Continuous Coverage Area Estimation Using Mobile Spectrum Crowdsensing and Deep Learning

Guilherme A. Thomaz, João P. de S. Machado, Marcello L. R. de Campos, and Miguel Elias M. Campista
Universidade Federal do Rio de Janeiro (UFRJ), Rio de Janeiro, Brazil
guiaraujo@gta.ufrj.br, joao.machado@smt.ufrj.br, campos@smt.ufrj.br, miguel@gta.ufrj.br

***Abstract*—Radio spectrum crowdsensing replaces costly drive tests with scalable, yet potentially highly sparse, distributed data collection. The problem of estimating the coverage area in real time as new samples arrive has not been well explored in the literature. This work proposes a solution to this issue by combining a U-Net deep learning model for interpolation with a simple post-processing algorithm for generating closed binary coverage masks. Hence, one of our main contributions compared with the state of the art is the integration of a traditional ray-tracing tool with a realistic urban mobility simulator to evaluate the temporal dynamics of crowdsensing. In simulation, the estimated coverage area error is smaller than 30% after only 300 seconds, depending on the experimental setup. In addition, this work contributes to the state of the art by evaluating the influence of mobility, geographical, and transmission parameters on the intersection-over-union between the estimated and ground-truth coverages.**

***Index Terms*—Crowdsensing, Radio Environment Map (REM)**

I. Introduction

Radio spectrum management conducted by national regulatory agencies ensures the efficient usage of this scarce resource. A critical metric in mobile networks is the coverage area in which the received signal from a given spectrum band must have sufficient power to provide acceptable service levels, as defined by regulatory standards [1], [2]. For example, regulatory agencies compute the spectrum occupancy, which, for a point-to-area mobile system, is the product of frequency bandwidth (MHz) and coverage area (m^2) [2]. The computation primarily relies on simulators that use data reported by mobile service operators as input [3]. Drive-test campaigns monitor the received signal power at specific locations to evaluate the consistency of reported data [1], [4]. Despite their crucial role in spectrum planning and interference mitigation, the high costs of drive tests limit scalability, especially in more isolated regions.

Current work in the literature has proposed using crowdsensing to leverage the large number of user devices for cost-effective spectrum data collection [5]–[7]. This approach does not replace drive test campaigns, which is still considered the ground-truth for spectrum monitoring. However, it provides a valuable alternative for identifying potential locations where discrepancies exist between actual measurements and expected coverage. All previous work focuses solely on building a Radio Environment Map (REM) to reduce the estimated signal power error within a region of interest (ROI). The lack of coverage area estimation is an important shortcoming. Additionally, REM estimation is performed after the crowdsourced dataset completion. Experiments considering a continuous spectrum monitoring system that updates its results in real time as new data arrives are not explored in the literature.

This work proposes the use of crowdsensing spectrum data to continuously update estimates of the coverage area for each given frequency band and ROI. The core contribution is a learning-based strategy for building a coverage mask from very sparse data points as they arrive in real-time. Experiments combine online data collection simulator based on SUMO (Simulator of Urban MObility), ray-tracing tool for computing the expected RSRP (Reference Signal Received Power) at each bin in a spatial grid, and U-Net deep learning (DL) model for REM interpolation [8]–[10]. Our simulation results show that the coverage area can be estimated with less than 30% error within 300 seconds in the evaluated scenarios. They also reveal the extent to which changes in the number of participants, the map shape, and radio-frequency (RF) parameters can reduce system performance. The contributions are: (i) continuous coverage-area estimation from cumulative sparse crowdsensing data; (ii) the BWS step to obtain closed coverage regions; (iii) a SUMO and ray-tracing framework to study the temporal evolution of the estimate; and (iv) an evaluation of map geometry, power, frequency, threshold, and number of participants. The interpolator is adopted from [10].

The remainder of this work is organized as follows. Section II presents related works. Section III details the proposed strategy. Section IV describes the experimental methodology. Section V discusses the results. Finally, Section VI concludes this work and presents directions for future work.

II. Related Work

This section presents prior work on spectrum crowdsensing, cell coverage estimation, DL-based radio mapping, and the dynamics of real-time spectrum data collection for continuous spectrum database updates.

