A Deep Split-Step Wavelet Model for the Long-Range Propagation

Thomas Bonnafont, Benjamin Chauvel, Abdelmalek Toumi

A deep split-step wavelet model for the long-range propagation

Thomas Bonnafont*, Benjamin Chauvel†, Abdelmalek Toumi*,
*Lab-STICC, UMR CNRS 6285, ENSTA Bretagne, 29806 Brest, France, e-mail: thomas.bonnafont@ensta-bretagne.fr
†ENSTA Bretagne, 29806 Brest, France, email: benjamin.chauvel@ensta-bretagne.org

***Abstract*—This article presents a new approach based on a deep-learning method using a U-Net architecture to generate electromagnetic propagation over a specific terrain. For this purpose, the learning dataset is constructed artificially using a fast split-step wavelet (SSW) method. For this phase, the synthetic 1D profiles are randomly generated from rectangle and triangle shapes. This latter allows for conveying the "staircase" model used in SSW. To ensure a precise sampling of the underlying manifold, the study employs Latin Hypercube Sampling. To achieve robust and precise predictions, a specific loss function is proposed. To evaluate this approach, numerical tests are realized. These tests demonstrate the effectiveness of the proposed method in realistic terrain.**

***Index Terms*—electromagnetic propagation, split-step, wavelet, Deep Learning, U-Net**

I. Introduction

Fast and accurate modeling of electromagnetic (EM) propagation is very important for many applications, such as for the 5G network design or radar coverage prediction. Fast methods such as the knife-edge model [1] are widely used in this context to obtain a prediction. Nevertheless, the accuracy of the latter is low when accounting for real-life terrain. Thus, more complex models have been proposed, such as the one based on the surface integral equation [2], or on the parabolic wave equation [3], [4], [5], or ray-tracing-based models [6], [7], which trade the computation time for better accuracy.

Recently, machine-learning-based methods have received a lot more attention from the propagation community [8]. Indeed, the latter introduces a framework that could lead to achieving near real-time prediction, as a knife edge model, while keeping a good accuracy. A large part of these models is based on an artificially created dataset to overcome the cost of real-life measurements [8], [9], [10].

For example, for indoor propagation, machine learning methods based on the U-Net [11] architecture have been introduced and have shown promising results [9], [10]. In particular, one can cite the Deep Ray model [9], where an artificially constructed dataset using a ray launching method has been used for the training.

The context of atmospheric propagation over rural terrain has also been considered, with artificial data being created with simple propagation models [12], or one where the machine-learning is used to accelerate part of the method [13]. Nevertheless, recent models where the data have been created with an accurate propagation model have shown interesting results for path loss prediction [10]. In the aforementioned models, the samples have been created to "look like" real-life terrain, which leads to accurate results for reliefs with matching shapes.

In this article, we introduce a machine-learning method that allows for predicting path loss from an input terrain profile. The latter is based on a U-Net architecture since it has shown very good results for other related applications [9], [10]. Here, the dataset is created using a parabolic wave equation model solved using the split-step wavelet method [4], [5]. Unlike previous work, we constructed the 1D terrain profiles using only rectangle and triangle shapes to convey the underlying staircase model used in SSW, resulting in a more versatile framework. The Latin Hypercube Sampling (LHS) [14] is used for precise sampling of the underlying manifold.

II. The underlying model

A. Notation and discretization

From now on, an $\exp(j\omega_0 t)$ time variation is assumed for the field, with $\omega_0 = 2\pi f_0$ the angular frequency. We work in the Cartesian coordinate system (x, z) with x the propagation direction and z the altitude. The refractive index, denoted by n, is assumed to be slowly varying along the propagation direction.

The overall computational domain is of maximum range $x_{\max}$ and $z_{\max}$. Besides, we want to compute the field above the ground, thus $z \geq 0$. Furthermore, we assume the source is placed at $x_s \leq 0$ and that the field is known at $x = 0$. Therefore, we have $(x, y) \in [0, x_{\max}] \times [0, z_{\max}]$.

