Machine-Learning-Based Combined Path Loss and Shadowing Model in LoRaWAN for Energy Efficiency Enhancement

Mauricio González-Palacio[1], Diana Tobón-Vallejo[1], Lina M. Sepúlveda-Cano[2], Santiago Rúa[3], and Long Bao Le[4]

[1]Electronics and Telecommunications Engineering Department, Universidad de Medellín, Colombia
[2]Accountancy Department, Universidad EAFIT, Colombia
[3]Electronics Engineering Department, Universidad Nacional Abierta y a Distancia, Colombia
[4]Institut National de la Recherche Scientifique, Canada

***Abstract*—Many practical Internet of Things (IoT) applications require deploying End Nodes (ENs) in hard-to-access places where replacing batteries is difficult or impossible. As a result, the ENs demand high energy efficiency. Long Range Wide Area Network (LoRaWAN) is an IoT protocol that aims to achieve low energy consumption. However, the energy consumption in LoRaWAN is related to transmission power, which can be set mainly based on path loss and shadow fading modeling and link budget analysis. Hence, appropriately setting this transmission power parameter saves energy and guarantees reliable communication links. Traditional path loss and shadow fading modeling and transmission power setting do not consider the variations caused by different environmental effects. In this work, we show via real-life data analysis that path loss and shadow fading depend on environmental variables. We propose Machine Learning models to calculate the empirical path loss and shadow fading, which is used to set the transmission power to save ENs' energy. Our models include the effects of distance, frequency, temperature, relative humidity, barometric pressure, particulate matter, and Signal to Noise Ratio. Specifically, the models are based on Multiple Linear Regression, Support Vector Regression, Random Forests, and Artificial Neural Networks, exhibiting a Root Mean Square Error (RMSE) up to 1.566 dB and R^2 up to 0.94. For energy saving, the developed models serve to set the transmission power and Spreading Factor based on the Adaptative Data Rate (ADR) algorithm principles, which reduces the link margin saving energy up to 43% compared with the traditional ADR protocol.**

***Index Terms*—LoRaWAN, Energy, Path Loss Models, Shadow Fading, Machine Learning, Environmental Variables.**

I. Introduction

THE Internet of Things (IoT) is an Industry 4.0 enabler in charge of collecting, transmitting, storing, and analyzing in-field data in different applications such as smart cities, smart grids, and environmental monitoring, among others [1]. A fundamental part of IoT includes the access networks to connect the End Nodes (ENs) to Internet, where both, communications range and energy, are essential but competing requirements. Furthermore, other requirements like Quality of Service (QoS), security, and flexibility are challenging to meet without affecting energy consumption [2]. Long Range Area Network (LoRaWAN) [3] is a popular Low Power Wide Area Network (LPWAN) technology since it improves the performance related to these requirements with low energy consumption, finding an optimal tradeoff between power and data rate [4]. Thus, the link budget and network planning for LoRaWAN can be performed using traditional radio-frequency techniques, where designers typically aim to achieve communications reliability and energy efficiency. These tasks use theoretical/empirical path loss, shadow fading, and channel models from analytical (Friis [5], Ray-Tracing [6]) or empirical perspectives (e.g., Okumura-Hata [7]), which do not consider the IoT network constraints in general. However, these approaches may not achieve very high accuracy since the complexity of electromagnetic wave propagation in not-guided channels introduces high uncertainties due to the multipath phenomenon [6]. For instance, the Okumura-Hata model was fitted with antennas' heights between 30 and 100 m; nevertheless, common IoT end-node antennas do not reach these heights, as they are typically deployed at ground level [8], and its standard deviation is about 10-14 dB [6], which is inadequate for IoT networks where the transmission powers are $20\,\mathrm{dBm}$ or less [3].

To overcome the limitations mentioned above, active research regarding path loss modeling in LoRaWAN deployments has been done recently. For instance, Anzum *et al.* [9] fitted a Simplified Path Loss Model with Lognormal Shadow Fading (SPLMSF) to characterize the influences of foliage in oil palm crops using LoRaWAN. Furthermore, Alobaidy *et al.* [10] proposed a semi-empirical machine-learning-based path loss model for Long-Range (LoRa) protocol and use it to improve the reliability in water quality monitoring. Batalha *et al.* [11] performed measurement campaigns using LoRa and fitted different semi-empirical path loss models and presented its impact on coverage, SNR, and received packets. Bianco *et al.* [12] fitted a LoRa SPLMSF to use it for localization in mountain environments. Callebaut *et al.* [13] characterized urban, forest, and coastal environments and proposed different path loss models to increase the reliability of LoRa links. Finally, El Chall *et al.* [14] provided different path loss models for indoor, campus, and city scenarios. However, the uncertainty in Combined Path Loss and Shadowing (CPLS) predictions still require including a link margin in the link

budget to guarantee link stability, which entails an extra energy cost [15].

The analytical and empirical approaches surveyed so far assume favorable conditions of stationarity, which are challenging to meet in real deployments [16]. Furthermore, the channel variability is accentuated by some environmental variables like temperature [17], relative humidity [18], barometric pressure [19], rain [20], and pollution [21]. Since the CPLS affects the chosen transmission power, and the transmission power affects the energy consumption, an inaccurate CPLS prediction could waste the ENs' energy since a higher link margin could be overestimated. So, some efforts have been invested in determining such effects on path loss. Particularly, Deese *et al.* [17] proposed a multi-linear model that depends on the temperature and the relative humidity achieving a correlation factor from 0.7 to 0.9 in path loss prediction. In addition, Fang *el al.* [18] showed that high values of relative humidity cause an exponential growth in path loss. Furthermore, Lombardo *et al.* [22] compared the performances of LoRaWAN and NB-IoT technologies in different application scenarios (underground, underwater, and within metal enclosures) and analyzed the effects of the temperature in different path loss models. Moreover, the International Telecommunication Union (ITU) offers the recommendation P676, where the temperature and barometric pressure effects are evaluated [19]. Finally, Gadagkar *et al.* [23] showed different issues in wireless communications, where it is argued that particulate matter can cause scattering in the propagation, affecting path loss estimations. Nevertheless, none of the reviewed proposals have examined the impact of the diverse environmental effects. Therefore, this paper proposes a data-driven model to quantify CPLS in LoRaWAN using Machine Learning (ML) techniques to save energy, bearing in mind the above-mentioned limitations. Our contributions are threefold:

1) We implemented a LoRaWAN experimental setup in an urban environment, including some ENs that embed temperature, relative humidity, barometric pressure, and particulate matter sensors. In addition to these variables, we also collect the Received Signal Strength Indicator (RSSI), Signal to Noise Ratio (SNR), Time on Air (ToA), and Spreading Factor (SF).
2) We developed a set of machine-learning-based environment-aware CPLS models which considers the distance between the ENs and the Gateway (GW), transmission frequency, temperature, relative humidity, barometric pressure, particle matter, and SNR. Our models are based on Multiple Linear Regression (MLR), Artificial Neural Networks (ANNs), Support Vector Regressors (SVRs), and Random Forests (RFs).
3) With the achieved accuracy in the CPLS prediction, we proposed an enhanced Adaptive Data Rate (ADR) algorithm for Transmission Power Control (TPC) in LoRaWAN. With the enhanced ADR algorithm, we can obtain a Packet Delivery Ratio (PDR) greater than 99% with a link margin of $4\,\mathrm{dB}$ for the best path-loss machine learning model. In comparison, the traditional ADR scheme needs a link margin up to $15\,\mathrm{dB}$ to achieve the same PDR [15] (in our experimental setup, it is achieved with a link margin of $11\,\mathrm{dB}$). Thus, we obtained a maximum energy improvement of 43%.

The rest of this paper is organized as follows. Section II shows the experimental setup used to perform the measurements and a preliminary analysis of the collected data where we compared our models versus the traditional CPLS models. Section III provides the CPLS model fittings, such as the MLR model (including the environmental variables) and the ML-based CPLS models (also including the environmental variables). Section IV presents the application of the proposed CPLS models in the enhanced ADR scheme, showing the corresponding numerical results in function of the PDR, ToA, and energy improvements. Finally, Section V presents the conclusions. For ease of reference, the list of important abbreviations in the paper is given in Table I.

TABLE I
ABBREVIATIONS USED IN THIS PAPER

Abbreviation	Description
ADR	Adaptative Data Rate
ANN	Artificial Neural Network
ANOVA	Analysis of Variance
BW	Bandwidth
CPLS	Combined Path Loss and Shadowing
DT	Decision Tree
EN	End Node
GW	Gateway
IoT	Internet of Things
ISM	Industrial Scientific and Medical
ITU	International Telecommunication Union
LoRa	Long Range
LoRaWAN	Long Range Wide Area Network
LOS	Line of Sight
LPWAN	Low Power Wide Area Network
ML	Machine Learning
MLP	Multilayer Perceptron
MLR	Multiple Linear Regression
MQTT	Message Queue Telemetry Transport
MSE	Mean Square Error
NB-IoT	Narrow-Band IoT
NS	Network Server
OMS	Ordinary Minimum Squares
ONNX	Open Neural Network Exchange
QoS	Quality of Service
RF	Random Forest
RL	Return Loss
RMSE	Root Mean Square Error
RSSI	Received Signal Strenght Indicator
SF	Spreading Factor
SNR	Signal to Noise Ratio
SPLMSF	Simplified Path Loss Model with Lognormal Shadow Fading
SVM	Support Vector Machine
SVR	Support Vector Regressor
ToA	Time on Air
TPC	Transmission Power Control
TTN	The Things Network
VNA	Vector Network Analyzer
VSWR	Voltage Standing Wave Ratio

II. EXPERIMENTAL DATA ANALYSIS AND DESIGN GOALS

This section shows the experimental setup used to collect the data for further CPLS analysis and presents a raw data analysis. Moreover, we summarize the behavior of different conventional path loss and shadowing models. Finally, we present the corresponding design goals.

A. Experimental Setup

Figure 1 shows the system architecture we employed to obtain the measurement data, which are used to train the machine-learning-based CPLS models. We have deployed four static ENs in an urban environment and located a GW guaranteeing Line of Sight (LOS) between the ENs and the GW. The GW was linked to The Things Network (TTN), a Network Server (NS) for LoRaWAN. Then, we used the Message Queue Telemetry Transport (MQTT) broker provided by TTN to transport the collected data to a MySQL database server. The details of the implementation are described in the following.