Zeng *et al.* propose spectrum crowdsensing to accelerate data collection for anomaly identification in LTE networks [5]. Data-driven approaches usually face challenges with out-of-distribution (OOD) generalization, introducing bias risks. Mapping RF usage at every location across multiple ROIs and bands requires enormous data volumes and diversity. The authors, however, do not address strategies to mitigate data collection for large-scale spectrum management tasks, such as RSRP interpolation.

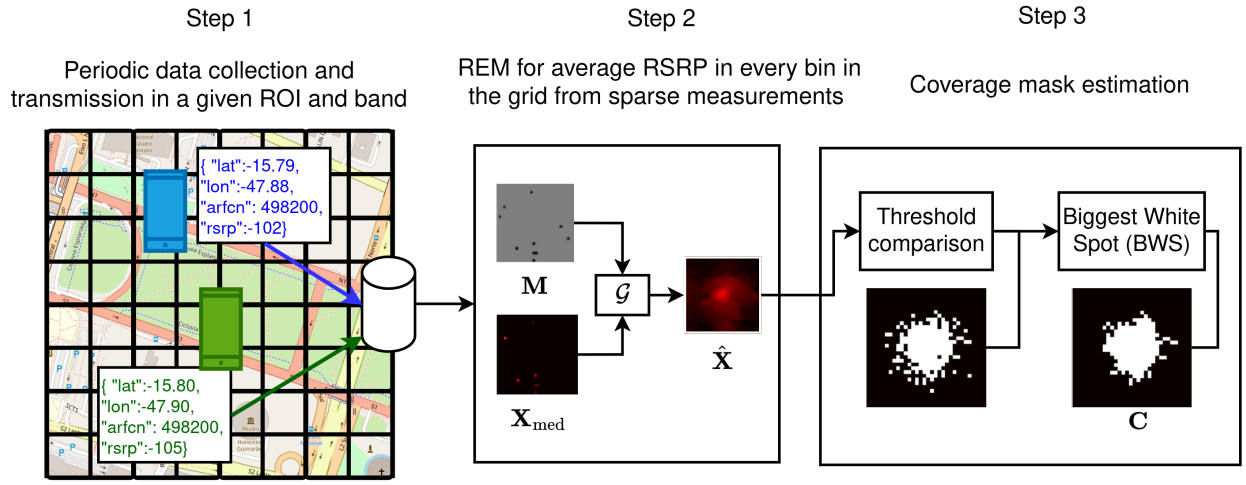


Fig. 1: Pipeline for the proposed strategy for coverage estimation. Steps 2 and 3 are periodically repeated as new crowdsensing data arrives at the center database.

Basere and Kostanic use intensive drive tests to collect RSRP samples from a cell tower in Melbourne [11]. Next, the well-known Kriging algorithm interpolates these samples to produce a complete REM. The paper discusses the impact of ROI bin size on the statistics of the aggregated RSRP from multiple users in the same bin. However, the impact of the mobility pattern used for data collection, as well as the geographical diversity of the different regions and the RF parameters, is not taken into account.

Sato *et al.* interpolate a complete REM from sparse measurements by leveraging a DL model to estimate the path loss (PL) before applying Kriging to estimate, via interpolation, the effects of shadowing [12]. The results confirm the impact of using DL in reducing the interpolation RMSE (Root Mean Squared Error). However, the gap between these theoretical results and the problem of coverage area estimation in sparse crowdsensed data is not addressed.

Lin *et al.* tackle the challenge of collecting sufficient data in practical mobility environments for spectrum-related training and prediction tasks [13]. They combine vehicular traces generated by SUMO with 3GPP PL models to produce a vehicular communication dataset in a four-lane street. The goal is to train a DL model for joint compression and reconstruction of sub6GHz and THz bands, enabling vehicles to predict occupied subcarriers for dynamic sidelink cellular access. The results report the model performance only after the simulation ends, so that results on the temporal evolution of the predictions as new data continuously arrive are missing.

Unlike previous works, this paper shifts the focus from static signal power mapping to the continuous estimation of coverage areas from highly sparse crowdsensed data. By integrating SUMO with a ray-tracing tool and a DL U-Net interpolator, the proposal uniquely captures the temporal dynamics of crowdsensing as new samples arrive. Furthermore, it bridges the gap concerning the analysis of how mobility configurations, geographical map shapes, and RF parameters impact the intersection-over-union and area of the estimated coverage masks.