For numerical reasons, a discretization is performed along x and z, with respective steps Δx and Δz. In addition, we denote by $u_x[\cdot]$ the reduced field u at position x discretized along z.

B. The parabolic wave equation model

The parabolic wave equation (PWE) [3] corresponds to a reduction of the Helmholtz equation where only the forward propagation in a paraxial cone is considered. This allows for a wider mesh size along the propagation direction and efficient computation. In this work, we use the wide-angle PWE [3] with $45°$ validity cone around the x-axis. This latter is given by

$$\frac{\partial u}{\partial x} = -jk_0 \left(\sqrt{\frac{1}{k_0^2}\frac{\partial^2}{\partial z^2} + 1} - 1 \right) u - jk_0(n-1)u, \quad (1)$$

where u denotes the reduced field [3], and k_0 the free-space wave number.

Note that the PWE model is particularly adapted to our goal since the effect of the refraction, the ground composition, and the relief can be accounted for.

C. Overview of split-step wavelet

In this section, we briefly introduce split-step wavelet (SSW) [4], [5], which is an efficient computational scheme to solve (1).

Indeed, by going back and forth in the wavelet and spatial domain, SSW efficiently computes the field iteratively. A propagation step from x to $x + \Delta x$ is performed as follows

$$u_{x+\Delta x} = \mathbf{RLW}^{-1}\mathbf{PC}_{V_s}\mathbf{W}u_x, \tag{2}$$

where $\mathbf{W}$ denotes the discrete wavelet transform (performed via the fast wavelet transform), $\mathbf{C}_{V_s}$ a compression with a hard-threshold V_s operator, $\mathbf{P}$ the wavelet-to-wavelet propagator [5], and $\mathbf{R}$ and $\mathbf{L}$ the operators that account for the relief and the refraction, respectively. The latter is performed using a phase-screen [3]. Furthermore, the relief is accounted for using the staircase model [3]. In addition, $\mathbf{P}$ is a sparse operator that computes the free-space propagation in the wavelet domain [5]. Finally, the ground composition is taken into account using the local image method [4].

To conclude this part, SSW allows us to efficiently construct the artificial data needed for the training of the machine-learning method. Besides, numerous phenomena can be introduced such as the relief, the ground composition, or the refraction.

III. Deep-SSW

In this section, we introduce the proposed machine learning model, based on a U-Net architecture, that aims at predicting the path loss for given inputs (such as relief or ground composition). The latter is then trained with the artificial SSW dataset, and thus called from here on Deep-SSW. For this version, only the effect of the terrain is taken into account, but the framework can be generalized to account for other phenomena, such as the ground composition or the transmitter altitude. It should be noted that the input and output are 1D vectors, but a 2D generalization is the object of a future paper.

A. The deep learning architecture

The objective of the machine-learning model is to approximate the function f that maps the input physical phenomena, denoted by ϕ, to a path loss prediction y by $\hat{f}_\theta : \phi \mapsto \hat{y}$. The latter depends on the weights θ of the network, with $|\theta|$ the dimensionality of the network.

Here, ϕ contains terrain data, i.e., the altitude at each discrete point along x, leading to $\phi \in \mathbb{R}^{N_x}$, and thus $y \in \mathbb{R}^{N_x}$ corresponding to the path loss at the transmitter altitude. The function f, which would be SSW, allows us to obtain y, the real path loss, from ϕ such that $y = f(\phi)$. Thus, by optimizing θ, we want to obtain $\hat{y} = f_\theta(\phi)$ as close as possible to y, for a given loss function. This latter is described in detail in Section III-B.

