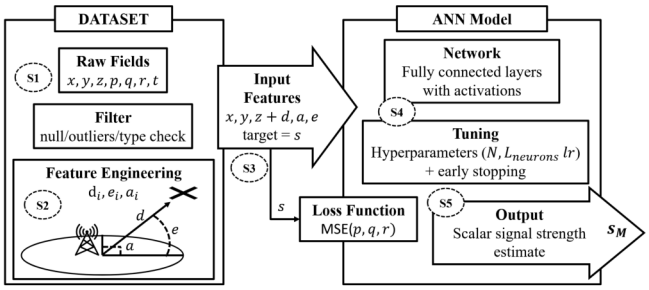


Fig. 4. Pipeline Architecture

- *Step 1 (S1): Raw Data Preprocessing and Filtering:* We apply thresholding, type enforcement, and noise filtering to the raw input fields.
- *Step 2 (S2): Feature Engineering:* Spatial and contextual features are derived from GPS coordinates and tower metadata.
- *Step 3 (S3): Input Features:* Coordinates and engineered features values form the model input with true signal values used for training.
- *Step 4 (S4): ANN Training and Tuning:* An ANN is trained using the input features and tuning config. If predicting new values, input is passed through the learned network.
- *Step 5 (S5): Signal Quality Prediction:* A vector of RSSI readings with each corresponding to an input point.

Unlike prior ANN-based or U-Net approaches [6], [7], our model fuses directional tower features (distance, azimuth, elevation) with 3D coordinates and optional temporal context. This combination enables cross-altitude generalisation from sparse data, reducing reliance on dense spatial sampling. The novelty lies in integrating engineered tower features and raw dataset as input to a lightweight fully-connected architecture for UAV-scale 3D signal prediction.

B. Input and Feature Extraction (S1, S2 & S3)

The input to the ANN consists of two sets of features: raw spatial coordinates and engineered propagation-aware features. Specifically, we use the local Cartesian coordinates (x_i, y_i, z_i) to represent the drone's position at each timestamp. To enhance the spatial context, we additionally compute engineered features:

- Distance d_i: Euclidean distance from the drone to the tower, derived from local XYZ coordinates.
- Azimuth angle a_i: The horizontal bearing between the drone and tower, computed from GPS coordinates.
- Elevation angle e_i: The vertical angle between the drone and tower, computed using relative height and horizontal distance.

Together, the input vector is constructed as: $\mathbf{x}_i = [x_i, y_i, z_i, d_i, a_i, e_i]$. This combination provides absolute position and directional signal context to the model. The inclusion of a, e and d is motivated by the directional nature of antennas and propagation patterns in real-world environments. If testing in timeseries configuration, a set of attached timestamps t is added to the input.

C. Model Training (S4)

1) ANN Optimisation: We use a feedforward ANN composed of 4 fully connected layers. Each layer is followed by batch normalisation and ReLU activation. Dropout is applied after each hidden layer to reduce overfitting. The architecture is trained using mini-batch gradient descent with the Adam optimiser. The model passes forward the input through its hidden layers, applying weights, biases, activation functions, batch normalisation, and dropout. We use the *keras* tuner module to speed up the process of hyper parameter tuning.

2) Loss Function: While training, we use this MSE loss:

$$\mathcal{L}_{\text{RSSI}} = \frac{1}{N}\sum_{i=1}^{N}(\hat{s}_i - s_i)^2,$$

where $\hat{s}_i$ is the predicted RSSI value, and s_i is the ground truth value for a sample, i. For single-target predicting, such as RSSI, we observed that directly predicting RSSI yields better accuracy than deriving it from other indicator fields, where $\hat{p}_i, \hat{q}_i, \hat{r}_i$ are the predicted RSRP, RSRQ, and SINR values, and p_i, q_i, r_i are the real values. In those case, s is replaced by p, q, r in the loss function and s derived the predictions.