III. Proposed Strategy for Coverage Estimation

This section describes the proposed strategy for estimating RF coverage from crowdsensed data. The geographic territory is logically divided into square ROIs. Each ROI is therefore divided into $N \times N$ square blocks, called bins, forming a discrete grid. Each cellular channel, identified by ARFCN (Absolute Radio Frequency Channel Number), has its carrier frequency and bandwidth. In TDD (Time Division Duplexing), the uplink and the downlink channels overlap, whereas in FDD (Frequency Division Duplexing), they are separated. For each ROI and channel, the proposed strategy involves building a REM for the expected RSRP in each bin in the grid. Figure 1 illustrates the steps involved in a single ROI and frequency band.

In crowdsensing, each participant owns a smartphone running a software for gathering data from the device's RF system. For example, G-NetTrack and RantCell applications measure RSRP, ARFCN, and geographical coordinates at each moment in time by using the `Telephony` and `Location` APIs (Application Programming Interface) from the Android operating system (OS). Clients periodically send their collected data to a web server, which incorporates it into the database in real-time, as illustrated in Step 1 in Figure 1. For a given ROI and frequency channel, the RSRP at each bin is the average RSRP reported by all users in this location and frequency channel. In practical crowdsensing deployments, these features are available inside other apps, such as apps for taxi services and urban navigation [14]. In these cases, only a small fraction of the smartphone owners may be available at each ROI and each frequency band. Moreover, crowdsensing participants are not expected to continuously stream their samples in order to reduce battery and network

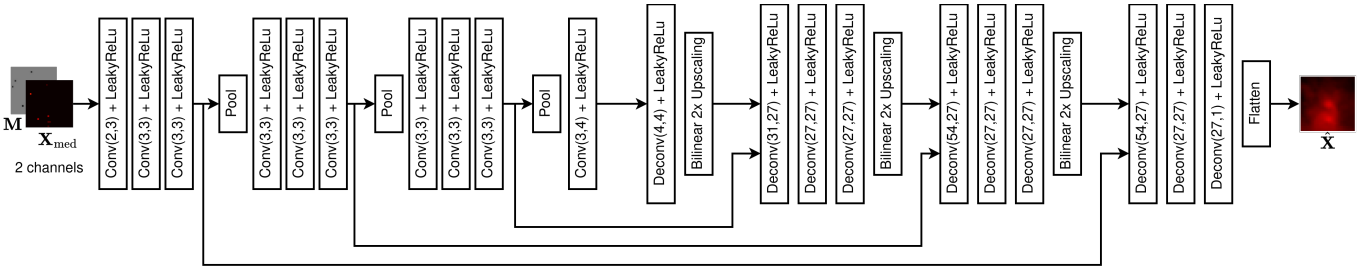


Fig. 2: Skip connection U-Net architecture. `Conv(a,b)` represents a convolutional layer with `a` input channels, `b` output channels, and kernel size 3. `LeakyReLU` represents the rectifying linear unit operation with negative slope of 0.3. `Pool` represents a layer for 2D pooling with dimensions 2×2. `Deconv(a,b)` represents a deconvolutional layer with stride and padding values equal to 1. Finally, the network also uses image upscaling layers and a flatten layer.

usage. Therefore, the problem, known as REM interpolation, is to construct an accurate estimate of the RSRP across the entire grid from RSRP measurements at only a few bins.

Formally, each element $[i, j]$ of a sparse measurement matrix $\mathbf{X}_{\text{med}} \in \mathbb{R}^{N \times N}$ reveals the RSRP at the bin located at row i and column j in the grid, also represented as bin $[i, j]$. In the absence of a measurement, $\mathbf{X}_{\text{med}}[i, j] = 0$. A binary mask matrix $\mathbf{M} \in \mathbb{R}^{N \times N}$ is used to identify bins with measurements, indicated by the value 1 in the corresponding elements. The REM generation model $\mathcal{G}$ estimates a matrix $\hat{\mathbf{X}} = \mathcal{G}(\mathbf{X}_{\text{med}}, \mathbf{M})$ using the measurements $\mathbf{X}_{\text{med}}[i, j]$, where $\mathbf{M}[i, j] = 1$. The need to differentiate bins with zero RSRP from bins with unknown values justifies the usage of a binary mask $\mathbf{M}$. That way, $\mathbf{M}[i, j] = 0$ indicates to $\mathcal{G}$ that the RSRP in bin $[i, j]$ needs to be interpolated from the RSRP in the known bins. As new samples arrive, the interpolation process must be repeated to keep the estimate $\hat{\mathbf{X}}$ up-to-date. This process is represented in Step 2 in Figure 1.