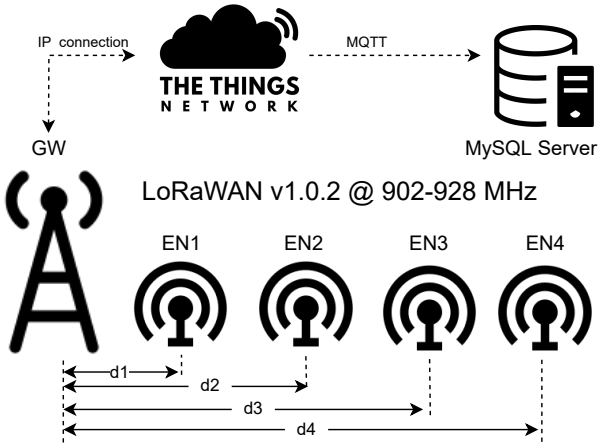
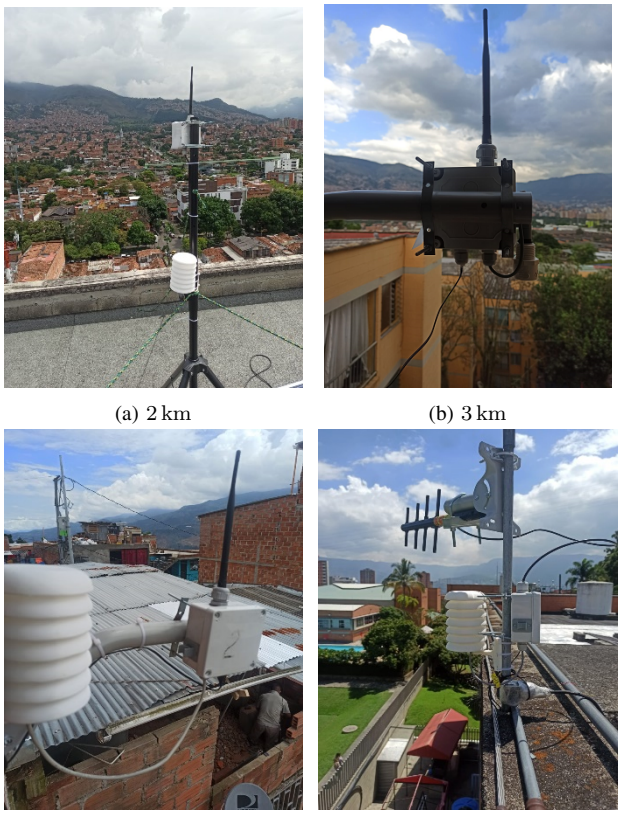


Fig. 1. System model with ENs, GW, NS, and Data Base (DB) server. The ENs send the information to the GW via LoRaWAN, and it retransmits the information to the NS. The NS exposes an MQTT broker that is used to retrieve and store the data on a DB server.

1) End Nodes: We have implemented and deployed four LoRaWAN ENs connected to a GW. Our ENs are based on the Pycom LoPy4 platform, including an ESP32 microcontroller and a Semtech SX1276 LoRaWAN radio. We performed our measurements in the Industrial, Scientific, and Medical (ISM) band of 902-928 MHz and set the transmission power of the LoPy4 to 20 dBm. Three ENs have an omnidirectional antenna (Mobile Mark ref. PSKN3-900) with a peak gain of $3\,\mathrm{dBi}$, and the other EN has a 4-elements Yagi Uda antenna (Pulse Larsen ref. YA6900W) with a peak gain of $8.8\,\mathrm{dBi}$. We checked the S11-S21 parameters for all the antennas, cables, and connectors using a Vector Network Analyzer (VNA). These parameters allowed to calculate the Voltage Standing Wave Ratio (VSWR) and the Return Loss (RL). We used this parametrization to calculate the link budget in each radio link accurately. The LoRaWAN testbed is shown in Figure 2.

2) Sensors: To capture the variations of the weather, we connected a set of sensors to each EN, as follows: *i)* an Aosong DHT22 sensor for temperature (accuracy: $\pm 0.5\,^{\circ}\mathrm{C}$) and relative humidity (accuracy: $\pm 2\,\%$), *ii)* a Bosch BMP280 sensor for barometric pressure (accuracy: $\pm 1\,\mathrm{hPa}$), *iii)* a Honeywell HPMA115S0 sensor for particulate matter of particles with sizes less than $2.5\,\mu\mathrm{m}$ (PM2.5) (accuracy: $\pm 15\,\%$). In addition, we included a Texas Instruments INA219 energy sensor (accuracy: $\pm 0.5\,\%$) in the printed circuit board to quantify the energy consumption under different radio configurations. We added a Stevenson screen to protect the sensors from rainy conditions without losing accuracy or avoiding possible sensor saturations. We selected all the sensors with digital communications to diminish the uncertainty of analog-to-digital conversions. Figure 3 shows pictures of the ENs.

Fig. 2. System LoRaWAN testbed. It includes the embedded system (LoPy4 and SX1276), the thermohygrometer (DHT22), the barometer (BMP280), the PM2.5 sensor (HPMA115S0), and the energy sensor (INA219).



(a) 2 km (b) 3 km

(c) 6 km (d) 8 km

Fig. 3. Deployment of ENs at different distances to the GW. All the locations have LOS to the GW.

3) Gateway: We installed a Dragino LG308 GW to receive the information from the ENs. This GW incorporates a Semtech SX1257 and a Semtech SX1301 LoRaWAN radios, with a sensitivity of $-140\,\mathrm{dBm}$. Once the GW receives the information from the ENs, it resends the data to the NS TTN via Ethernet. Besides, we installed two-panel antennas (brand Wilson Electronics ref. 311155) with a peak gain of $4.4\,\mathrm{dBi}$.

We also checked the S11 parameter for both antennas using a VNA to get an accurate experimental CPLS measurement.

4) Logged Information: Each EN sends data of temperature (10 bit), relative humidity (10 bit), barometric pressure (14 bit), PM2.5 (12 bit), PM10 (12 bit), and energy (16 bit) to the server. The total frame length is 74 bits $\approx$ 10 bytes. To capture the behavior of the different radio configurations, we have combined the transmissions in various frequencies in the band of $902 - 928$ MHz with different SFs in the bandwidth of 125 kHz. Furthermore, we also added dummy bits to the frame to assess the impact of the frame size. In that way, each EN sends information in 215 different radio configurations. We configured the sample time in each node according to the maximum airtime of 400 ms, which is obtained when SF equals 7, frame length equals 242 bytes, and bandwidth is set to 125 kHz [3]. According to the duty cycle policies [3], each transmission must consume 1% of the time and wait for 99% to resend the information. In that way, our sample time is $400\,\text{ms} \times 99 \approx 40\,\text{s}$. Once the data arrived at the NS, we recorded RSSI, SNR, ToA, and frame size. TTN exposes an MQTT broker, so we coded an MQTT consumer to get all the information and save it into a MySQL database. To perform further analyses, we used in-field data from October 2021 to March 2022. Our database has around one million registers, including all previously mentioned variables.

5) Network Deployment: We deployed the ENs at different distances, as shown in Figure 1. The nodes were installed in Medellín, Colombia. Medellín is located in the Central Mountain Range of the Andes and has tropical weather with two rainy and two dry seasons [24]. In the dry season, it rains nine days per month, and in the rainy season, it rains 24 days per month on the average. The PM2.5 increases from dry to rainy seasons [25]. Regarding the locations, we searched places where the ENs and the GW achieve LOS propagation conditions. The coordinates of each device in the network and its corresponding distances (*d*), antenna heights (*h*) and altitudes (*Alt*) are shown in Table II.

TABLE II
LOCATIONS, DISTANCES, ANTENNA HEIGHTS OF THE ENS

Device	d [m]	**Lat** [°]	**Long** [°]	h [m]	Alt [m]
GW	N/A	6.2700	−75.5479	5.0	1699
EN1	2140	6.2654	−75.5664	40.0	1476
EN2	3450	6.2748	−75.5785	12.5	1494
EN3	6100	6.3164	−75.5764	8.0	1723
EN4	8260	6.2320	−75.6117	12.0	1566

B. Conventional Path Loss Models

First of all, we fitted the conventional CPLS models and checked their performance. It is because the determination of the CPLS when planning an LPWAN is fundamental to selecting the transmission parameters (e.g., transmission power) using a link budget calculation, given by Eq. (1)

$$P_T \geq L_T - G_T + P_L(d, f, \ldots) - G_R + L_R + LM + S, \quad (1)$$

where P_T is the transmission power, L_T is the loss in the transmitter associated with connectors and cables, G_T is the gain of the transmitter antenna, P_L is the path loss, d is the distance, f is the frequency, G_R is the gain of the receiving antenna, L_R is the loss in the receiver associated with connectors and cables, LM is the chosen link margin, and S is the receiver sensitivity. Since the losses and gains related to cables, connectors, and antennas are fixed once the node is deployed, the transmission power P_T can vary with the P_L variations because the channel exhibits the shadow fading phenomenon [6]. In particular, the LM compensates for possible inaccuracies in the link budget. However, the excess or defect of the estimation of this parameter will cause a waste of power whenever it is overestimated or connection problems whenever it is underestimated. Some conventional models to estimate the path loss are Friis [5], Two-ray [6], and Okumura-Hata [26]; however, these models do not capture the shadow fading phenomenon. In that way, the SPLMSF model is a combination of a path loss and a lognormal shadowing term given by Eq. (2)

$$PL_{SPLMSF} = -K + 10\gamma \log_{10}(d/d_0) + \psi, \quad (2)$$

where $K = 20\log_{10}\left(\frac{\lambda}{4\pi d_0}\right)$ is a dimensionless constant that depends on the characteristics of the antennas and the average channel attenuation, λ is the wavelength, d is the distance, d_0 is the far-field distance, γ is the path loss exponent, and ψ is a random variable from a lognormal distribution, which characterizes the shadow fading.

We tested different models with the field data and show the obtained results in Figure 4 and Table III. Figure 4 shows that some models, like Friis and SPLMSF (with γ = 2.7 and K = 84.2), are close to the experimental average path loss. In contrast, others like Two-ray and Okumura-Hata models are not accurate enough due to the antennas' heights and the high variability of these models [6]. On the other hand, it can also be seen that the field measurements can be deviated from the mean path loss due to shadowing. Furthermore, the determination factor R^2 [27] reaches 0.8243 (in the case of the SPLMSF), so this model can only capture 82.43% of the shadowing effect. The consequence of having a low R^2 is that it will be necessary to use a higher LM to reach an acceptable PDR; however, an overestimated LM increases the energy consumption.

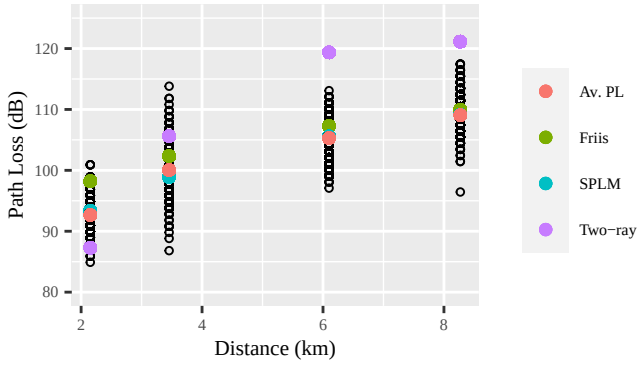


Fig. 4. Conventional path loss models versus distance.