In this work, a U-Net architecture [11] is used, since it has shown good results for related topics [9], [10]. The proposed architecture is shown in Figure 1. It consists of two main parts: the feature detection, the descending part, and the field construction, i.e., a regression, the ascending stage.

a) Feature detection: The first part intends to extract useful features from the input vector. Here, the latter goes through 5 encoding levels of convolution, batch normalization, ReLU activation, and sub-sampling to identify the essential information. In our framework, the convolution kernel is set to k. To add information without increasing the kernel size, and the number of parameters to optimize, we use atrous (or dilated) convolution, as in [9]. It consists of artificially increasing the kernel window by adding 0 (spacing) between the weights. The parameter of kernel widening is called dilation rate and is denoted by dr here. This latter is amplified throughout the feature detection part to capture details further away for lower scales, as shown in Figure 1.

b) Field construction: For the second part, the network aims at constructing the field from the features it has identified in the first stage. It consists of 5 decoding levels, where we go from the lowest scale, details about the relief, to the highest scale through transpose convolution, convolution, batch normalization, and ReLU activation window. Here the kernel for the convolution is of size 2. Furthermore, at each stage of the construction, up-sampling is performed using copied parts of the skipped connection and the output of the transpose convolution. At the end a convolution with a size 1 kernel returns the desired path loss prediction.

This architecture reminds us of the SSW method. Indeed, the operation $\mathbf{C}_{V_s}\mathbf{W}$ and $\mathbf{W}^{-1}\mathbf{P}$ could be seen as the downward and upward stages. By doing so, we can explain the number of stages that correspond to the levels of the wavelet decomposition in SSW.

B. The objective function

Now that the architecture has been introduced, we need to optimize the weight θ of the network. To do so we first need to define the objective loss function.

In the context of regression, two main loss functions are used. The first one is the mean square error (MSE), or L^2-norm error. Indeed, this has the advantage of leading to accurate point-to-point descriptors, since the square magnified the outliers. Besides, the latter is usually used in the electromagnetic community as a measure of error. Nonetheless, to have a good point-to-point accuracy, the output path loss could lose on smoothness. Therefore, one can think of using the L^1-norm, which leads to smoother models at the cost of point-to-point accuracy.

In our work, we propose to combine both the previous cost functions. The proposed empirical risk to minimize is thus defined as

$$\inf_{\theta \in \mathcal{R}^{|\theta|}} \mathop{\mathbb{E}}_{\phi \sim \mathcal{L}_\phi} \left(\alpha \| f(\phi) - \hat{f}_\theta(\phi) \|_1 + \beta \| f(\phi) - \hat{f}_\theta(\phi) \|_2^2 \right), \tag{3}$$