Given an estimated REM $\hat{\mathbf{X}}$, the next step involves comparing the power at each bin in the grid with a given power level threshold (δ). The result is a binary coverage mask $\mathbf{C} \in \mathbb{R}^{N \times N}$ where $\mathbf{C}[i, j] = 1$ if $\hat{\mathbf{X}}[i, j] > \delta$. Spectrum regulators define thresholds for the minimum signal power in dBm, which are typically used to define the coverage area [1]. Step 3 in Figure 1 presents the resulting coverage mask after the comparison as a black-and-white image. The white mask does not define a closed contour due to weak-signal holes within the region and along the border. Therefore, the threshold comparison is followed by a procedure to find the Biggest White Spot (BWS), defined in Algorithm 1. The result is a closed coverage region with a well-defined contour, as presented in Step 3 in Figure 1. Since each bin dimension is known, the coverage area is the total area of the coverage mask.

Algorithm 1 Biggest White Spot (BWS).

Require: Binary matrix $\mathbf{C}_{\text{in}}$
Ensure: Binary matrix $\mathbf{C}_{\text{out}}$ with the largest filled spot
1: $\mathbf{C}_{\text{filled}} \leftarrow$ binary_fill_holes($\mathbf{C}_{\text{in}}$)
2: $\mathbf{L} \leftarrow$ label_non_adjacent_spots($\mathbf{C}_{\text{filled}}$) ▷ $\mathbf{L}$ is a matrix where each non-adjacent white spot gets an integer number (1, 2, ...) called label, while the background is zero.
3: $K \leftarrow$ num_nonzero_labels($\mathbf{L}$) ▷ Count the number of white spots (labels different from 0)
4: **if** $K = 0$ **then**
5: **return** zeros_matrix ▷ Just black background. No white spot
6: **end if**
7: $T \leftarrow$ bincount($\mathbf{L}$) ▷ Count the number of occurrences of each label. T is a dictionary that maps each label to the respective number of white pixels
8: $T[0] \leftarrow 0$ ▷ Ignore background (label 0)
9: $l_{\text{max}} \leftarrow$ argmax(T) ▷ Find the label with the maximum white pixel count
10: $\mathbf{C}_{\text{out}} \leftarrow$ filter_label($\mathbf{L}, l_{\text{max}}$) ▷ Result matrix is one only at the positions where $\mathbf{L} = l_{\text{max}}$
11: **return** $\mathbf{C}_{\text{out}}$

IV. Experimental Methodology

This section describes the methodology for evaluating the performance of the proposed coverage area estimation strategy. The hardware used for simulation is an Ubuntu machine with an AMD Ryzen 9 7950X 16-Core processor, 64GB of RAM, and an NVIDIA GeForce RTX 5090 GPU for model inference. All experiments are repeated 20 times, yielding metric computation with 95% confidence interval (C.I.).

The SUMO tool [8] simulates urban regions with approximately 1,600×1,600 meters, collected from the OpenStreetMap (OSM) API. A Python script creates the SUMO subprocess, records the locations of vehicles moving around on the streets every second, and writes the results to a Comma-Separated Values (CSV) file. A second Python script in the simulation pipeline randomly selects V vehicles and produces a 32×32 mask matrix $\mathbf{M}$ at each time interval T, in seconds. The n-th binary mask $\mathbf{M}$ is equal to 1 at all positions corresponding to bins where a vehicle has been located before (from 0 seconds up to nT seconds).

The ray tracing simulation tool used by Teganya *et al.* [9] is used in our experiments to compute the signal power at each of the 32×32 bins of the ROI. The tool computes the REM $\mathbf{X}$

TABLE I: ROIs characteristics for Experiment 1.

