IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 24 June 2022, accepted 20 July 2022, date of publication 9 March 2023, date of current version 15 March 2023.

*Digital Object Identifier 10.1109/ACCESS.2023.3245361*

RESEARCH ARTICLE

# Topology-Based Machine Learning: Predicting Power Line Communication Quality in Smart Grids

**FRANCESCO MARCUZZI AND ANDREA M. TONELLO, (Senior Member, IEEE)**
Institute of Networked and Embedded Systems, Alpen-Adria-Universität Klagenfurt, 9020 Klagenfurt, Austria
Corresponding author: Francesco Marcuzzi (fmarcuzzi89@gmail.com)

**ABSTRACT** Smart Grids (SG) envision the exchange of both power and data, enabling system and customers to generate and transfer energy in a more efficient and balanced way. Among the relevant communication technologies, we find Power-Line Communications (PLC), which allow for data transmission on the electrical cables used for power delivery. Despite the hostile medium, PLC offer reliability and data rates to support exchange of control traffic, smart metering and sensor network applications. The distribution portion of the power delivery network, which we focus on in this work, is topologically complex, which makes channel prediction complicated. We show how random realistic topologies can be generated and then used to train a Machine Learning (ML) algorithm to infer PLC link quality (based on channel response) exploiting solely topology descriptors. We eventually show how precisely the communication quality can be inferred from the SG topology through ML. In doing so, we also discuss how the ML approach offers the common ground between top-down and bottom-up approaches for network characterization and how it enables smart decision making in the SG.

**INDEX TERMS** Bottom-up channel modeling, machine learning, network performance, power line communication, smart grid, theoretical evaluation, top-down channel modeling.

## I. INTRODUCTION

The Smart Grid (SG) concept envisions bidirectional communications on the power delivery infrastructure, enabling power consumers to provide power back to the grid and communicate with it, creating smarter, cheaper, greener energy generation and delivery. The SG was designed as a centralized system, although nowadays the paradigm is shifting towards a distributed approach for energy generation, computation capabilities and system control [1], [2], [3], [4]. The inclusion of Distributed Energy Resourcers (DER) complicates the management of the grid greatly and requires more finely engineered solutions.

The emerging paradigm is called Advanced Metering Infrastructure (AMI), which is based on small building elements called microgrids where information is processed locally before being shared with the rest of the grid. Microgrids rely on the distribution portion of the power infrastructure, making it effectively a last mile application between the Distribution Network Operator (DNO) and the final user. In fact, SGs can be seen as a modernization of the legacy electric distribution networks.

The power line medium can be exploited for communications: Power Line Communications (PLC) offer a cheap and effective solution for simultaneous transmission of data and power. The medium is suited for low-speed applications because of noisiness and harsh channel attenuation (especially at high frequencies). In fact, the microgrids/distribution sections of the power grid, encompass relatively small spaces while presenting complex topologies [5]. Nevertheless, many improvements are constantly emerging and PLC can be relied on for control traffic exchange and meter polling applications.

In preparation to this work, a dataset of real PLC access network deployments (implementing the G3-PLC protocol [6], [7]) was analyzed in detail to extract patterns and important characteristics of the infrastructure. All DNOs have

The associate editor coordinating the review of this manuscript and approving it for publication was Filbert Juwono.

extensive information on their infrastructure topology, thus we can define all fundamentals elements of a network's topology and ask the question "by knowing the topological identity of the network, how precisely can we infer PLC performance?".

The identified important parameters were used to build a bottom-up random network simulator, to augment synthetically and in a statistically representative way the dataset and eventually train a learning machine to understand the network's performance characteristics based solely on topological identifiers/features. The random network generator in itself is also a tool to evaluate characteristics of real networks.

PLC are fundamental for SG, and there have been already considerations regarding the dependency between topology and network performance [8], [9], [10], [11], [12], [13], which we started from to develop this work.