D. Output (S5)

Finally, a prediction of signal quality is generated using the trained model. It outputs a single continuous value representing the predicted RSSI for each input sample. This is achieved through a final dense (fully connected) output layer with a single neuron and no activation function, enabling direct scalar regression. Once trained, the ANN can be used to predict signal strength at arbitrary locations by performing a forward pass with new spatial inputs. For each point, the model processes the input features through its learned network structure and outputs a single scalar representing the estimated RSSI. This allows the system to generalise signal strength predictions across previously unseen areas.

E. ANN Experimentation

1) Hyperparameter tuning: It is important to tune parameters with even basic neural networks like this. To create

generalised models, we must train and test on different sets whilst adjusting parameters such as the number of layers L, neurons per layer N_{neurons}, dropout rate p_{dropout}, and learning rate η. This process can easily create overfitted models unless adequate attention is paid toward the types of splits we perform and test against.

2) Input: It is easy to generate models which are *constrained* rather than *informed* by the size or contents of the input. During experimentation, we found that including extra features of the tower d_i, a_i, e_i can reduce the error when testing models on points within of the same layer, and drastically reduce error when predicting unseen cross layer predictions. To generalise well, we must find features that aid the model to predict unseen data that is different in topology.

3) Complexity: We found that reducing the complexity of the model such as lowering L and N_{neurons} created a more generalisable model, able to more consistently predict cross-dataset. This was especially useful when the size of the train set is smaller, as it helps further general trends rather than fitting to the available data, which may misrepresent the rest of the dataset. This resulted in a 1000 epoch convergence cap, layers: 4, Neurons per Layer: 256, Dropout Rate: 0.5, and Learning Rate: 0.01.

VI. Performance Evaluation

In this section, we discuss the results of each test configuration and discuss the reasoning for perceived trends. We evaluate the ANN and baselines to answer three questions: (1) How well do they generalise across altitude layers? (2) How do data sparsity and partitioning affect performance? (3) Can directional and temporal features improve prediction stability?

A. Experimental Configuration

In order to test the model performance, we tested each baselines and ANN scheme using a range of configurations. Outlined in Section IV-C, we test under same-layer and cross-layer conditions using block and random splits, as well as trajectory-based splits.

B. Impact of Altitude - Cross-layer Prediction

As shown in Fig. 5, we found that baselines more accurately predict within the same layer in general, while ANN predicts well across sets in most cases, both in the random and block splitting. This demonstrates that the ANN learns vertical dependencies beyond spatial continuity, indicating its potential in future multi-altitude 6G planning tasks. IDW and Kriging struggled to predict readings at different altitudes. This may be because they are fundamentally 2D spatial interpolation methods that assume signal values change smoothly over distance, and do not account for vertical non-linear relationships between altitude and signal strength. Ensemble performs better likely due to how it combines methods to create a more generalised approach, as well as the tree structure being able to store relationships in a non-linear capacity.

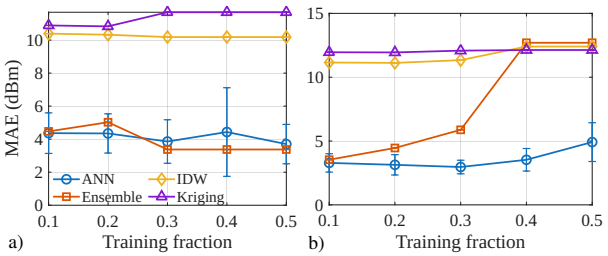


Fig. 5. Comparison of cross-layer model performance using where Rows predicts spiral (a) and Spiral predicts rows (b). Other methods suffer while ANN learns vertical dependencies beyond basic spatial continuity.

C. Impact of Partitioning on Performance

As shown in Fig. 6, random splitting shows moderate results from most methods, where Median MAE does not exceed 3.5. This is understandable as it is the simplest case where even basic interpolation approaches can handle filling in the gaps of 2D input sets. Block splitting reveals approaches struggled with larger empty regions, with ANN error increasing the least overall and performing on top in Spiral set for larger splits. This begins to reveal the benefits of ANN, the lower error trend suggests it can infer topological or directional propagation behaviour absent from 2D interpolators. While more primitive approaches struggle to improvise with points far from the known area, ANN extrapolates known relationships well.