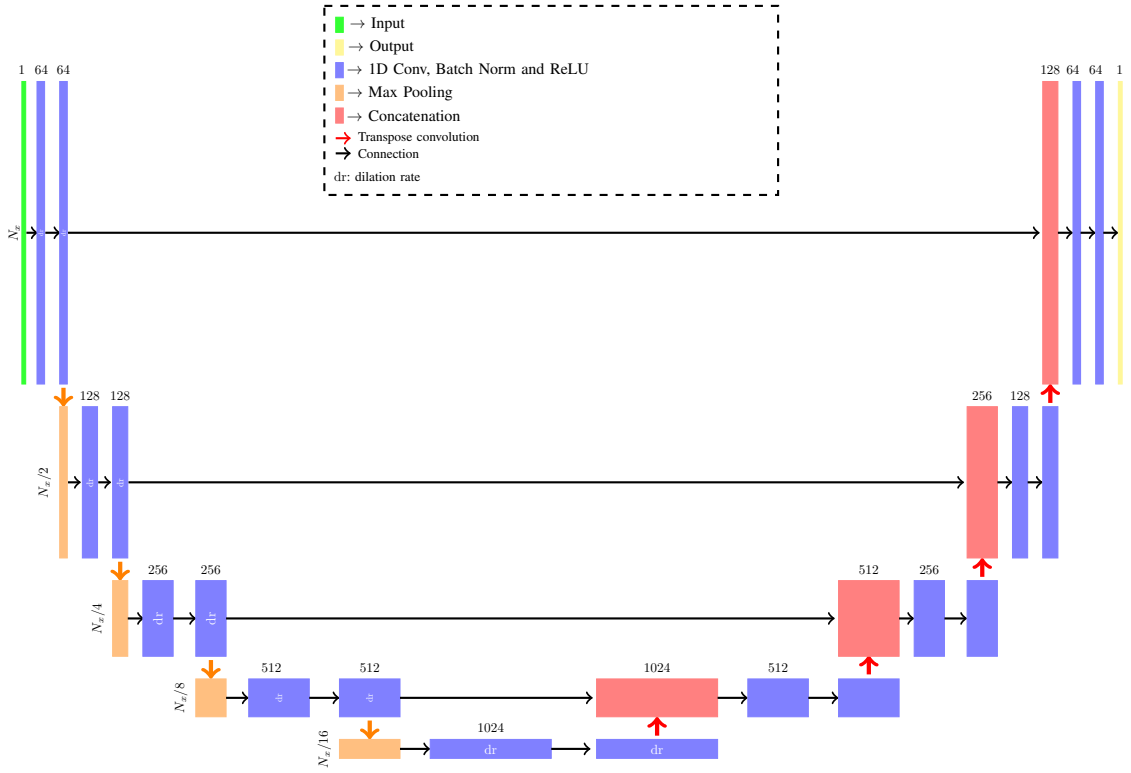

Fig. 1. Architecture of the proposed U-Net based network for path loss prediction.

where $\alpha \in [0,1]$ and $\beta \in [0,1]$ are parameters to adjust to obtain the maximum of the advantages of both the L^1 and L^2 loss functions, i.e., a smooth descriptor with good point-to-point accuracy. Here, we denote by $\mathbb{E}$ the mean value over the samples ϕ. In equation (3), $\mathcal{L}_\phi$ corresponds to the distribution of the samples, described in Section III-C. Finally, the problem (3) is solved using a usual Adam optimizer [15] for the stochastic gradient descent[1]

C. Data and sampling strategy

In this section, the method used to obtain the distribution of the samples $\mathcal{L}_\phi$ is described. Ideally, we want the latter to cover the manifold of the physical phenomena while being also as little as possible for computational efficiency.

An LHS [14] sampling strategy is used here to ensure good coverage of the physical phenomena's manifold. The idea is to create different obstacles – triangles and/or rectangles – of different heights, widths, and at different positions. This follows the underlying staircase model used for the terrain ϕ in SSW. This sampling strategy has the advantage of being versatile since it can easily be generalized either for introducing more obstacle shapes or more physical phenomena.

In more detail, we create 1000 samples for terrain containing from 2 to 5 different obstacles, leading to a total of 4000 equidistributed samples in terms of the number of obstacles. Focusing on the rural environment, the height's extent of each obstacle is in $[0, 65]$ m. Furthermore, the path loss is computed over 80 km with $N_x = 1600$. Since supervised training is used, the labels are the true fields, i.e. u, in dB at the altitude of the transmitter. A sample example, consisting of the terrain, ϕ, and the true field y, is plotted in Figure 2.

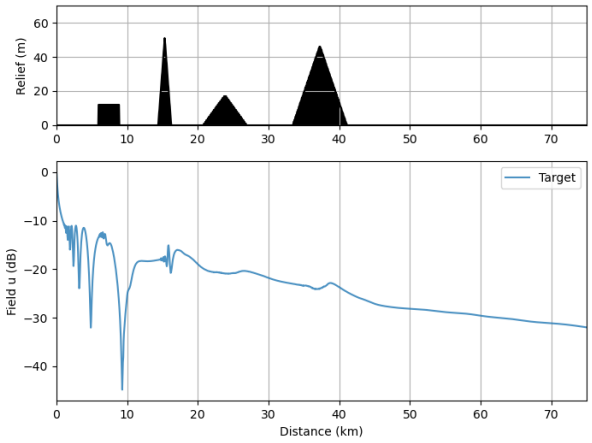

Fig. 2. Example of a sample for the dataset. The terrain and the corresponding field (u in dB) are plotted.