Network performance is very strongly related to the channel quality. Channel models are mainly categorized into two families: (i) bottom-up approaches, which use physically realistic models to understand what the behavior of the channel is like [14] and [15]; (ii) top-down approaches look at real measurements to optimize generation of physical channel implementations [16], [17], [18]. Machine Learning (ML) approaches to top-down models can be found in [19] and [20]. A complete overview on ML approaches for PLC can be found in [21].

Top-down models are very fast in computing link descriptors, but quite approximate. On the other hand, bottom-up approaches offer rational and realistic results, based on physical observable phenomena and relations, at the expense of a very heavy computational cost. Furthermore, the deterministic model supposes an a priori knowledge of the network's topology, which is not trivial (some methods are mentioned in Section II-B). In our case, we combine these two approaches to create a hybrid bottom-up/top-down approach to channel generation, to obtain fast, precise results.

Initially, the aim of this work was to use the datasets of real PLC access network deployment provided by an utility to further the knowledge about PLC networks and to analyze the relation between topology and network performance. The non-trivial task of analyzing this dependency brought us to use the common characteristics of different networks to define a unified topology model (main characteristics are introduced in [22]). This general model encompasses all the observed real deployments and allows for both physical and higher layer modeling.

Challenges for the SG (to name a few) include increased needs for scalability, reliability, high-level visualization tools [23], [24] and reduced control traffic [25]. With our work we look closely at the PLC technology used as infrastructure for the SG, improving on the aforementioned challenges. Namely, we: (i) create a random SG infrastructure generator based on real data for effective statistical analysis; (ii) implement a topology visualizer to enable at-a-glance infrastructure analysis; (iii) define physical-level PLC performance parameters based on Transmission Line theory, which shows good performance in modeling realistic PLC channels [26]; (iv) analyze performance in relation to topology, both with big data analysis and Machine Learning approaches. Additionally, (v) we discuss on how the ML approach can be used as a hybrid bottom-up/top-down tool for PLC network characterization.

This paper is structured as follows: Section II presents the G3-PLC field data, its observation and consequent implementation of random network generation, with specific solutions to reduce variability in the output networks; Section III explains the main bottom-up features used for the deterministic implementation of physical properties of the network; additionally, noise generation and the important parameter of node density are discussed. A few basics of Artificial Neural Networks are introduced in Section IV. Section V defines performance metrics and discusses results. Conclusions are drawn and summarized in Section VI.

## II. RANDOM POWER LINE NETWORK GENERATOR

Despite the heterogeneity of SG eletrical topologies where PLC are deployed, the accurate observation of them reveals that common features exist. This solicits the idea of developing a statistically representative grid topology generator. Such generator shall be flexible and simple: the user can specify topology, power supply modes, frequency bands, physical properties of the medium. This can be used to quickly analyze how topology changes influence the network's behavior.

To develop a random network generator, we firstly analyzed real deployments of PLC access networks and their structure. In particular, a data set of distribution networks in France was used. Based on this, we identified three types of logical elements that can be used to create a network:

1) **Nodes**: Each end node represents a final user connected to the power line infrastructure for power delivery and to a PLC modem which implements networking capabilities. In a SG interpretation of the current application, the end nodes are meters used by the local power provider to keep track of energy consumption. An additional node is the Central Concentrator (CCo) where the MV to LV transformation happens. Normally, no more than 10 different LV backbones depart from the CCo: we hence define the network as divided in sectors.
2) **Linkage Points (LP)**: connection points between the end nodes and the LV lines. Each LP connects up to 35 nodes.
3) **LV lines**: The LV lines comprise the main backbone of the infrastructure. The topology envisions the network in sectors, each one containing a tree-topology of power lines. Communication between nodes of different sectors is possible, although we assume that the CCo introduces a certain amount of attenuation.