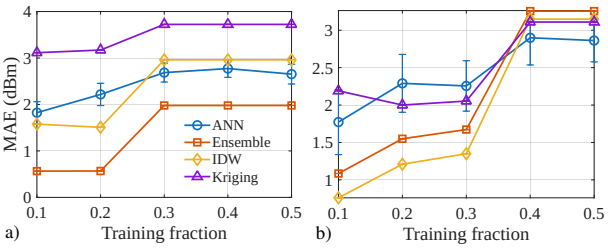


Fig. 6. Comparison of same-layer block predictions, where Rows (a) is predicting a subset region of itself and Spiral (b) also predicts a subset region of itself. Under restrictive data conditions, ANN maintains reasonable performance.

D. Impact of Azimuth, Elevation Angle and Distance

As shown in Fig. 7, the use of Azimuth, Elevation Angle and Distance (AED) proved useful for reducing the confusion of model predictions. While AED features are generally less interpretable and spatially intuitive than raw coordinates, ANN maintains strong performance even when trained solely on AED. Traditional methods degrade significantly under AED input, suggesting that ANN is uniquely capable of extracting useful patterns from representations. This indicates the model is primarily learning channel characteristics and directionality, rather than relying or overfitting absolute coordinates. This insight opens up potential for generalising across paths and environments, and suggests future work could involve learning tower pose to further improve predictions. Fig. 7 also shows how the use of a, e, d compared to

x, y, z produces a worse accuracy in most methods. However, the ANN is better able to generalise using either input set. This reduced accuracy is possibly due to the larger network size being passed a smaller feature size.

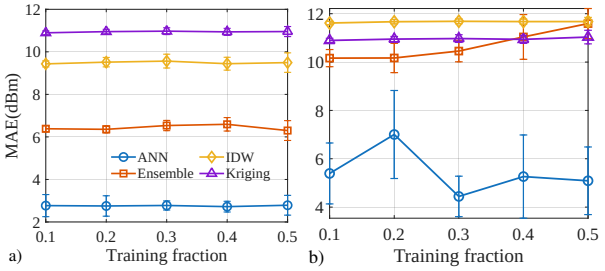


Fig. 7. Use of exclusively (a) x,y,z via RTK vs (b) Azimuth, Elevation, 3D Distance, tested on a random split of Rows cross-predicting Spiral layer. Robust ANN performance when coordinates are removed suggests channel fading learning while primitive methods fail.

E. Impact of Time-Series

We considered how methods may discern s_i from ordered timestamps attached to positions and features. ANN performs better than other baselines where variation is introduced. Fig. 8 shows that timeseries improved the performance of ANN, more than other methods. Using timestamps generally also stabilised otherwise turbulent predictions. Our results show accuracy is linked heavily to flightpath. Time-series inclusion reduces variance and improves accuracy in prediction, indicating potential for trajectory-aware prediction. This is useful in approaches for trajectory prediction, where organised paths may be utilised. However, this does not reflect real-world flight paths where bespoke surveying may not be possible due to conditions and restrictions, so we must consider, in future, how we might combine both flight path approaches and where the valuable flight behaviour lies when predicting future points.

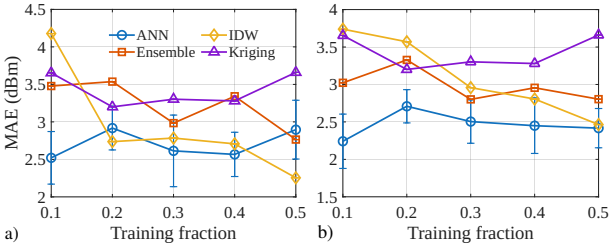


Fig. 8. Predicting RSSI based on Spiral drone trajectory using (a) x,y,z position vs (b) time-series x,y,z. Inclusion of timeseries improves accuracy and reduces variation.

We also examined ANN error over time on both sets and found it increased over time, but not equally. Spiral error showed little correlation with time, likely due to spatial revisits. Rows error increased more, possibly due to limited spatial exposure during training. Perhaps we should consider combining path strategies to better capture informative flight behaviours for prediction.