TABLE III
PERFORMANCE OF CONVENTIONAL PATH LOSS MODELS

Model	RMSE (dB)	R^2
Friis	4.00	0.8242
Two-ray	10.97	0.8042
Okumura-Hata	39.69	0.8056
SPLMSF	2.66	0.8243

C. Design Goals

It can be noticed from Table III that although the models' Root Mean Square Error (RMSE) is generally low compared with the average path loss obtained from the measurements, the R^2 is low if the goal is to lower the value of the LM in Eq. (1) and then save energy. So, our first design goal is to propose a CPLS model that allows lowering the LM and the P_T as described by Eq. (3)

$$CPLS = f(d, f, T, RH, BP, PM, SNR) + \psi, \quad (3)$$

where T is the temperature, RH is the relative humidity, BP is the barometric pressure, PM is the particulate matter, and ψ is the shadow fading term. We have also included the SNR in our model since the receiver sensitivity in a LoRaWAN radio link depends on this metric [28] and is given by Eq. (4),

$$S = -174 + 10 \cdot log_{10}(BW) + NF + SNR_{limit}, \quad (4)$$

where S is the receiver sensitivity (in dBm), BW is the bandwidth (in Hz), NF is the noise figure (in dB), and SNR_{limit} is the worst SNR that the receiver can tolerate to demodulate the received signal reliably (in dB). Because our CPLS model aims to reduce the transmission power in the link budget, we considered this variable in the same way that the ADR algorithm [29] and another recent path loss model approach [12].

Furthermore, our second design goal is related to the application of the proposed CPLS model to save energy in LoRaWAN. To this end, we propose a TPC based on the principles adopted by the ADR.

III. ML-BASED COMBINED PATH LOSS AND SHADOW FADING MODELS

This section presents a set of machine-learning-based models (parametric and nonparametric) that improve the RMSE and R^2 when predicting CPLS. These improvements will allow us to save energy when these machine-learning-based CPLS models are employed for TPC, as we will show in Section IV.

A. Environment-Aware CPLS models

The process to fit the CPLS models is depicted in Figure 5. For all the techniques, we performed the following steps: *i)* collect the measurement data from the experimental setup, *ii)* remove the outliers generated by eventual saturations or wrong readings from sensors, and *iii)* divide the resulting database into two subsets for training (80 %) and for testing (20 %), using the library $caTools$ of RStudio. The subsets were created using a fixed seed to ensure the reproducibility of the model fitting. The details of individual machine-learning based CPLS models will be presented in the following subsections.

1) Multiple-Linear-Regression-based CPLS model: Multiple Linear Regression is a modeling strategy used to capture the relationship between a response variable y and a set of predictor variables $\{x_1, ..., x_m\}, (m > 1)$ [27] given by Eq. (5):

$$y = \beta_0 + \sum_{i=1}^{m} \beta_i x_i, \quad (5)$$

where β_0 is an independent parameter, and β_i are the weights of the corresponding predictor variables. Besides, the Ordinary Minimum Squares (OMS) technique determines the optimal values for the weights β_i. An Analysis of Variance (ANOVA) can be carried out to ensure the statistical significance of each predictor variable [27]. Finally, it is also necessary to perform some goodness-of-fit tests to guarantee that the residual errors are normal [30], uncorrelated [31], and homoscedastic [32]. This analysis ensures that the model weights are unbiased and optimal. In our case, we propose the MLR model in Eq. (6):

$$\begin{aligned} P_L =& \beta_0 + 10 \cdot \gamma \cdot \log_{10}(d) + 20 \cdot \log_{10}(f) + \beta_1 \cdot T + \\ & \beta_2 \cdot RH + \beta_3 \cdot BP + \beta_4 \cdot PM + \beta_5 \cdot SNR + \psi, \end{aligned} \quad (6)$$

where β_0 is the model intercept (equivalent to K in Eq. (2)), γ is the path loss exponent, and β_i with $i \in \{1, .., 5\}$ are the model weights. The path loss exponent is multiplied by ten from Eq. (2), assuming a far-field distance $d_0 = 1\,\mathrm{m}$. The constant for the frequency term is fixed to 20 from the Friis model [5]. In addition, we performed the ANOVA to ensure the significance of all the predictor variables and modeled the PDF of the shadow fading term (Appendix A), as shown in Figure 5.

2) Artificial-Neural-Network-based CPLS model: Artificial Neural Networks are a set of bio-inspired algorithms that can be used for either classification or regression purposes. The ANNs achieve good performance by emulating the interactions of neurons inside a brain. Regarding ANN architectures, the Multilayer Perceptron (MLP) is one of the most popular architectures [33]. In the case of regression, a set of input variables x_i (distance, frequency, temperature, humidity, pressure, PM2.5, and SNR) are used to predict the value of the output variable y (CPLS) as depicted in Figure 6. It is essential to point out that the magnitudes of the input variables are in different orders, e.g., frequency (in Hertz) is given in magnitudes of 10^8, while distance (in km) is given in magnitudes of 10^3. So, to obtain the maximum regression accuracy, we have used a standard scaler (Figure 5) to have the input variables centered on the corresponding mean μ_i, and with a standard deviation s_i using the expression $x_{is} = (x_i - \mu_i)/s_i$, where x_{is} is the resulting variable after scaling. Besides, the best ANN configuration is achieved by selecting the optimal number of neurons in k different layers. In the case of MLP, all the neurons in layer h_j are fully connected to the neurons in the previous layer h_{j-1} and the neurons to the next layer h_{j+1}. The effect of each connection is weighted by a set of coefficients W determined by an optimization method (normally gradient descend algorithm). This nonparametric regression method does not need to state

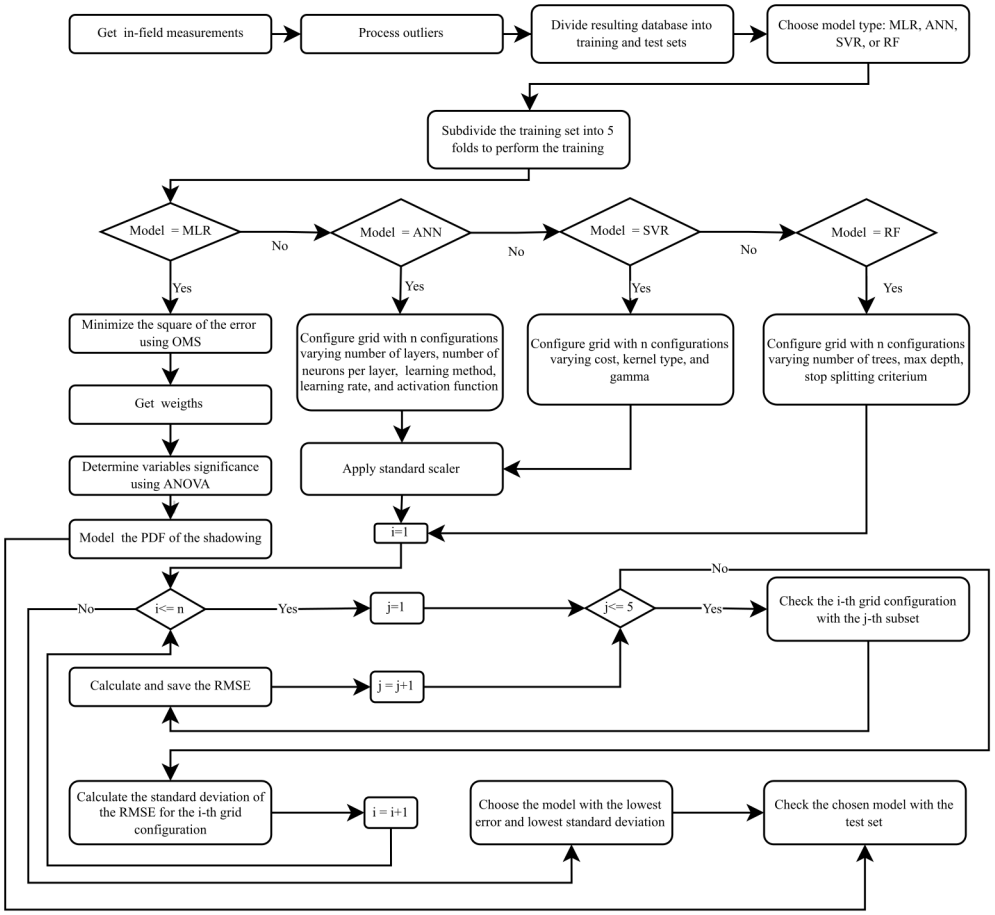


Fig. 5. Fitting process for the MLR, ANN, SVR, and RF based CPLS models.

a priori assumptions between the input and output variables and can model data in nonlinear processes [34]; that is, they do not need any previous knowledge about the process, for instance, the probability distribution of the residuals, contrary to the parametric methods like MLR.

To train the ANNs, we used the library Scikit Learn (`sklearn`) from Python [35], particularly the class of MLP for regression, `MLPRegressor`. We used a laptop Dell Latitude 3410 (Intel Core i7 10^{th} generation, 16 GB RAM, solid-state hard disk 1 TB). The hyperparameters we considered for the ANN training were: *i)* the ANN architecture (i.e., the number of hidden layers and the number of neurons per layer), *ii)* the regressor learning rate schedule for weight updates (*constant*, *invscaling*, and *adaptative* [35]), *iii)* the regressor learning rate α when the constant schedule is chosen, and *iv)* the activation function (Figure 5).

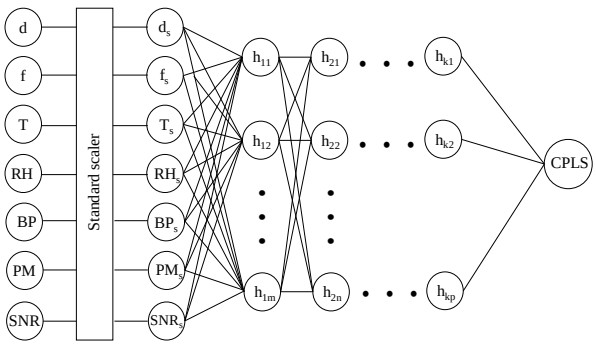


Fig. 6. ANN-based CPLS model. Our model has seven input neurons (previously scaled) and one output neuron.

3) Support-Vector-Regression-based CPLS: Support Vector Regressions are a set of algorithms for regression based on Support Vector Machines (SVMs). In the case of classification, the idea behind SVMs is to separate a collection of previously tagged classes by finding the corresponding hyperplane W that maximizes the margin, i.e., the separation among the classes [36]. However, in many cases, the classes are not linearly separable; for this reason, it is recommended to use kernel functions [37] to perform a transformation that increases the number of dimensions, allowing the separation of such classes. The main advantage of the SVR is that it can deal with sparsity, non-linearity, and high dimensionality of the input data [38]. The SVR hyperparameter tuning is performed through cross-validation, and finally, the configuration that exhibits the best results will be chosen. The most notable drawback of SVRs

is that the training time depends on the number of samples and scales as $O(n^3)$ [39]. Some previous references of CPLS models using SVRs can be found in [40]. However, there are no previous references for LoRaWAN nor consider the effects of environmental variables.

On the other hand, the SVR training is similar to the ANN, including the same input features and the same CPLS output. Nevertheless, we used a processing cluster (16 nodes with octa-core processors Intel(R) Xeon(R) CPU E5-2680 0 @ 2.70GHz, and 256 GB RAM) since the computational complexity of the SVR training is $O(n^3)$, and our training set has 792.600 rows. As we previously explained, we included a scaler for the input features. We also configured a grid search to explore what hyperparameter's configuration achieves the best performance regarding RMSE and standard deviation in the cross-validation using the training set, considering cost, kernel, and gamma hyperparameters (Figure 5). Then, we used the test set to check the ability of the SVR to preserve the same performance in the presence of unknown data.