All these elements are visible in Figure 1, where blue thick lines identify the LV lines, the downward purple triangles

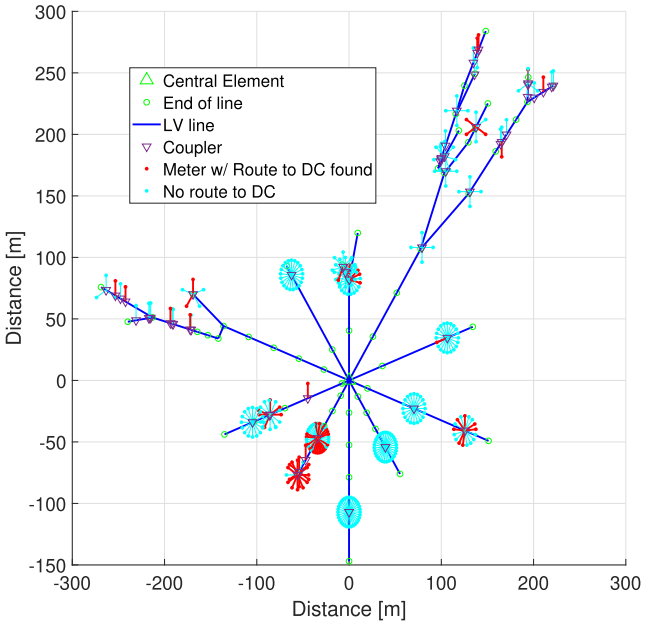



**FIGURE 1. Example of electric topology: representation of the network including logical connections between all elements, while also proportionally depicting cable lengths.**

show the LPs, the cyan and red dots the nodes and the central green triangle the CCo.

The structure of the network is thus a cluster of trees, with the CCo as root and the nodes as leaves. Since our objective lies in finding relations between the topology of the network and the quality of the links between pairs of nodes, we devise a visualization tool that allows us to look at what we refer to as the electrical topology of the network. This shows both logical connections between elements and lengths of the cables. No geographical information is retained in this representation, i.e. the real respective positions of nodes and lines in the territory; we consider only the properties of the power line infrastructure, which can be interpreted as a graph. Other types of topology, further considerations on the network and how to represent it were presented in [22].

The topology here shown is in accordance to description of residential LV PLC access networks given in [5, Ch. 2.3] and [12]. Reference [29] also briefly analyses the topology of LV access networks, describing them as mostly radial; the authors also mention difficulty in retrieving data for this part of the network, which strengthens the idea of developing a statistically representative network topology generator.

Our realistic random-network simulator is based on the topological properties of real networks, with the aim of creating a large number of representative networks. In order to do this, we fit stochastic models to recreate some observable phenomena in the real networks. The used distributions are discussed in the following and summarized in Table 1:

1) **Number of nodes per LP**: most of the LPs ($\sim$ 53%) were connected to only one node, with the rest of the distribution concentrating on low values (2 to 7) and having few LPs with a number of nodes in the low 20s to low 30s. The distribution resembled a long tail distribution, having many of the occurrences far from the mean value. A generalized Pareto distribution with rounding was used. Shape, scale and location ($k$, $\sigma$ and $\theta$) parameters can be found in Table 1. A comparison between the final empirical number of nodes per LP and the theoretical one can be seen in Figure 2.
2) **Number of LPs per LV Line**: when considering the number of LPs per single LV line, the most recurring value turned out to be 0 (55%), with decreasing empirical probability for the cases of 1 to 5 LPs, and null above. A Poisson distribution was used for the number of LPs per LV line, whose $\lambda_{LP}$ value is shown in Table 1.
   All the LV lines directly connected to the CCo share two properties: (i) they contain no LPs and (ii) they are followed by a single LV line. The final emerging distribution can be observed in Figure 3. The empirical result is different from the Poisson distribution because we modified the creation process in order to always have at least one LP in each sector.
3) **Length of LV Line**: each newly generated LV line is assigned also a certain length. In order to approximate the real cable lengths we used a Rayleigh distribution. This can be seen as a chi distribution with two degrees of freedom, or, in other words, as the length of a normally distributed two-dimensional vector. Let's consider a two-dimensional vector $D = \{X, Y\}$, with $X, Y \sim \mathcal{N}(0, \sigma_{xy})$. Then, the vector's length $L = \sqrt{X^2 + Y^2}$ follows a Rayleigh distribution, which is characterized by the parameter $\sigma_{xy}$ and shows a mean of $\mu_L = \sigma_{xy}\sqrt{\frac{\pi}{2}}$.
   This can be interpreted as $X$ and $Y$ as being geographical distances in a Cartesian 2D-space where the network is deployed. We imagine the z-coordinate variations to be negligible, i.e. for the cables all to exist at the same altitude. The process to create the LV lines length was also constrained in creating values greater than a minimum value, which was set at 6 meters.
4) **New LV line creation**: in order to create the tree topology in each sector, we devise the following approach. Each time a new LV line is created, a process decides how many new LV lines will be found downstream of it. By observing the real dataset, this value can be either 0, 1 or 2, with respective probabilities shown in Table 1.