Now, to train the network, the dataset needs to be divided into training and test datasets. Here, we use 80% for the training and the remaining data (20%) as the test dataset.

[1]To accelerate the training procedure, the bias of the last layer is set at first to the mean value of the field over the samples.

IV. Numerical experiments

This section is devoted to numerical simulations. The objectives are first to optimize the network, then to validate the training process, and finally to validate and evaluate the method on real input terrain data.

A. Optimization of the network and training parameters

In this part, we optimize the hyperparameters of our network. In particular, we first focus on setting α and β to obtain the best of the linear combination of the MAE and MSE. Second, the kernel size k of the convolution blocks and the dilation rate dr are jointly optimized to obtain a good accuracy while keeping a low training time.

a) Setting α and β: In this paragraph, we try to optimize the loss function, i.e. to jointly set α and β to obtain a smooth estimator with good point-to-point accuracy. To do so, we train the network with different pair (α, β). Then, we compute the MSE of the different networks over a dataset consisting of 200 samples when 6 reliefs are considered, and also for the propagation over a real-life terrain (obtained from IGN data [16]). In the first case, all MSEs are of the same order 0.2 for all the networks, except the one trained with the MAE, which is at 0.35. In the second case, we plot the MSE computed for all the pairs in Fig. 3.

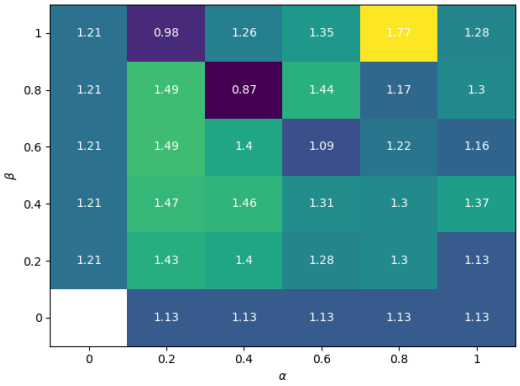

Fig. 3. MSE computed for the propagation over a real (IGN data) landform for different α and β.

From Fig. 3, one can see that the combination $\alpha = 0.4$ and $\beta = 0.8$ leads to the best accuracy in terms of MSE. Besides, except for some pairs, the network is quite robust since the error is of the same magnitude. Nonetheless, in the following, the optimized Deep-SSW corresponds to the one with $(\alpha, \beta) = (0.4, 0.8)$.

b) Choosing the kernel size and the dilation rate: Second, we focus on finding the best choice in terms of accuracy and training time for the kernel size and dilation rate for the convolutions. The same method is used here, i.e. we train different networks with different pairs and compare them in terms of MSE. The kernel size varies in $k \in \{3, 5, 7\}$, leading to 10.8 millions, 15 millions and 19.2 millions of training parameters for each network, respectively. The dilaration rate is set to $\mathrm{dr} \in \{1, 3, 6, 12, 24, 48\}$. The goal is to find the best duo in terms of both MSE and resource allocation.

First, in Fig. 4, we plot a heatmap of the MSE computed for the different doublet (k, dr), computed over the 6 obstacles dataset.

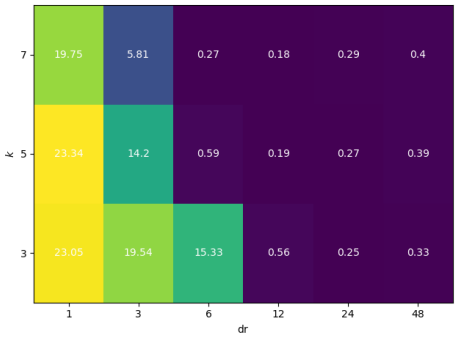

Fig. 4. MSE computed over the 6 obstacles dataset for different k and dr.