Min Lat, Lon	**Max Lat, Lon**	**Characteristics**
-22.9180, -43.1855	-22.9046, -43.1704	Very dense urban coast
-11.3404, -48.3716	-11.3281, -48.3569	Rural zone
-12.2788, -38.9416	-12.2667, -38.9270	Medium-size metropolis
-15.8000, -47.8875	-15.7871, -47.8738	City with flat terrain
-24.9268, -53.4799	-24.9139, -53.4647	Transition to rural zone

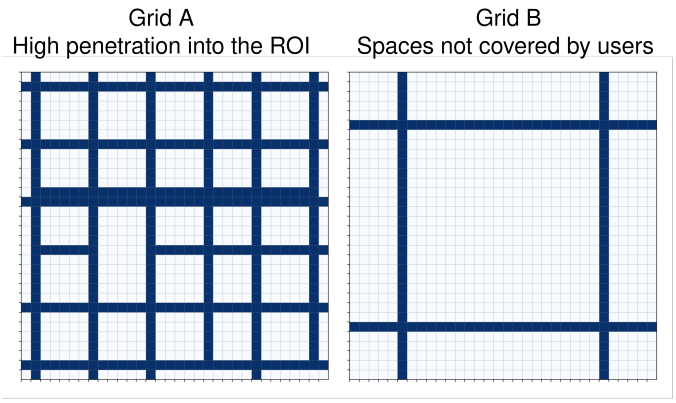


Fig. 3: Artificial 1600×1600 meters ROIs for Experiment 2. The grid contains 32×32 bins of 50×50 meters each. The bins in blue indicate the presence of a street.

assuming the presence of a radio-base transmitter at the center with carrier frequency f (MHz), and transmission power P (dBm). At each timestamp, $\mathbf{X}_{\text{med}}$ is computed as the point-wise matrix product between the binary mask at this time ($\mathbf{M}$) and the baseline REM ($\mathbf{X}$), both generated by the pipelines described above. This allows us to simulate a scenario in which a fraction of the available vehicles periodically reports crowdsensed spectrum data at a given point in space.

The system also computes $\hat{\mathbf{X}}$ at each time interval using the generator $\mathcal{G}$. The generator $\mathcal{G}$ can be implemented using the Kriging algorithm or a data-driven generative model that learns to minimize the RMSE between the generated output $\hat{\mathbf{X}}$ and the expected output $\mathbf{X}$. ML models can lead to lower interpolation errors in real test environments if the training dataset is representative. The generator used in our experiments is a U-Net deep neural network implemented by Locke *et al.* [10]. Figure 2 illustrates the architecture, which uses convolutional layers to process the inputs $\mathbf{X}_{\text{med}}$ and $\mathbf{M}$ as 32×32 pixel images. The model training script, written in PyTorch, runs for 50 epochs with the Adam optimizer at a learning rate of 7×10^{-4} and uses batches of 16 maps generated with ray tracing. The performance of REM interpolation models is not the focus of this paper, since the proposed coverage area estimation is agnostic to the specific model architecture. Therefore, information about the dataset and the training parameters can be found in [10].

The coverage mask $\mathbf{C}$ and the corresponding coverage area are computed for each timestamp from $\hat{\mathbf{X}}$. Since the coverage map is built from an estimation of the REM, it differs from the real coverage map that would be constructed if the system had access to the exact power at every bin. Therefore, we compare the estimated coverage mask with the real one, computed using the baseline REM $\mathbf{X}$, following the exact same methodology. Two metrics are used to evaluate how close the estimated coverage region is to the baseline. The first is the coverage area error, defined as

$$\text{CAE} = \left| \frac{\text{Real area} - \text{Estimated area}}{\text{Real area}} \right|,$$

which approaches zero as the estimated coverage area approaches the real one. CAE does not indicate over- or underestimation and can exceed 100%. The second metric is the Intersection-Over-Union (IOU), which is typically used to compare bounding box positions in object detection systems [15]. IOU $= 0$ if the coverage masks do not intersect, and IOU $= 1$ if they are identical.

TABLE II: Experiment 1 summarized results for $V = 10$.