By employing these few stochastic distributions, each sector is generated according to a process shown in Algorithm 1. This leads to the creation of realistic networks, albeit very diverse ones.

Some emerging properties of the generated networks can be found in the number of nodes per sector and per network. Almost 10% of all sectors contain a single node, while the average sector contains 27 nodes. Overall, 1% of all sectors contain more than 170 nodes with some getting up

**TABLE 1. Values for the processes used to create a random LV power line network.**

| Process to describe | Distribution | Attributes |
|---|---|---|
| Number of LPs per LV line | Poisson | $\lambda_{LP} = \frac{10}{11}$ |
| Number of nodes per LP | Rounded Generalized Pareto | $k = 1.2084$, $\sigma_{xy} = 1.5009$ and $\theta = 1$ |
| Length of an LV line | Rayleigh with minimum value | $\sigma = 29.6864$ and $min = 6$ |
| Probability of ending, continuing or branching LV line | Custom | $p = \{\frac{3}{10}, \frac{17}{30}, \frac{4}{30}\}$ |

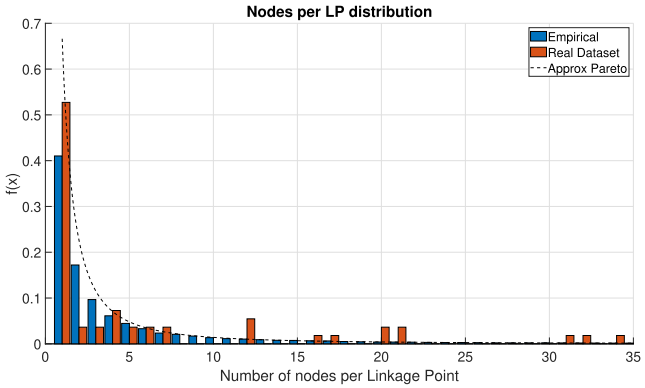



**FIGURE 2. The black dashed line presents the Pareto distribution used to approximate the real dataset (in red). The blue bars show instead the final distribution of nodes per Linkage Point obtained by capping the theoretical Pareto at 35 and rounding it to integer values. This was obtained by generating a significant number of networks. The parameters used to generate the purely theoretical distribution are listed in Table 1.**

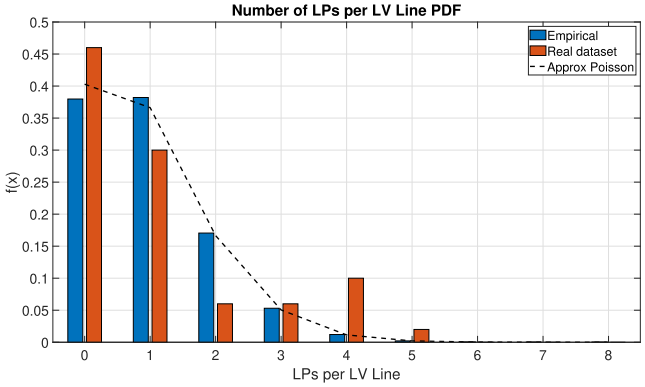



**FIGURE 3. This image presents three statistics: the theoretical one (black dashed) shows the Poisson distribution implemented on the discrete domain, i.e. the distribution created to approximate the random number of LPs connected to a generic LV line. The red bars show the data extracted from the real dataset, the number of LPs on a generic LV line. Finally, the blue bars show the empirical statistic measured across many randomly generated networks. The parameter used for the theoretical Poisson can be found in Table 1.**

to 400 nodes, although more than 90% of sectors contain less than 70 nodes. The distribution of the number of nodes per sector still resembles a Pareto distribution. On the other hand, the number of nodes per network tends to follow a normal distribution with mean of $\mu_{NPN} \simeq 270$ and a standard