4) Random-Forest-based CPLS: The Random Forest is a generalization of the Decision Trees (DTs). A DT can be described as a set of nested conditionals, where the conditions are evaluated using the inputs, x_i, and some threshold values v_i [41]. The algorithm evaluates the nested conditionals (nodes) until it finds a final assignment (a leaf) that delivers an average of the subgroup to find the regression value of a new set of inputs. The threshold values v_i are calculated by minimizing a cost function of the dispersion of the corresponding subgroups by an error metric like the Mean Square Error (MSE) [41]. An example of a DT with one feature x_i and three threshold values v_1, v_2, and v_3 is presented in Figure 7. However, the most reported issue of the DTs is overfitting; that is, the model can learn the training data with low bias, but it does not achieve the same performance when unknown data is introduced (causing high variance) [42]. This phenomenon is because the trees are highly dependent on the input data so an imperceptible variation can generate a different tree architecture. This limitation leads to the development of RFs [41]. The idea behind RFs is to train several trees (usually tens or hundreds) with different subsets (conformed using bootstrapping) of the original training set, choosing some input features randomly for each condition in each tree. All the trees' outputs are averaged to get the regression value. Since the training subsets for diverse trees are different, the architectures will also vary, and other trees will compensate for the sensitivity of a particular tree caused by the training set. This characteristic helps preserve low bias and variance (avoid overfitting).

The training process of the RF was similar to the previous methods and ran on the same PC as the ANNs (the same inputs and the same CPLS output). We applied the class `RandomForestRegressor` from the library `sklearn` for Python [35]. The RF algorithm is based on decision rules, so using the standard scaler for the input features is unnecessary. Similarly, we used a grid search method to explore the best configuration over different hyperparameter configurations and used five-fold cross-validation to minimize the effect of overfitting, considering the number of trees, max depth per tree, and stop criterium (Figure 5).

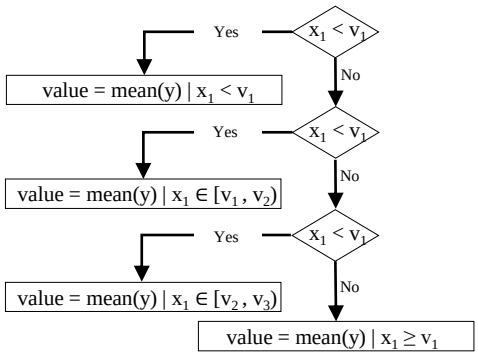


Fig. 7. Random Forest with three decision rules (tree depth = 3) for one variable, x_1, with three threshold values, v_1, v_2, and v_3

B. Numerical results

To evaluate the performances of each CPLS model, we based our analysis on the RMSE, the R^2, and the standard deviation of the RMSE when the hyperparameters of each model were calculated for the cross-validation. The results are shown in Table IV. The following sections will discuss how these results were achieved.

TABLE IV
MODELS' PERFORMANCES

Regressor	Subset	RMSE (dB)	R^2
Friis	Training	4.007	0.8241
	Test	4.002	0.8246
Two-ray	Training	10.977	0.8040
	Test	10.981	0.8049
Okumura-Hata	Training	39.6951	0.8054
	Test	39.7115	0.8064
SPLMSF	Training	2.512, $s = 0.013$	0.8311
	Test	2.66	0.8243
MLR	Training	1.951, $s = 0.00129$	0.905
	Test	1.951	0.905
ANN	Training	1.706, $s = 0.135$	0.935
	Test	1.613	0.935
SVR	Training	2.021, $s = 0.428$	0.9376
	Test	1.626	0.9342
RF	Training	1.919, $s = 0.339$	0.9458
	Test	**1.566**	**0.9389**

1) MLR-based CPLS model: After applying the model in Eq. (6), we obtained an RMSE = $1.95\,\mathrm{dB}$ and an R^2 = 0.9049 for both the training and test sets. The achieved R^2 outperforms the R^2 (from 0.7 to 0.9) presented in previous works [17]. The model weights for the proposed model are shown in Table V.

Qualitatively, the model weights can be interpreted as follows: *i)* distance exhibits a positive effect on path loss, that is, when the distance increases, the path loss also increases, which attends the theoretical models; besides, the path loss exponent $\hat{\gamma} = 2.205$ meets the expected values for microcells [6], *ii)* the effect of temperature is positive, that is, higher temperatures cause high CPLS, which attends to what other proposals have found (e.g., in [17]), *iii)* relative humidity also has a positive impact on CPLS (e.g., [17]), *iv)* barometric

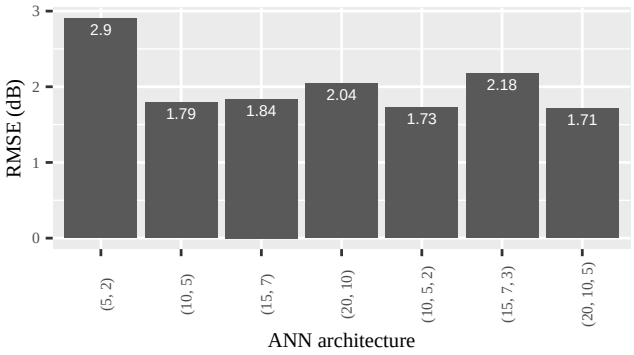
TABLE V
MODEL WEIGHTS FOR THE MLR MODEL

Weight	Variable	Value
$\hat{\beta}_0$	Intercept	-431.03 dB
$\hat{\gamma}$	Distance	2.205 dB
$\hat{\beta}_1$	Temperature	$0.0859\,\mathrm{dB/°C}$
$\hat{\beta}_2$	Rel. humidity	$0.0012\,\mathrm{dB/\%}$
$\hat{\beta}_3$	Bar. pressure	$0.3991\,\mathrm{dB/hPa}$
$\hat{\beta}_4$	PM2.5	$0.000\,222\,\mathrm{dB/\mu g/m^3}$
$\hat{\beta}_5$	SNR	$-0.6236\,\mathrm{dB}$

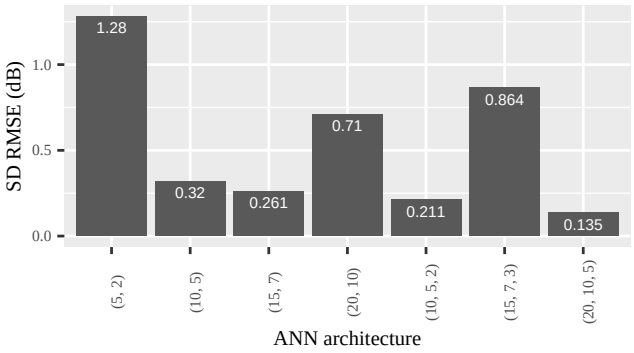
pressure effect is positive, so as the pressure increases, the CPLS also increases (higher pressure implies that there is more water vapor concentration that acts as a scatterer, attenuating the signal [19]), and *v)* the SNR has a negative effect on CPLS, i.e., when the SNR is lower, the CPLS increases [12]. After inspecting the model's ANOVA, we found that all the predictor variables are significant (see Appendix A). We also found that the shadow fading may be modeled more accurately using a t-Student distribution (see Appendix A).

2) ANN-based CPLS model: To fit the model in Figure (6), we chose different configurations similar to those employed in [43]. This fit includes the ANN architecture for the hidden layers, that is, the numbers of layers and neurons per layer needed to obtain adequate performance. It is suggested to use two or three hidden layers progressively descending the number of neurons. In that way, we used the following configurations (indicated using a tuple representation): *i)* $(5, 2)$, *ii)* $(10, 5)$, *iii)* $(15, 7)$, *iv)* $(20, 10)$, *v)* $(10, 5, 2)$, *vi)* $(15, 7, 3)$, and *vii)* $(20, 10, 5)$. Concerning the regressor learning rate for the method 'constant,' we considered learning rates of 0.0001 (default value) and 0.05 (augmenting the regularization penalty in the loss function). Since we perform a regression task, the recommended activation function is the Rectifier Linear Unit (RELU) [44]. Because we considered seven architectures, three different learning rate methods, and two different learning rates, we checked 42 configurations. Furthermore, we have included a cross-validation process with five folds to minimize the overfitting impact. So, we iterate the same configuration for all training-test subsets, getting a corresponding RMSE. The accuracy of the corresponding configuration will depend on the RMSE (low bias). In contrast, the ability of generalization (low variance) will rely on the standard deviation of the RMSE for the different validation subsets. Finally, as we have 42 configurations for different hyperparameters and five cross-folds, we checked 210 configurations. We achieved the results at convergence after 2000 epochs or tolerance of 10^{-4}. The average cross-validation RMSE values for different ANN architectures are shown in Figure 8 (a). This figure shows the results of the best configurations (Average RMSE = $1.7058\,\mathrm{dB}$) achieved with $\alpha = 0.0001$, constant learning rate, and a setup with three hidden layers, with 20, 10 and 5 neurons. Furthermore, in Figure 8 (b), we presented the corresponding standard deviations of the cross-validation RMSE for the best configurations, finding that the same architecture achieves the lowest value ($SD_{RMSE} = 0.135\,\mathrm{dB}$). It means that the selected architecture has the best capacity for generalization.

Subsequently, we used the selected configuration to calculate the RMSE (and standard deviation s of the RMSE in the case of cross-fold validation) and R^2 for both the training and test sets, obtaining the results shown in Table IV.



(a) RMSE ANN

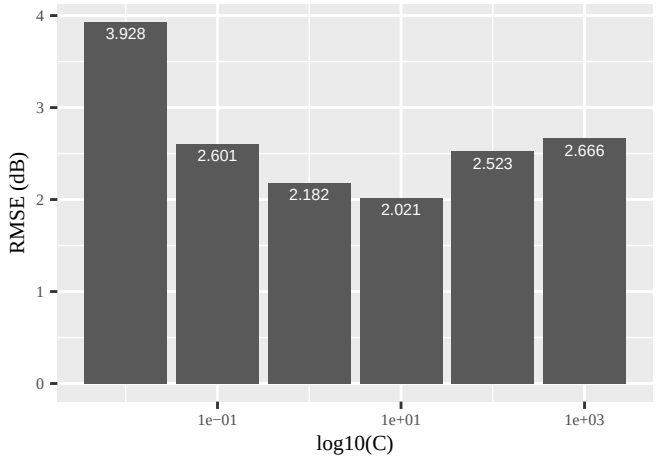


(b) SD RMSE ANN

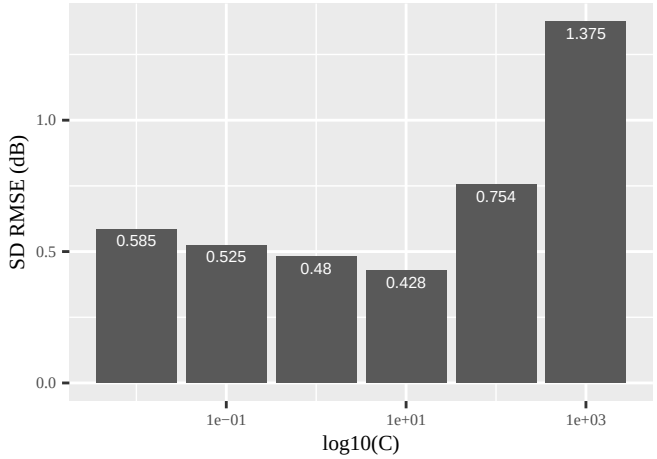
Fig. 8. RMSE and standard deviation for the best ANN configurations. The best fit is achieved with $\alpha = 0.0001$, constant learning rate, and architecture of (20,10,5).