In this case, we can see that setting $k = 5$ with $\mathrm{dr} = 12$ gives a very good MSE, close to the one for $k = 7$. Therefore, to keep the number of parameters of the network to optimize, this seems a good choice. To further study the effect of these parameters, we do the same when computing the propagation above an IGN data terrain. This leads to the results given in Fig. 5.

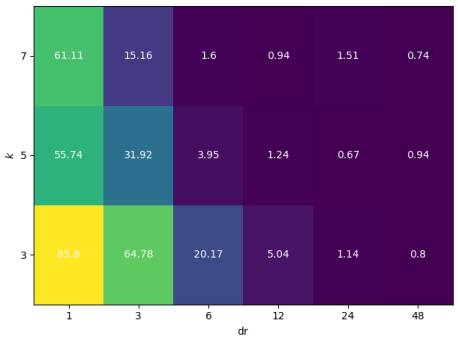

Fig. 5. MSE computed for the propagation over a real (IGN data) landform for different k and dr.

In this case, the same conclusion holds for the kernel size k, which can be set to $k = 5$. Nonetheless, one can see that for a more realistic relief, a higher dr is necessary to have a good accuracy. This result has led us to try to increase the dilation rate over the different levels of the encoder to introduce more information while descending, as in [9], [10]. Therefore, we use an increased dilation rate, which is set at 12 for the first two layers and 24 for the other layers of the encoder. This leads to an MSE of 0.2 on the 6 obstacles dataset and 1.01 on the IGN data, which is as expected between the MSE of a constant $\mathrm{dr} = 12$ and $\mathrm{dr} = 24$, leading to a more versatile predictor.

B. Test with IGN terrain data

In this section, the proposed Deep-SSW method is tested with true IGN [16] terrain data. The idea is to see how accurate the method is for real-life applications. Thus, we compare the machine-learning algorithm to SSW.

In this test, we compute the field from Paris to Chartres, two French cities distant by around 80 km and with a usual rural terrain elevation. Nonetheless, to be in between the altitude bounds of the LHS sampling, we normalize the IGN profile to 60 m at most. A staircase model is then applied to the latter. The terrain and the computed fields with SSW and its Deep version are plotted in Figure 6. Besides, for Deep-SSW, we set $k = 5$ and $\mathrm{d}r$ increasing from 12 to 24 through the encoder stage, as mentioned in the previous section.

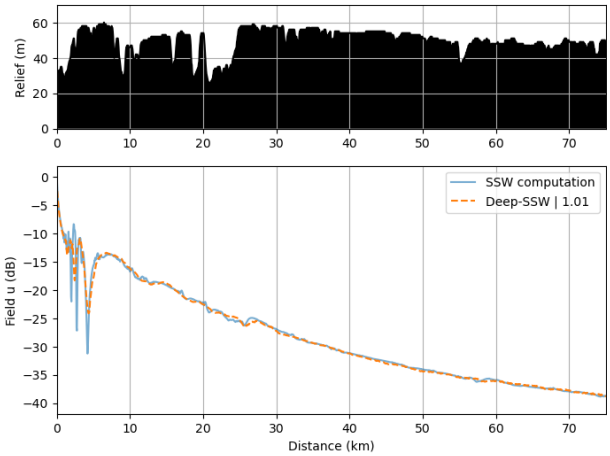

Fig. 6. Propagation from Paris to Chartres computed with both SSW and Deep-SSW. The Deep-SSW is plotted in orange dotted line. The MSE is also given in the legend for reference.

As expected, since the data are completely different from the ones used to train Deep-SSW the MSE $= 1.01$ has increased, compared to the validation test of the previous section, but is still low. Nonetheless, the overall variation of the field has been well retrieved by the machine-learning method. The main errors come from the very low and rapidly oscillating extrema due to the diffraction from the rough part at the beginning.

Finally, the inference time here is below 0.07 s, almost real-time on a conventional desktop computer, whereas with SSW it took around 5 s. Therefore, Deep-SSW allows computing an accurate first glance at the propagation even with real landform data.