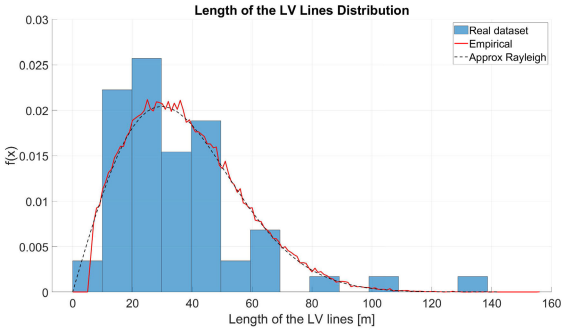



**FIGURE 4. This image shows distributions related to the length of the created LV lines. The bars show the distribution from the real dataset, the black dashed line the approximation with a pure Rayleigh distribution and the red solid line is the distribution emerging from randomly generated networks. The generator forces LV lines to be at least 6 m long. The Rayleigh distribution parameter is reported in Table 1.**

**Algorithm 1** Algorithm Describing the Creation of a Sector in the Random Network Generation Process

```
Sector Generation Start
Control Variable ← 0
while Control Variable == 0 do
    Pop Vector initialized as empty array
    Initialize 1st LV Line, no LPs, only one descendant
    Add 2nd LV line to Pop Vector
    while Pop Vector not Empty do
        Pop last element of Pop Vector
        Generate random LV Line parameters
        Add randomly 0, 1 or 2 elements to Pop Vector
    end while
    if Is there at least 1 LP in the sector then
        Control Variable ← 1
    end if
end while
```

deviation of $\sigma_{NPN} \simeq 110$ - this results in 95% of all networks containing between roughly 50 and 500 nodes. This range is quite large, thus we try to devise a new approach to better control the number of nodes in our generated network for easy load characterization.

### A. MODIFIED DISTRIBUTION APPROACH

In order to control better the generated output, we heuristically modify the distributions shown in Table 1 to intuitively tune the dimension of the network (Small, Medium, Big) and the density of nodes (Sparse, Medium, Dense). This is done by modifying the values of the original distributions and using some hard limits to have stricter ranges for the generated values of LV lines, Linkage Points and branches. These values are shown in Table 2 and Table 3. More specifically, the approach to change density involved generating more ("Dense" case) or fewer ("Sparse" case) LPs per LV line and nodes per LP. Some additional hard limits on the number of generated elements were also added. On the other hand, for the size case we introduced a minimum and

**TABLE 2. Values for the processes used to create a random LV power line network in the modified distributions approach: density.**

| Network Density | Sparse | Medium | Dense |
|---|---|---|---|
| Number of LPs per LV Line, Poisson | | | |
| $\hat{\lambda_{LP}}$ | $\frac{3}{4}\lambda_{LP}$ | $\lambda_{LP}$ | $\frac{5}{4}\lambda_{LP}$ |
| Min Number | 0 | 0 | 0 |
| Max Number | 3 | 3 | NP |
| Number of nodes per LP, Rounded Generalized Pareto | | | |
| $\hat{k}$ | $\frac{k}{2}$ | $k$ | $2k$ |
| $\hat{\sigma_{xy}}$ | $\frac{\sigma_{xy}}{2}$ | $\sigma_{xy}$ | $2\sigma_{xy}$ |
| $\hat{\theta}$ | 1 | 1 | 1 |
| Min Number | 1 | 1 | 1 |
| Max Number | 25 | 30 | 35 |