3) SVR-based CPLS model: To fit the model, we considered the following hyperparameters: *i)* cost (C), which serves as a regularization term (taking values of 0.01, 0.1, 1, 10, and 1000); *ii)* type of kernel (Radial Basis Function - RBF and polynomial); and *iii)* gamma (used for the RBF kernel), that defines how accentuated the influence of a single training example is [45] (taking values of 0.001, 0.01, 0.1, and 1). Under these hyperparameter variations, we got 40 different candidates. In addition, we also used the cross-validation technique (five-folds) previously described to prevent overfitting, so we got 40 $\times$ 5 = 200 trainings. The results of the best configurations were plotted in Figure 9 achieved when the kernel is RBF, $C = 10$, gamma $= 0.1$, obtaining an average RMSE $= 2.021\,\mathrm{dB}$, and a standard deviation of $0.428\,\mathrm{dB}$ (see Table IV).

4) RF-based model: To fit our RF regressor, we used the following hyperparameters: *i)* number of trees in the forest: 100, 200, and 300, *ii)* max depth of trees: 6, 9, and 12, *iii)* min number of training observations per leaf to stop splitting: 1, 2, and 4, *iv)* the minimum number of training observations required to split an internal node: 2, 10, 100, and 1000. The results of the best fits are presented in Figure 10, where it can be noticed that the best RMSE ($1.92\,\mathrm{dB}$) and standard deviation of RMSE ($0.339\,\mathrm{dB}$) is achieved when max depth

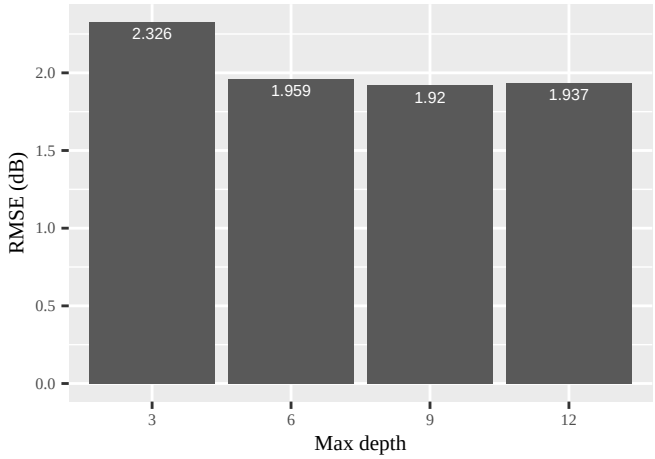


(a) RMSE SVR

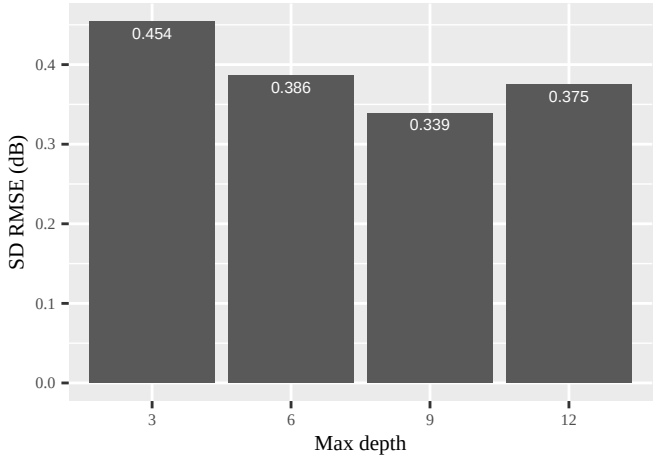


(b) SD RMSE SVR

Fig. 9. RMSE and SD for the best SVR configurations. The best fit is achieved with $C = 10$, gamma $= 0.1$, and the RBF kernel.



(a) RMSE RF



(b) SD RMSE RF

Fig. 10. RMSE and SD for the best RF configurations, achieved with the max depth is 9, min number of samples per leaf is 1, min number of samples to split is 100, and number of trees is 100.

equals 9, min number of samples per leaf is 1, min number of samples to split equals to 100, and the number of trees in the forest equals to 100. The RF performance is presented in Table IV.

C. Discussion

We proposed parametric and nonparametric CPLS models in this section and presented the corresponding RMSE, R^2, and standard deviations for the five-fold cross-validation fitting process. Regarding the parametric method (MLR), we showed that including some environmental variables improves the performance metrics, particularly in the shadow fading prediction, compared with the classic SPLMSF model. We also showed that the shadowing is t-distributed (Appendix A), helping estimate more accurately the LM and reducing the energy consumption, as we will show in Section IV.

However, it is essential to point out that although we concluded that the residuals are not strictly normally distributed and uncorrelated, the model exhibits heteroscedasticity (see Appendix A), so the estimators are unbiased but suboptimal [27]. Because of that, we have used other nonparametric techniques that reduce the error, e.g., ANNs, SVRs, and RFs. We noticed that all the methods have a similar RMSE in the test set and improve the errors of other traditional models. Moreover, the obtained R^2 achieved a noticeable improvement regarding the prediction of the shadow fading term ψ. Furthermore, our results can be contrasted with other machine-learning-based path loss models (VI), reducing the errors and increasing the R^2, as shown in Table VI.

IV. Energy Efficient Design Using Developed ML-based Models

The proposed machine-learning based CPLS models can be used for TPC to achieve improved energy consumption in the ENs. One way to achieve this is by reducing the LM from the link budget task before the deployment of the network. Nonetheless, LoRaWAN includes the scheme ADR that can be used to adjust the transmission parameters dynamically (i.e., the SF and the P_T) by changing the admissible Signal to Noise Ratio (SNR_{limit}), which is a function of the SF. The SF can take values from 7 ($SNR_{\text{limit}} = -7.5\,\text{dB}$) up

TABLE VI
COMPARISON OF OUR ML-BASED CPLS MODELS VERSUS OTHER APPROACHES IN LITERATURE

Algorithm	Objective	Input variables	Error (dB)	R^2	Authors
ANN	Estimate an environment-aware CPLS	Distance, frequency, temperature, relative humidity, barometric pressure, particulate matter, SNR	**RMSE = 1.613**	**0.935**	**Our approach**
	Estimate path loss at different frequencies in rural, urban, and suburban environments	Distance, transmitter and receiver height, frequency, and diffraction loss.	MAE = 0.23	0.852	[46]
	Estimate path loss in indoor scenarios	Transmitter position, transmitter gain, transmitter height, type of interior (corridor, room), distance, number of walls, number of windows	RMSE = 4.23	Not reported	[47]
	Estimate path loss in urban and suburban scenarios	Distance, widths of streets, building heights, building separation, difference between the antenna height and the rooftop height	RMSE = 6.78	Not reported	[48]
	Estimate path loss considering micro-variations on the terrain (caused mainly by trees)	Distance, tree height, and tree width	MAPE = 4.26%	Not reported	[43]
	Estimate path loss in a Wireless Sensor network in a rural environment	Distance, mean percent tree canopy cover, terrain complexity, vegetation variability, source canopy coverage, receiver canopy coverage	RMSE = 5.15	0.51	[49]
RF	Estimate an environment-aware CPLS	Distance, frequency, temperature, relative humidity, barometric pressure, particulate matter, SNR	**RMSE = 1.566**	**0.9389**	**Our approach**
	Estimate path loss in unmanned aerial vehicles environment	Distance, transmitter height, receiver height, path visibility, elevation angle	RMSE = 1.64	Not reported	[50]
	Estimate path loss in a Wireless Sensor network in a rural environment	Distance, Mean percent tree canopy cover, Terrain complexity, Vegetation variability, Source canopy coverage, Receiver canopy coverage	RMSE = 3.72	0.51	[49]
	Estimate path loss model in a rural scenario for air-to-ground communications using unmanned aerial vehicles	Distance, frequency	RMSE = 2.367	0.9292	[51]
SVR	Estimate an environment-aware CPLS	Distance, frequency, temperature, relative humidity, barometric pressure, particulate matter, SNR	**RMSE = 1.626**	**0.9342**	**Our approach**
	Estimate path loss in IEEE802.11 networks	Distance	RMSE = 1.71	0.917	[40]
	Estimate path loss in suburban environments	Distance	RMSE = 6.2	Not reported	[52]
	Estimate path loss in suburban environments	Distance	RMSE = 4.47	Not reported	[53]

to 12 ($SNR_{\text{limit}} = -20\,\text{dB}$) [29], with steps of $-2.5\,\text{dB}$, so higher SF values can tolerate more adverse channel conditions and larger distances. However, high SF values imply that the ToA increases, and the LoRaWAN radio must consume more energy [15]. One popular ADR scheme is proposed by TTN [54] and works as follows,

- First, the network server collects 20 SNR samples and obtains the maximum.
- Then, it selects a link margin LM and calculates the margin excess M_e as,

$$M_e = SNR_{\text{max}} - SNR_{\text{limit}} - LM$$

- Finally, it varies the SF and decreases P_T as needed to get $M_e = 0$.

Nevertheless, the ADR scheme has some drawbacks. First, the LM must be overestimated in highly varying channels, where the shadow fading effects are considerable, to get an acceptable PDR, so it usually takes values from 5 to 15 dB [15] (a significant power excess regarding energy constraints in LPWANs). Second, if the sample rate in the ENs is low, the SNR measurements collected will not capture the channel variability correctly increasing the convergence time [55]. Third, suppose the network server does not receive measurement data from an EN (a packet drop). In that case, the corresponding SNR measurement will not be considered to calculate the transmission parameters, causing a suboptimal solution [15]. So, we propose an enhanced ADR scheme with the following features: *i)* it runs directly in the EN, so the device does not need to get the feedback from the NS, *ii)* it estimates the expected RSSI in the GW at the moment of the transmission, then it captures the current channel state, and *iii)* it allows decreasing the LM term according to the desired PDR.

A. Enhanced ADR Algorithm

Since we can estimate accurate CPLS values in the EN, we can forecast the expected RSSI in the GW, as shown in Algorithm 1. To this end, before each transmission, the EN recalculates the P_L and the link budget (Eq. (1)), and estimates the M_e (lines 3, 4, and 5) as shown in Eq. (7),

$$M_e = RSSI_{\text{estimated}} - (P_{\text{noise}} + SNR_{\text{limit}} + LM). \quad (7)$$

Then, we contemplate two scenarios: *i)* when the current transmission power and SF are insufficient to reach the GW, they should be incremented (lines 6-17), and *ii)* when the current transmission power and SF are overestimated, they should be decreased to save energy (lines 18-31). In the first case, we decrease the SF as needed to guarantee that the M_e is greater than zero, so connectivity is successful (line 8), then we recalculate the M_e with the new SF and adjust the transmission power (lines 13-17). In the second case, the procedure is similar; however, this time, we decrease the SF (line 21) and transmission power (lines 29 and 31) to improve energy consumption. The M_e expression in our approach also includes an LM term (lines 5, 10, and 22); however, it can be lowered since the $RSSI_{\text{estimated}}$ is sensitive to environmental changes.