V. Conclusion

In this study, we introduced a Deep Split-Step Wavelet model based on the U-Net architecture, which demonstrated strong performance even when applied to real-world landform data.

The main achievement of our work is the elaboration of a very general framework for the sample generation. Indeed, using physical insight into the underlying model, a LHS-based construction of terrain samples has been proposed. The latter is versatile and can very easily handle the introduction of other phenomenons or relief shapes. In addition, a U-Net network with dilated convolution has been proposed to predict the field at the transmitter altitude. To train the network, a dedicated loss function has also been introduced. Finally, numerical tests have shown that the proposed network can be used in place of the knife-edge method, to obtain more accurate field prediction in almost real-time.

Nonetheless, some work still needs to be done. We are currently developing a transfer learning and fine-tuning strategy, which could involve introducing additional outputs or handling different terrains with minimal computational expense. Finally, a generalization to the 2D case, integrating the refraction effects, will be the object of the future paper.

References

[1] J. Deygout, "Multiple knife-edge diffraction of microwaves," *IEEE Transactions on Antennas and Propagation*, vol. 14, no. 4, pp. 480–489, 1966.

[2] J. T. Hviid, J. B. Andersen, J. Toftgard, and J. Bojer, "Terrain-based propagation model for rural area-an integral equation approach," *IEEE Transactions on Antennas and Propagation*, vol. 43, no. 1, pp. 41–46, 1995.

[3] M. Levy, *Parabolic equation methods for electromagnetic wave propagation*. No. 45, IET, 2000.

[4] H. Zhou, R. Douvenot, and A. Chabory, "Modeling the long-range wave propagation by a split-step wavelet method," *Journal of Computational Physics*, vol. 402, p. 109042, 2020.

[5] T. Bonnafont, R. Douvenot, and A. Chabory, "A local split-step wavelet method for the long range propagation simulation in 2D," *Radio Science*, vol. 56, no. 2, pp. 1–11, 2021.

[6] D. Green, Z. Yun, and M. F. Iskander, "Path loss characteristics in urban environments using ray-tracing methods," *IEEE Antennas and Wireless Propagation Letters*, vol. 16, pp. 3063–3066, 2017.

[7] C. B. Findik and Ö. Özgün, "RTPLTool: a software tool for path loss modeling in 5G outdoor systems," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 30, no. 6, pp. 2385–2397, 2022.

[8] A. Seretis and C. D. Sarris, "An overview of machine learning techniques for radiowave propagation modeling," *IEEE Transactions on Antennas and Propagation*, vol. 70, no. 6, pp. 3970–3985, 2021.

[9] S. Bakirtzis, K. Qiu, J. Zhang, and I. Wassell, "DeepRay: Deep learning meets ray-tracing," in *2022 16th European Conference on Antennas and Propagation (EuCAP)*, pp. 1–5, IEEE, 2022.

[10] C. Brennan and K. McGuinness, "Site-specific deep learning path loss models based on the method of moments," in *2023 17th European Conference on Antennas and Propagation (EuCAP)*, pp. 1–5, IEEE, 2023.

[11] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pp. 234–241, Springer, 2015.

[12] M. Ayadi, A. B. Zineb, and S. Tabbane, "A uhf path loss model using learning machine for heterogeneous networks," *IEEE Transactions on Antennas and Propagation*, vol. 65, no. 7, pp. 3675–3683, 2017.

[13] A. Li, C. Yin, and Q. Zhang, "Predicting spatial field values under undulating terrain with 2W-PE based on machine learning," *IEEE Antennas and Wireless Propagation Letters*, vol. 21, no. 2, pp. 222–226, 2021.

[14] M. Stein, "Large sample properties of simulations using Latin hypercube sampling," *Technometrics*, vol. 29, no. 2, pp. 143–151, 1987.

[15] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, (San Diega, CA, USA), 2015.

[16] "Elevation lines data of the "Institut nationale de l'informations Géographique et Forestière" (IGN)." https://www.geoportail.gouv.fr/. Accessed: 03-05-2023.