δ (dBm)	CAE 95% C.I. range	IOU 95% C.I. range
-110	$0\% - 14.4\%$	$0.67 - 0.71$
-120	$1.85\% - 10.2\%$	$0.69 - 0.73$
-130	$11.3\% - 28.9\%$	$0.71 - 0.75$

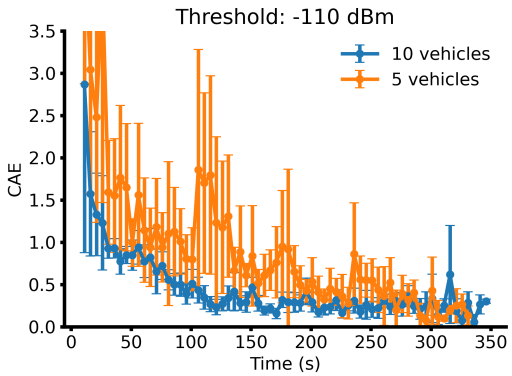
TABLE III: Experiment 2 summarized results for Grid A.

f (MHz)	P (dBm)	CAE 95% C.I. range	IOU 95% C.I. range
1500	25	$3.90\% - 8.20\%$	$0.56 - 0.60$
1500	20	$2.80\% - 11.86\%$	$0.61 - 0.65$
2500	25	$0\% - 7.74\%$	$0.61 - 0.65$
2500	20	$19.53\% - 30.13\%$	$0.31 - 0.37$

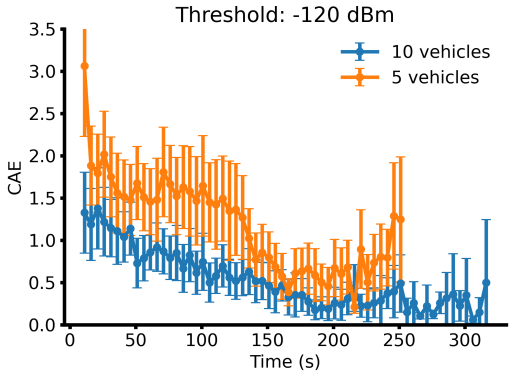
V. Results

The first experiment evaluates real maps for different numbers of vehicles ($V \in \{5, 10\}$) and thresholds ($\delta \in \{-110$ dBm, -120 dBm, -130 dBm$\}$). The fixed parameters are $T = 5$ s, $f = 2500$ MHz, and $P = 25$ dBm. At each experimental repetition, a Python script randomly selects one of the ROIs from Table I, which represents real regions in Brazil with very diverse characteristics.

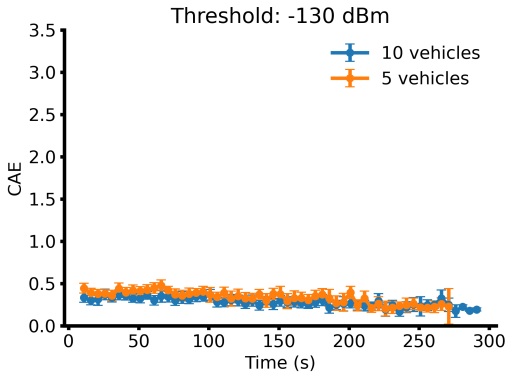
Figure 4 presents the results of Experiment 1, confirming that the coverage area error (CAE) decreases as the vehicles explore the real maps. With only 5 vehicles, the limited number of visited bins within the ROI impairs the interpolation quality in some repetitions, increasing confidence intervals, represented by error bars. Each experimental run finishes after all vehicles exit the ROI, explaining why experiments with more vehicles take, on average, more time to finish. Consequently, late-time points average over fewer runs, widening the intervals. When $\delta = -130$ dBm, both the estimated and the



(a)

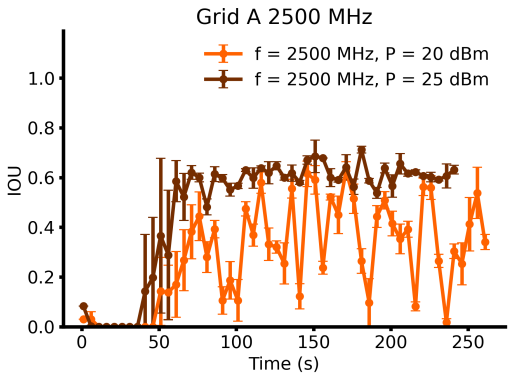


(b)