B. Numerical Results for PDR

We performed simulations to validate the effects of applying the different CPLS models in the PDR, and we also studied the behavior of the conventional ADR algorithm. As we commented in Section III-A, we divided our database into training and test subsets, so we used the test subset to assess the PDR. To this end, we used Python object-oriented programming to simulate the actual behavior of an EN. First, we coded a class that simulated the environmental variables and the SNR (using the test set). Second, we calculated the corresponding path loss/CPLS predictions using the Friis model, SPLMSF (with lognormal shadow fading), SPLMSFT (the letter 'T' indicates that we considered t-distributed shadow fading), MLR (with t-distributed shadow fading), ANN, SVR, and RF. Third, we obtained the P_T and SF parameters using algorithm 1, varying the LM term from 0 to 15 dB [15]. Fourth, we compared the estimated GW RSSI versus the actual GW RSSI provided by the in-field data using different LM values. We considered the following decision rules: *i)* if the actual RSSI was greater than the predicted RSSI, the packet was delivered successfully, and *ii)* if the actual RSSI was less than the predicted RSSI, the packet was dropped. The simulation results are presented in Figure 11.

Several observations can be drawn from Figure 11 as summarized in the following.

1) For the conventional ADR, PDR of 80% is achieved with an $LM = 6\,\text{dB}$, PDR of 85% with $LM = 7\,\text{dB}$, PDR of 90% with $LM = 8\,\text{dB}$, PDR of 95% with $LM = 9\,\text{dB}$, and PDR of 99% with $LM = 11\,\text{dB}$. It agrees with the expected LM reported in [15].
2) If we use the Friis model in algorithm 1, we obtain PDR of 80% with $LM = 0\,\text{dB}$, PDR of 85% with $LM = 1\,\text{dB}$, PDR of 90% with $LM = 2\,\text{dB}$, PDR of 95% with $LM = 3\,\text{dB}$, and PDR of 99% with $LM = 5\,\text{dB}$. At first glance, this result outperforms the other environment-aware CPLS models. However, this behavior is expected since the Friis model tends to overestimate the path loss causing energy waste, as we will show in subsection IV-C. So, the higher PDR is caused by an overestimation in the transmission parameters.
3) Employment of parametric models (i.e., SPLMSF, SPLMSFT, and MLR) in the enhanced ADR scheme outperform the ADR scheme. It is important to remark that the t-distribution improved the PDR since the SPLMSF has a worse behavior than SPLMSFT regarding the PDR. Furthermore, it can also be noticed that the MLR model (using the environmental variables) is even better than the SPLMSFT, so the inclusion of these variables to characterize the shadow fading term ψ helps increase the PDR.
4) The machine learning models (i.e., ANN, SVR, and RF) have a similar behavior regarding the PDR. Particularly, all these models achieve 95% PDR with an $LM = 3\,\text{dB}$ and PDR of 99% with an $LM = 4\,\text{dB}$. This result is similar to that achieved by the Friis model; however, these models do not overestimate the P_L, making them more energy-efficient.

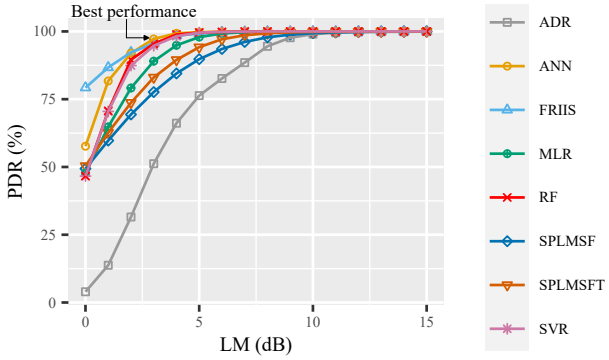


Fig. 11. PDR versus LM for different ADR schemes. The best performance is achieved with the ANN-based CPLS model

C. Numerical Results for Energy

We also evaluated the improvement of our enhanced ADR scheme (using the different CPLS models) compared with the conventional ADR scheme in terms of energy consumption. To this end, we calculated the radio power consumption using the data provided in the Semtech SX1276 datasheet, which shows the consumed current by the LoRaWAN radio when using different transmission powers with a supply voltage = 3.3 V, as shown in Table VII. We transformed the given power to mW. Then we fitted a simple linear regression to get a closed-form expression to forecast the power consumption of different transmission powers (from 0 to 20 dBm), obtaining an $R^2 = 0.95$.

We calculated the energy for each observation by $E = P_T \times ToA$. The ToA is obtained as a function of the SF,

Algorithm 1 Enhanced ADR Algorithm

inputs: d, f, T, RH, BP, PM, SNR, pl_model, current_tp, current_sf, LM, noise_power ▷ Define inputs
parameters: min_sf, max_sf, min_tp, max_tp, ltx, gtx, lrx, grx, snr_limit ▷ Define parameters
estimated_cpls ← cpls_model(d, f, T, RH, BP, PM, SNR) ▷ Calculate CPLS
estimated_rssi ← current_tp - ltx + gtx - estimated_cpls + grx - ltx ▷ Estimate RSSI in the GW
margin_excess ← estimated_rssi - (noise_power + snr_limit + LM) ▷ Calculate the margin excess
if margin_excess $<$ 0 **then** ▷ Current transmission parameters insufficient
    adjusted_sf ← false
    while margin_excess $<$ 0 *and* current_sf $<$ sf_max **do** ▷ Increase SF to guarantee EN/GW link
        current_sf ← current_sf + 1
        margin_excess ← estimated_rssi - (noise_power + snr_limit + LM)
        if margin_excess $>$ 0 **then**
            adjusted_sf ← true
    set_point_tp ← current_tp - margin_excess
    if set_point_tp $\leq$ max_tp **then** ▷ Adjust TP to minimize the margin excess
        current_tp ← set_point_tp
    else
        current_tp ← max_tp
if margin_excess $\geq$ 0 **then** ▷ Current transmission parameters overestimated
    adjusted_sf ← false
    while margin_excess $>$ 0 *and* current_sf $>$ sf_min **do** ▷ Decrease SF to the minimum to guarantee EN/GW link
        current_sf ← current_sf - 1
        margin_excess ← estimated_rssi - (noise_power + snr_limit + LM) ▷ Adjust TP to minimize the margin excess
        if margin_excess $\leq$ 0 **then**
            current_sf ← current_sf + 1
            adjusted_sf ← true
            break
    set_point_tp ← current_tp - margin_excess
    if set_point_tp $\geq$ min_tp **then**
        current_tp ← set_point_tp
    else
        current_tp ← min_tp
output: current_tp, current_sf ▷ Return transmission parameters

TABLE VII
POWER VERSUS CURRENT CONSUMPTION OF LOPY ENS

Power (dBm)	**Current** (mA)
7	20
13	29
17	87
20	120

the bandwidth (125 kHz), the payload size (we normalized the calculation to 1 byte), the coding rate (we used 4/5), the cyclic redundancy check (enabled by default), and the explicit header (enabled by default). The expression to calculate the ToA can be found in [3]. We depict in Figure 12 the ToA improvement compared to the conventional ADR vs. the PDR for the different CPLS models.

Subsequently, we used the results of algorithm 1, whose outputs are the SF and the P_T, to calculate the energy consumed with a given PDR. We used the conventional ADR as the reference to compare the improvements in our approach using all the models. We show the results in Figure 13.

The following observations can be drawn from the results presented in Figures 12 and 13.

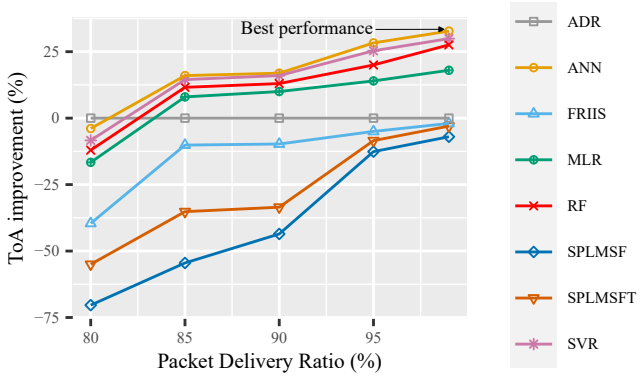


Fig. 12. ToA versus PDR for different ADR schemes. The best performance is achieved with the ANN-based CPLS model

1) It can be noticed that the ToA and energy improvements increase as the PDR is greater. It means that the selection of the LM can be performed more efficiently using our environment-aware ML CPLS models compared to the fixed values used in the conventional ADR in general.
2) However, the numerical results show that the conven-

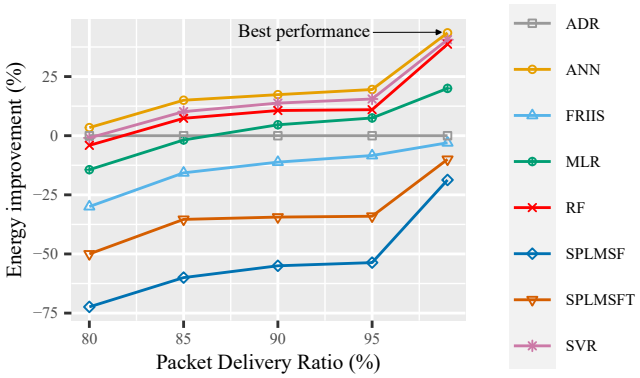


Fig. 13. Energy versus PDR for different ADR schemes. The best performance is achieved with the ANN-based CPLS model

tional ADR scheme obtains better results when PDR = 85%, which means that in non-critical applications, the traditional algorithm can be used to get the optimal transmission parameters.

3) The CPLS models resulting in most energy consumption and ToA are SPLMSF and SPLMSFT. These models are worse regarding energy consumption and ToA than the conventional ADR. This is expected since the ability of these models to predict the shadow fading ψ term is limited, as previously discussed. Besides, the Friis-based ADR scheme is also more energy-consuming than the traditional ADR algorithm, indicating that the good behaviour of PDR in Figure 11 is achieved using higher values of SF and transmission power.
4) The parametric MLR-based ADR improves the traditional ADR for PDRs greater than 90%. In particular, energy improvements up to 20% can be achieved using a computational-unexpensive formula considering the variations exerted by the weather.
5) The nonparametric-model-based ADR (i.e., ANN, SVR, and RF) outperform, in general, the conventional ADR. For instance, when PDR is 99%, the energy improvements are 43.5% for ANN, 40.6% for SVR, and 38.7% for RF, and the ToA improvements are 32.7% for ANN, 29.9% for SVR, and 27.5% for RF. These results are explained by *i)* the inclusion of the environmental variables in the models that can characterize the shadow fading ψ better than the SPLMSF, the SPLMSFT, the Friis model, and the conventional ADR, and *ii)* the ability of ML models to explain processes with non-linearities.