VII. Conclusions and Future Work

In this paper, we present an approach to sparse signal prediction using ANN combined with 3D spatial and temporal features. Our method outperforms baseline methods across block, cross-layer, and trajectory-based RSSI tasks, demonstrating consistent general performance, even for shorter paths. Our findings indicate that lightweight ANN architectures can outperform traditional methods even with sparse UAV datasets, specifically vertical reading estimation, highlighting a viable path toward autonomous, data-efficient 6G network monitoring and planning where understanding vertical signal propagation is paramount. In future, larger-scale and variable-altitude datasets will be essential for an accurate and generalised approach. We aim to explore alternative NN approaches and integrate dynamic flight path planning alongside network digital twins to further enhance prediction accuracy and support scalable deployment in real-world drone communication infrastructure.

References

[1] F. Chen and C. Liu, "Estimation of the Spatial Rainfall Distribution using Inverse Distance Weighting (IDW) in the Middle of Taiwan," *Paddy and Water Environment*, vol. 10, pp. 209–222, 2012.

[2] S.-S. Jan *et al.*, "Received Signal Strength Database Interpolation by Kriging for a Wi-Fi Indoor Positioning System," *Sensors*, vol. 15, no. 9, pp. 21 377–21 393, 2015.

[3] N. C. Matson and K. Sundaresan, "Online Radio Environment Map Creation via UAV Vision for Aerial Networks," in *Proc. IEEE INFOCOM*, 2024, pp. 81–90.

[4] A. Blaga *et al.*, "3DSAR+: A Single-Drone 3D Cellular Search and Rescue Solution Leveraging 5G-NR," *IEEE Open Journal of the Communications Society*, vol. 5, pp. 4808–4822, 2024.

[5] M. Höffmann *et al.*, "Optimal Guidance Track Generation for Precision Agriculture: A Review of Coverage Path Planning Techniques," *Journal of Field Robotics*, vol. 41, pp. 823–844, 2023.

[6] R. Shrestha *et al.*, "Radio Map Estimation in the Real-World: Empirical Validation and Analysis," in *2023 IEEE Conference on Antenna Measurements and Applications (CAMA)*, 2023, pp. 169–172.

[7] E. Krijestorac *et al.*, "Spatial Signal Strength Prediction using 3D Maps and Deep Learning," in *Proc. IEEE ICC*, 2021, pp. 1–6.

[8] M. Jaderberg *et al.*, "Spatial Transformer Networks," in *Advances in Neural Information Processing Systems*, C. Cortes *et al.*, Eds., vol. 28. Curran Associates, Inc., 2015.

[9] A. Vaswani *et al.*, "Attention Is All You Need," *CoRR*, vol. abs/1706.03762, 2017.

[10] W. He *et al.*, "A Hierarchical Spatial Transformer for Massive Point Samples in Continuous Space," in *Advances in Neural Information Processing Systems*, A. Oh *et al.*, Eds., vol. 36. Curran Associates, Inc., 2023, pp. 33 365–33 378.

[11] X. Zhao *et al.*, "NeRF2: Neural Radio Frequency Radiance Fields for 3D RF Propagation Prediction," in *Proc. MobiCom*, 2023.

[12] B. Kerbl *et al.*, "3D Gaussian Splatting for Real-Time Radiance Field Rendering," 2023.

[13] B. Fei *et al.*, "3D Gaussian Splatting as New Era: A Survey," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–20, 2024.

[14] W. Saad *et al.*, "A Vision of 6G Wireless Systems: Applications, Trends, Technologies, and Open Research Problems," *IEEE Network*, vol. 34, no. 3, pp. 134–142, 2020.

[15] S. Dang *et al.*, "What Should 6G Be?" *Nature Electronics*, vol. 3, pp. 20–29, 2020.

[16] M.-C. Dobrea and D.-M. Dobrea, "A UAV Development Platform for Intelligent Applications," in *2021 International Symposium on Signals, Circuits and Systems (ISSCS)*, 2021, pp. 1–4.

[17] F. Mead, "Drone-based Spatial 5G Signal Dataset and Prediction algorithms." [Online]. Available: https://github.com/mr-finnie-mac/DS2-Processing