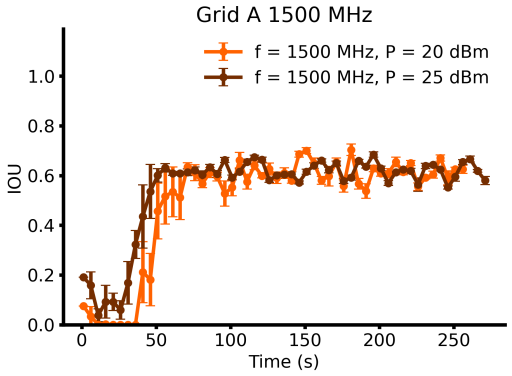


(c)

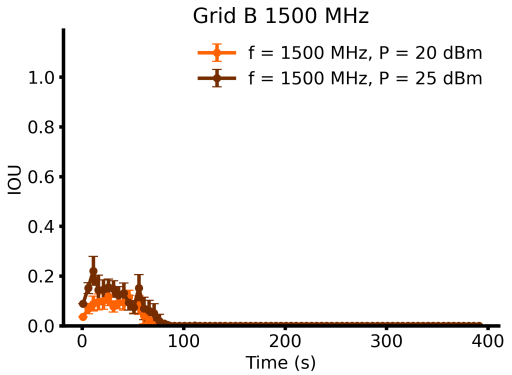
Fig. 4: Experiment 1 results. Coverage area error (CAE, as a fraction) for $f = 2500$ MHz and $P = 25$ dBm.



(a)



(b)



(c)

Fig. 5: Experiment 2 results. IOU for $V = 10$ vehicles and $\delta = -120$ dBm.

ground-truth coverage masks occupy almost the entire terrain from the beginning, leading to a lower CAE from the start.

Table II summarizes the range of the minimum CAE and IOU values for each coverage threshold when $V = 10$. As δ decreases, the coverage area increases, since more bins will have an RSRP above the threshold, which also leads to a higher IOU. On the other hand, the out-of-coverage area decreases, so that the number of samples collected in out-of-coverage bins is significantly reduced. This hinders the U-Net's ability to make good RSRP estimations in these regions. Therefore, the strategy does not significantly improve the coverage mask estimation as time passes, which explains the higher CAE values when δ decreases. For $\delta = -130$ dBm, CAE is the highest but so is IOU: CAE compares only area magnitudes, whereas IOU measures spatial overlap, so both metrics are needed.

The second experiment considers synthetic maps in which city blocks form a grid, as illustrated in Figure 3. The

experiment also evaluates the influence of different values of frequency ($f \in \{1500, 2500\}$ MHz) and transmission power ($P \in \{20, 25\}$ dBm), while keeping fixed $T = 5$ s, $V = 10$ vehicles, and $\delta = -120$ dBm.

Figure 5 presents the results of Experiment 2, confirming that the estimated coverage mask approaches the ground truth for Grid A. Grid B purposely represents a scenario where a large obstacle, such as a building, hinders data collection close to the base station, leading to a null estimated coverage mask. This is a failure mode of the method, detectable online (empty mask), and should flag insufficient data. It is also an extreme scenario, since in the real world, a mobile device would not be able to connect to a base station with such a weak signal. The scenario with $f = 2500$ MHz and $P = 20$ dBm combines the lowest transmission power with the highest signal attenuation due to the higher frequency. In this scenario, the orange curve in Figure 5a confirms the difficulty of obtaining a good estimate of the coverage area.

Table III summarizes the results for Grid A. The CAE confidence intervals statistically overlap in all scenarios except the one with lower power and higher attenuation (last row), which also has the worst IOU. Thus, only the combination of lowest power and highest frequency clearly degrades the estimation.

VI. Conclusion

This work presented a novel approach to continuous radio spectrum coverage estimation by combining a U-Net deep learning interpolation model with a binary coverage mask generation algorithm to process sparse crowdsensing data. By integrating a ray-tracing tool with SUMO to capture the temporal dynamics of data collection, the proposed system achieved coverage area estimation errors below 30% after just 300 seconds in simulation. The evaluation investigated how participant density, geographical map shapes, and radio-frequency parameters impact the intersection-over-union of the estimated masks. Results are limited to simulations with exact RSRP and positions, uniform participation, and no Kriging baseline or BWS ablation.