D. Discussion

In this section, we proposed an enhanced version of the ADR scheme for transmission power control using the CPLS estimation to calculate the SF and P_T transmission parameters. In that way, it can be concluded from Figures 11 and 13 that the nonparametric models lead to the best tradeoff between PDR and energy consumption since they obtain a PDR $= 99\%$ with a low LM, so the energy consumption can be improved up to 43% by using the ANN CPLS model.

V. Conclusions

In this work, we modeled the path loss and shadow fading phenomena in LoRaWAN using empirical data in an urban environment. First, we fitted the corresponding measurements versus Friis, Two-ray, and Okumura-Hata models since they are probably the most used approaches for wireless networks' link budget engineering tasks. Because our experimental setups were deployed guaranteeing LOS, the most accurate conventional model in terms of RMSE was the Friis model. Furthermore, we fitted the SPLMSF obtaining good results regarding RMSE but a low R^2. Furthermore, we proposed a set of CPLS models, considering the environment's effects, specifically temperature, relative humidity, barometric pressure, and particulate matter. We demonstrated that these variables impact the path loss and the shadow fading phenomena in the average value and the corresponding standard deviation. After that, we showed that the t-Student distribution fits better the shadow fading ψ term, which helps determine the CPLS more accurately. However, the heteroscedasticity persisted (see Appendix A), so we concluded that the weights are unbiased but suboptimal, so other regressors minimize the RMSE. With this motivation, we fitted a set of ML-based CPLS models (using ANN, SVR, and RF) that outperformed the conventional models and the MLR parametric model, obtaining an RMSE $= 1.57\,\mathrm{dB}$ and an $R^2 = 0.94$. These results allowed us to lower the link margin LM in the link budget stage. Furthermore, we proposed an enhanced ADR algorithm using the proposed CPLS models. We measured its effect on the PDR and energy consumption, finding that the ANN achieves the best compromise between good connectivity and low energy consumption, where a PDR $= 99\%$ is reached with an $LM = 4\,\mathrm{dB}$ with an energy improvement up to 43%.

We showed that the application of ML-based CPLS in an enhanced ADR scheme saves energy compared to the traditional scheme. Our results are based on extensive simulations. In the future, we plan to implement the proposed enhanced-ADR schemes in the ENs, which have constraints regarding processing power and memory size [56]. In that way, deploying the ML-based CPLS models and enhanced ADR schemes can be challenging. However, some active research is being taken to overcome these limitations and run ML models in embedded systems. For instance, an efficient framework called TinyML is being used to run SVR, RFs, and ANNs with low energy consumption, adopted by different vendors like Arduino, ST, Nordic, Raspberry, Nvidia, Espressif, among others [57]. Furthermore, TinyML has been used to deploy ML algorithms in LPWAN oriented to improve battery life, scalability, security, and performance, among others [58]. These implementations are interesting but outside the scope of the current work. Therefore, as future work, we propose the following:

1) We will deploy the RF, SVR, and ANN CPLS models and ADR enhancements in the ENs using existing frameworks like TinyML. Existent tools allow the conversion of `sklearn` models into open models like Open Neural Network Exchange (ONNX), providing portability between cloud-based and embedded platforms [59], so

our offline-trained models can be ported to the ENs.
2) Since our ENs incorporate an energy sensor, we will measure the actual consumption with the enhanced ADR schemes, where we will compare the results achieved by these enhanced designs in comparison with the traditional ADR scheme.

It is important to state that the training of the ML-based CPLS models is carried out offline, that is, in non-constrained devices (e.g., in the cloud), so the embedded system does not carry out the intensive computing charge of this task. Furthermore, the EN only computes the regression prediction using MLR (which is not computationally intensive), ANN [60], SVR [61], and RF models obtained from the training process.[62].

REFERENCES

[1] P. Asghari, A. M. Rahmani, and H. H. S. Javadi, "Internet of things applications: A systematic review," *Computer Networks*, vol. 148, pp. 241–261, January 2019.

[2] B. Mao, F. Tang, Y. Kawamoto, and N. Kato, "Ai models for green communications towards 6g," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 1, pp. 210–247, 2021.

[3] B. S. Chaudhari and M. Zennaro, *LPWAN Technologies for IoT and M2M Applications*. Academic Press, 2020.

[4] Y. Kawamoto, R. Sasazawa, B. Mao, and N. Kato, "Multilayer virtual cell-based resource allocation in low-power wide-area networks," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 665–10 674, 2019.

[5] H. T. Friis, "A note on a simple transmission formula," in *In Proc. IRE*, IEEE, Ed., vol. 34, no. 5, May 1946.

[6] A. Goldsmith, *Wireless communications*. Cambridge university press, 2005.

[7] C. Jiang, Y. Yang, X. Chen, J. Liao, W. Song, and X. Zhang, "A new-dynamic adaptive data rate algorithm of lorawan in harsh environment," *IEEE Internet of Things Journal*, February 2021.

[8] W. Tang, X. Ma, J. Wei, and Z. Wang, "Measurement and analysis of near-ground propagation models under different terrains for wireless sensor networks," *Sensors*, vol. 19, no. 8, p. 1901, April 2019.

[9] R. Anzum, M. H. Habaebi, M. R. Islam, G. P. Hakim, M. U. Khandaker, H. Osman, S. Alamri, and E. AbdElrahim, "A multiwall path-loss prediction model using 433 mhz lora-wan frequency to characterize foliage's influence in a malaysian palm oil plantation environment," *Sensors*, vol. 22, no. 14, p. 5397, 2022.

[10] H. A. Alobaidy, R. Nordin, M. J. Singh, N. F. Abdullah, A. Haniz, K. Ishizu, T. Matsumura, F. Kojima, and N. Ramli, "Low-altitude-platform-based airborne iot network (lap-ain) for water quality monitoring in harsh tropical environment," *IEEE Internet of Things Journal*, vol. 9, no. 20, pp. 20 034–20 054, 2022.

[11] I. S. Batalha, A. V. Lopes, W. G. Lima, Y. H. Barbosa, M. C. Neto, F. J. Barros, and G. P. Cavalcante, "Large-scale modeling and analysis of uplink and downlink channels for lora technology in suburban environments," *IEEE Internet of Things Journal*, 2022.

[12] G. M. Bianco, R. Giuliano, G. Marrocco, F. Mazzenga, and A. Mejia-Aguilar, "LoRa System for Search and Rescue: Path-Loss Models and Procedures in Mountain Scenarios," *IEEE Internet of Things Journal*, vol. 8, no. 3, pp. 1985–1999, March 2021.

[13] G. Callebaut and L. Van der Perre, "Characterization of lora point-to-point path loss: Measurement campaigns and modeling considering censored data," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 1910–1918, 2020.

[14] R. El Chall, S. Lahoud, and M. El Helou, "LoRaWAN network: Radio propagation models and performance evaluation in various environments in Lebanon," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2366–2378, March 2019.

[15] G. G. M. de Jesus, R. D. Souza, C. Montez, and A. Hoeller, "Lorawan adaptive data rate with flexible link margin," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 6053–6061, October 2020.

[16] S. M. Aldossari and K.-C. Chen, "Machine learning for wireless communication channel modeling: An overview," *Wireless Personal Communications*, vol. 106, no. 1, pp. 41–70, May 2019.

[17] A. S. Deese, J. Jesson, T. Brennan, S. Hollain, P. Stefanacci, E. Driscoll, C. Dick, K. Garcia, R. Mosher, B. Rentsch *et al.*, "Long-term monitoring of smart city assets via internet of things and low-power wide-area networks," *IEEE Internet of Things Journal*, vol. 8, no. 1, pp. 222–231, June 2020.

[18] Z. Fang, H. Guerboukha, R. Shrestha, M. Hornbuckle, Y. Amarasinghe, and D. M. Mittleman, "Secure communication channels using atmosphere-limited line-of-sight terahertz links," *IEEE Transactions on Terahertz Science and Technology*, 2022.

[19] I. T. Union, "Attenuation by atmospheric gases and related effects, itur p.676-13," *Recommendation ITU-R*, pp. 676–12, August 2022.

[20] O. Elijah, S. K. A. Rahim, V. Sittakul, A. M. Al-Samman, M. Cheffena, J. B. Din, and A. R. Tharek, "Effect of weather condition on lora iot communication technology in a tropical region: Malaysia," *IEEE Access*, vol. 9, pp. 72 835–72 843, May 2021.

[21] N. A. Shekh, V. Dviwedi, and J. P. Pabari, "Effect of sandstorm on radio propagation model of mars," in *Proc. International Conference on Mobile Computing and Sustainable Informatics*. Springer International Publishing, December 2020, pp. 441–447.

[22] A. Lombardo, S. Parrino, G. Peruzzi, and A. Pozzebon, "LoRaWAN Versus NB-IoT: Transmission performance analysis within critical environments," *IEEE Internet of Things Journal*, vol. 9, no. 2, pp. 1068–1081, May 2021.

[23] A. V. Gadagkar and B. Chandavarkar, "A comprehensive review on wireless technologies and their issues with underwater communications," in *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*. IEEE, 2021, pp. 1–6.

[24] IDEAM, "Características climatológicas de ciudades principales y municipios turísticos," IDEAM, Tech. Rep., 2020.

[25] Á. M. d. V. de Aburrá, "Factores que incrementan la contaminación en el Valle de Aburrá," 2020. [Online]. Available: https://www.metropol.gov.co/ambientales/calidad-del-aire/generalidades/condiciones-especiales

[26] Y. Okumura, "Field strength and its variability in VHF and UHF land-mobile radio service," *Rev. Electr. Commun. Lab.*, vol. 16, pp. 825–873, December 1968.

[27] J. J. Faraway, *Practical regression and ANOVA using R*. University of Bath Bath, 2002, vol. 168.

[28] D. H. Kim, E. K. Lee, and J. Kim, "Experiencing LoRa network establishment on a smart energy campus testbed," *Sustainability (Switzerland)*, vol. 11, no. 7, March 2019.

[29] LoRa-Alliance, "RP002-1.0.1 LoRaWAN® Regional Parameters," 2020. [Online]. Available: https://lora-alliance.org/sites/default/files/2020-06/rp_2-1.0.1.pdf

[30] F. J. Massey Jr, "The Kolmogorov-Smirnov test for goodness of fit," *Journal of the American statistical Association*, vol. 46, no. 253, pp. 68–78, April 1951.

[31] K. J. White, "The Durbin-Watson test for autocorrelation in nonlinear models," *The Review of Economics and Statistics*, pp. 370–373, May 1992.

[32] D. M. Waldman, "A note on algebraic equivalence of White's test and a variation of the Godfrey/Breusch-Pagan test for heteroscedasticity," *Economics Letters*, vol. 13, no. 2-3, pp. 197–200, May 1983.

[33] K. L. Priddy and P. E. Keller, *Artificial neural networks: an introduction*. SPIE press, 2005, vol. 68.