**TABLE 3. Values for the processes used to create a random LV power line network in the modified distributions approach: size.**

| Network Size | Small | Medium | Big |
|---|---|---|---|
| Probability of 0,1 or 2 offspring LV lines, Custom | | | |
| Custom PDF | $\frac{1}{60}\{22, 32, 6\}$ | $\frac{1}{60}\{21, 32, 7\}$ | $\frac{1}{60}\{20, 32, 6\}$ |
| Min Branches per Sector | 0 | 0 | 1 |
| Max Branches per Sector | 3 | 5 | NP |
| Range of distances | | | |
| Minimum distance (at least one node) | 0 | 150 | 250 |
| Maximum distance (all LV Lines) | 300 | 400 | 600 |

maximum number of branches per Sector, effectively limiting the expected number of LV lines in the network. Also, soft limits on distances were introduced: a minimum distance $d_{min}$ was used to ensure that at least one node in each sector would have $d_{from-CCo} > d_{min}$; a maximum distance $d_{max}$ was used to make sure that once an LV line ended with $d_{end-LV-line} > d_{max}$ would not generate any new offspring. The results of using this kind of generative approach can be seen in Figure 5 and 6.

### B. DENSITY OF LOADS

The load density has been shown to strongly influence the signal attenuation due to high interference because of presence of reflecting elements [9]. A fundamental requirement in designing a measure of load density is that it needs to represent effectively networks of different size. Even though it would be possible to specify density as a number of nodes per unit area, the area of a network is not easily definable, as the represented topology takes into consideration cable lengths, but not real, geographical node location. The solution here presented is to use together the number of nodes in a network and, for every node, a set of $P$ quantiles extracted from the CDF of the distances between said node and all the others in the network, with which it may communicate. For every node $i$, the CDF of distances $F_D(d)$ is implemented on the set:

$$D_i = \{d_{i,1}, d_{i,2} \dots d_{i,j} \dots d_{i,N}\} \tag{1}$$

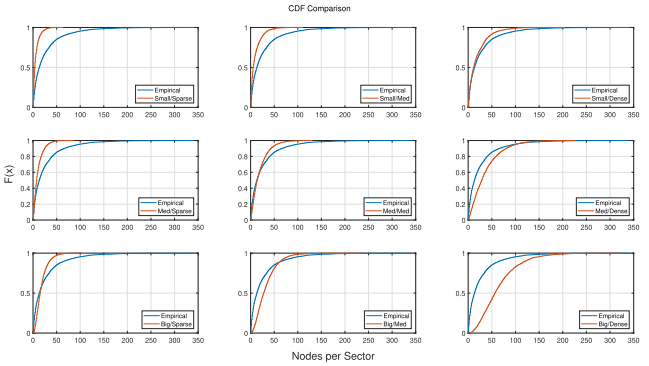



**FIGURE 5. This collection of graphs show the influence of the Modified Distribution Approach on the distribution of the number of nodes per sector. This intuitive and heuristic approach to the generation of a network allows to reduce considerably the variance of the final result. The "empirical" label on the legend of each graph identifies the curve of the original distribution where network generation is carried out with the parameters shown in Table 1.**

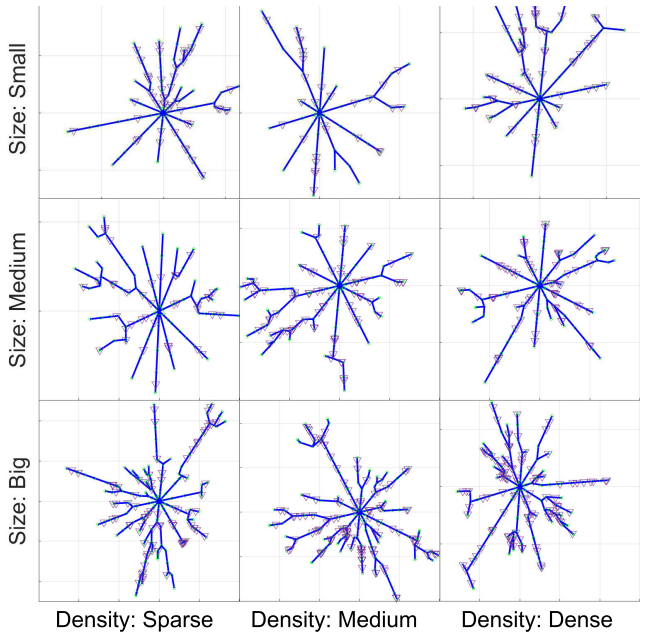



**FIGURE 6. This collection of graphs show the influence of the Modified Distribution Approach on the generated networks. The underlying grid in each graph shows 200 meter divides. In this representation of the network, the nodes were removed for simplicity of visualization, although the LPs are still present to account for varying density.**