As future work, we expect to train the U-Net using maps generated by our simulation framework and compare its performance with Kriging and other interpolation techniques on real collected data, including measurement noise and location errors. Another possible research direction consists of training a model to directly generate the coverage mask from sparse measurements.

Acknowledgment

This work received technical support and research funding from the Center for Advanced Studies in Digital Communications and Technological Innovations (Ceadi) of the National Telecommunications Agency (Anatel), through a Decentralized Execution Agreement established between the regulatory agency and the Federal University of Rio de Janeiro, within the scope of Project No. 4/2024 (53500.060216/2024-35), "Artificial Intelligence (AI) solutions for spectrum management."

This work was partially funded by the National Institute of Science and Technology (INCT) STREAM, with support from the National Council for Scientific and Technological Development (CNPq)—grant no. 409179/2024-8, as well as CNPq projects 310234/2025-5, 407304/2025-8, 408255/2023-4, and 405940/2022-0; the Coordination for the Improvement of Higher Education Personnel – Brazil (CAPES) – Funding Codes 001, 88887.954253/2024-00, and 88881.987122/2024-01; FAPERJ grants E-26/200.438/2026 and E-26/210.778/2025; and FAPESP grants 2023/00673-7 and 2023/00811-0.

References

[1] Agência Nacional de Telecomunicações (Anatel), "Portaria nº 2453 — Procedimentos de Fiscalização." [Online]. Available: https://informacoes.anatel.gov.br/legislacao/procedimentos-de-fiscalizacao/1724-portaria-2453

[2] International Telecommunications Union, "Handbook National Spectrum Management," International Telecommunications Union, Geneva, Tech. Rep., 2015.

[3] Agência Nacional de Telecomunicações (Anatel), "Resolução nº 548, de 22 de dezembro de 2010," 2010. [Online]. Available: https://informacoes.anatel.gov.br/legislacao/resolucoes/2010/47-resolucao-548

[4] CEPT/ECC, "CEPT Document 1280." [Online]. Available: https://docdb.cept.org/download/1280

[5] Y. Zeng, B. Liu, Y. Li, D. Giustiniano, and S. Banerjee, "Sustainable Spectrum Crowdsensing," *IEEE Transactions on Networking*, vol. 34, pp. 2954–2965, 2026. [Online]. Available: https://ieeexplore.ieee.org/document/11328871

[6] E. Testi and A. Giorgetti, "Wireless network analytics for the new era of spectrum patrolling and monitoring," *IEEE Wireless Communications*, vol. 31, pp. 230–236, Aug. 2024.

[7] R. C. Bastida, "Medição de Desempenho de SMP Utilizando Funções do Android," trabalho de conclusão de curso, Universidade Federal do Espírito Santo (UFES).

[8] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using sumo," in *21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 2575–2582.

[9] Y. Teganya and D. Romero, "Deep Completion Autoencoders for Radio Map Estimation," *IEEE Transactions on Wireless Communications*, vol. 21, no. 3, pp. 1710–1724, Mar. 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9523765

[10] W. Locke, N. Lokhmachev, Y. Huang, and X. Li, "Radio Map Estimation with Deep Dual Path Autoencoders and Skip Connection Learning," in *2023 IEEE 34th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Sep. 2023, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/10293748

[11] A. Basere and I. Kostanic, "Cell coverage area estimation from receive signal level (rsl) measurements," in *proceedings of the world congress on Engineering and Computer Science*, vol. 1, 2016, pp. 1–7.

[12] K. Sato, K. Inage, and T. Fujii, "On the Performance of Neural Network Residual Kriging in Radio Environment Mapping," *IEEE Access*, vol. 7, pp. 94 557–94 568, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8763960

[13] C.-H. Lin, S.-C. Lin, C.-Y. Wang, and T. Chase, "A c-v2x platform using transportation data and spectrum-aware sidelink access," in *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2021, pp. 1293–1298.

[14] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 32–39, 2011.

[15] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.