[34] Y. O. Ouma, C. O. Okuku, and E. N. Njau, "Use of Artificial Neural Networks and Multiple Linear Regression Model for the Prediction of Dissolved Oxygen in Rivers: Case Study of Hydrographic Basin of River Nyando, Kenya," *Complexity*, vol. 2020, May 2020.

[35] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, and V. Dubourg, "Scikit-learn: Machine learning in Python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, November 2011.

[36] W. S. Noble, "What is a support vector machine?" *Nature biotechnology*, vol. 24, no. 12, pp. 1565–1567, December 2006.

[37] N. Deng, Y. Tian, and C. Zhang, *Support vector machines: optimization based theory, algorithms, and extensions*. CRC press, 2012.

[38] L.-r. Tian and X.-d. Zhang, "A Convergent Nonlinear Smooth Support Vector Regression Model," in *Proc. International Conference on Industrial Engineering and Engineering Management 2014*. Springer, 2015, pp. 205–207.

[39] H. Guo and W. Wang, "Granular support vector machine: a review," *Artificial Intelligence Review*, vol. 51, no. 1, pp. 19–32, January 2019.

[40] M. González-Palacio, L. Sepúlveda-Cano, R. Montoya, Á. Rocha, C. Ferrás, P. C. López-López, and T. Guarda, "Simplified Path Loss Log-normal Shadow Fading Model Versus a Support Vector Machine-Based Regressor Comparison for Determining Reception Powers in WLAN

Networks," in *Proc. Advances in Intelligent Systems and Computing*. Cham: Springer International Publishing, January 2021, pp. 431–441.
[41] A. Cutler, D. R. Cutler, and J. R. Stevens, "Random forests," in *Ensemble machine learning*. Springer, 2012, pp. 157–175.
[42] T. G. Dietterich and E. B. Kong, "Machine learning bias, statistical bias, and statistical variance of decision tree algorithms," Oregon State University, Tech. Rep., 1995.
[43] Y. Egi and C. E. Otero, "Machine-learning and 3d point-cloud based signal power path loss model for the deployment of wireless communication systems," *IEEE Access*, vol. 7, pp. 42 507–42 517, March 2019.
[44] J. Schmidt-Hieber, "Nonparametric regression using deep neural networks with ReLU activation function," *The Annals of Statistics*, vol. 48, no. 4, pp. 1875–1897, August 2020.
[45] Scikit-learn, "RBF SVM parameters," 2021. [Online]. Available: https://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters
[46] 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, May 2017.
[47] I. Popescu, D. Nikitopoulos, I. Nafornita, and P. Constantinou, "ANN prediction models for indoor environment," in *Proc. IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*. IEEE, September 2006, pp. 366–371.
[48] I. Popescu, A. Kanstas, E. Angelou, L. Nafornita, and P. Constantinou, "Applications of generalized RBF-NN for path loss prediction," in *Proc. The 13th IEEE international symposium on personal, indoor and mobile radio communications*, vol. 1. IEEE, December 2002, pp. 484–488.
[49] C. A. Oroza, Z. Zhang, T. Watteyne, and S. D. Glaser, "A machine-learning-based connectivity model for complex terrain large-scale low-power wireless deployments," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 576–584, December 2017.
[50] G. Yang, Y. Zhang, Z. He, J. Wen, Z. Ji, and Y. Li, "Machine-learning-based prediction methods for path loss and delay spread in air-to-ground millimetre-wave channels," *IET Microwaves, Antennas & Propagation*, vol. 13, no. 8, pp. 1113–1121, April 2019.
[51] S. Duangsuwan and M. M. Maw, "Comparison of path loss prediction models for UAV and IoT air-to-ground communication system in rural precision farming environment," *Journal of Communications*, vol. 16, no. 2, pp. 60–66, February 2021.
[52] K.-C. Hung, K.-P. Lin, G. K. Yang, and Y.-C. Tsai, "Hybrid support vector regression and GA/TS for radio-wave path-loss prediction," in *Proc. International Conference on Computational Collective Intelligence*. Springer, 2010, pp. 243–251.
[53] K.-P. Lin, K.-C. Hung, J.-C. Lin, C.-K. Wang, and P.-F. Pai, "Applying least squares support vector regression with genetic algorithms for radio-wave path-loss prediction in suburban environment," in *Advances in Neural Network Research and Applications*. Springer, 2010, pp. 861–868.
[54] The-Things-Network, "Adaptative Data Rate." [Online]. Available: https://www.thethingsnetwork.org/docs/lorawan/adaptive-data-rate/
[55] A. Farhad, D.-H. Kim, S. Subedi, and J.-Y. Pyun, "Enhanced lorawan adaptive data rate for mobile internet of things devices," *Sensors*, vol. 20, no. 22, p. 6466, November 2020.
[56] T. S. Ajani, A. L. Imoize, and A. A. Atayero, "An overview of machine learning within embedded and mobile devices–optimizations and applications," *Sensors*, vol. 21, no. 13, p. 4412, 2021.
[57] N. Schizas, A. Karras, C. Karras, and S. Sioutas, "Tinyml for ultra-low power ai and large scale iot deployments: A systematic review," *Future Internet*, vol. 14, no. 12, p. 363, 2022.
[58] L. Dutta and S. Bharali, "Tinyml meets iot: A comprehensive survey," *Internet of Things*, vol. 16, p. 100461, 2021.
[59] S. Nakandala, K. Saur, G.-I. Yu, K. Karanasos, C. Curino, M. Weimer, and M. Interlandi, "A tensor compiler for unified machine learning prediction serving," in *14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20)*, 2020, pp. 899–917.
[60] D. Mengistu and F. Frisk, "Edge machine learning for energy efficiency of resource constrained iot devices," in *SPWID 2019: The Fifth International Conference on Smart Portable, Wearable, Implantable and Disabilityoriented Devices and Systems*, 2019, pp. 9–14.
[61] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource constrained edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1205–1221, 2019.
[62] G. Cornetta and A. Touhafi, "Design and evaluation of a new machine learning framework for iot and embedded devices," *Electronics*, vol. 10, no. 5, p. 600, 2021.
[63] Student, "The probable error of a mean," *Biometrika*, pp. 1–25, March 1908.

APPENDIX A
ANOVA AND SHADOW FADING ANALYSIS FOR THE MLR-BASED CPLS MODEL

To ensure the statistical significance of the CPLS model proposed in Section III-A1, we performed the corresponding ANOVA. According to this analysis, all the parameters have statistical significance. It supports the conclusion that the CPLS depends statistically on environmental variables. Moreover, to ensure that the set of estimators β_i and the path loss exponent γ are unbiased, the model residuals (i,e., the shadow fading term ψ) must fulfill some goodness-of-fit metrics: normality, independence, and homoscedasticity. To address the normality assumption, we ran the Kolmogorov Smirnov test finding no normality in the distribution of the residuals (p-value $= 2.2 \times 10^{-16}$). To understand the type of probability density function, we inspected the QQ plot (see Figure 14) and found that the model exhibits fat tails. We also applied the Durbin Watson test (DW = 1.67), finding that the variables are uncorrelated. Finally, we applied the Breusch Pagan test to check homoscedasticity (p-value $= 1 \times 10^{-16}$), indicating that there is heteroscedasticity.

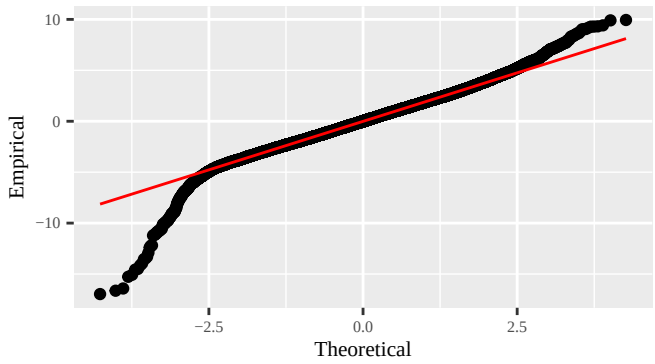


Fig. 14. Quantile-Quantile plot of residuals for the MLR CPLS model. It can be noticed that the model residuals exhibit fat tails.

The fat tails on residual errors physically mean that the shadow fading values can be more prone to be farther from the mean than the values predicted by the normal distribution. It implies that the shadow fading term in Eq. (6) should be carefully modeled to avoid overestimation or underestimation, leading to extra energy consumption or connectivity problems when the LM in Eq. (1) is not appropriately chosen. Finding a probability density function that adequately addresses the fat-tailed behavior while preserving the low-skewness characteristics is mandatory. In fact, a t-Student distribution [63] meets these needs. This distribution arises when some samples are taken from a normally distributed process, but it is not possible to establish the Z-statistic to determine the corresponding confidence intervals since the standard deviation σ of the population is unknown. This case is accentuated specifically in measurement campaigns for determining empirical CPLS models when there are not enough samples to ensure that the population and the sample standard deviations are the same. Thus, the suitable way to determine this statistic is by using the sample standard deviation s, so the Z-statistic becomes a t-statistic and is given by Eq. (8),

$$t = \frac{\bar{\psi} - \mu}{s - \sqrt{\nu}}, \tag{8}$$

where $\bar{\psi}$ is the mean of the residuals (shadow fading), μ is the population mean, s is the sample standard deviation, and ν is the number of degrees of freedom. If ν tends to infinity, the t-distribution becomes the normal distribution. In that way, the shadow fading term ψ from Eq. (6) is t-distributed with a PDF given as Eq. (9),

$$p(\psi) = \frac{\Gamma(\nu+1)}{\sqrt{\nu\pi}\cdot\Gamma(\nu/2)} \cdot \left(1 + \frac{\psi^2}{\nu}\right)^{-\frac{\nu+1}{2}}, \tag{9}$$

with Γ the gamma function. In the case of our experimental setup, we determined the value of $\nu = 11.43$ using the maximum likelihood criterion, getting an $R^2 = 0.996$ in the corresponding QQ plot, as shown in Figure 15.

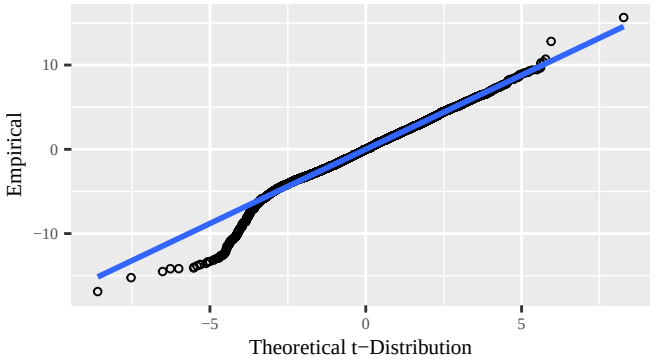


Fig. 15. Quantile-Quantile plot for residuals versus t-distribution for the MLR model. The fat tails are softened compared to the exhibited in Figure 14.

Comparing the QQ plots from Figure 14 and Figure 15, it can be noticed that *i)* the right tail was softened, and *ii)* the left tail decreased from $10\,\mathrm{dB}$ to $4\,\mathrm{dB}$. It means that the t-distribution characterizes better the shadow fading in the channel since it finds a balance between the energy consumption and the PDR, helping choose an adequate LM.