with $N$ number of nodes in the network, $d_{i,j}$ electric cable distance between nodes $i$ and $j$ and with $j \neq i$; $D_i$ will thus contain $N-1$ elements. In order to extract $P$ quantiles from $F_D(d)$ we need to identify $P$ points $\hat{d_k}$ such that:

$$F_D(\hat{d_k}) = \frac{k}{P}, \quad \forall k \in \{1, 2 \dots P\} \tag{2}$$

Naturally, for $k = P$ we will find that $F_D(\hat{d_k}) = 1$, thus the distance:

$$\hat{d_P} = \min_d \{F_D(d) = 1\} \tag{3}$$

If $N-1$ is a multiple of $P$, quantile extraction is trivial. If not, the generic quantile $\hat{d_k}$ is extracted as a linear interpolation of

the closest values present in $D_i$:

$$d_l = \max_{d \in D_i} \{F_D(d) \leq F_D(\hat{d}_k)\}$$
$$d_g = \min_{d \in D_i} \{F_D(d) \geq F_D(\hat{d}_k)\}$$
$$\text{s.t. } d_l < \hat{d}_k < d_g$$
$$m_k = \frac{F_D(d_g) - F_D(d_l)}{d_g - d_l}$$
$$q_k = F_D(d_g) - m_k d_g$$
$$\hat{d}_k = \frac{F_D(\hat{d}_k) - q_k}{m_k}$$

where $F_D(x_k)$ behaves as specified in Equation (2). Eventually density of loads will be described for each node by a set of $P$ quantiles:

$$\hat{D}^{(P)} = \{\hat{d}_1, \hat{d}_2 \ldots \hat{d}_P\}$$

Topology can be inferred by a node by exploiting measurements of load admittance [33], since the relation between distance and measured load admittance in another point of the network can be expressed in closed form. The performance of this approach depends on the measurement technique, the frequency range of the measurement, and the presence of noise. A different approach to track the distance between loads, could be to employ a topology discovery algorithm (such as [34]) and use the packets' timestamps to evaluate it.

## III. FROM TOPOLOGY TO PERFORMANCE

To investigate PLC network performance, we need a metric that directly relates to topology and significantly expresses how well communication can be established and carried out. Signal-to-Noise-Ratio (SNR) is such a metric, as it directly relates to error rates and link capacities. Assuming a Gaussian noise distribution, the SNR allows to compute the link quality indicators and decide the modulation to be used as it is done by known standards [6], [7].

For this task, we consider the narrowband (NB) spectrum between 3 and 500 kHz, which comprises all existing NB bands as reported in Table 4. The main challenges in devising a good PLC network simulator regard (i) finding a good network model that allows realistic considerations (which is thoroughly discussed in Section II) and (ii) an efficient and precise method of calculating its performance. The defined topology will be fundamental in this process, as it is the base that allows us to carry out the latter task with realistic inputs.

What we do, is basically picking random nodes that to be strong or weak noise sources, assign them values of noise Power Spectral Densities (PSDs) and compute the effect they have on the other nodes of the network. This is done by use of the channel frequency response, which is computed through the method presented in [14] and cables' physical properties as in [27]. Transmission Line Theory is used to implement the voltage ratio approach, which consists in computing the transfer function in the frequency domain of a generic link. Specifically, this is done by:

**TABLE 4. NB PLC Spectra standards.**

| Location | Standard | Band [kHz] |
|---|---|---|
| Europe | EN 50065-1 CENELEC | 3-148.5 |
| Utilities | CENELEC A | 3-95 |
| Consumer | CENELEC B | 95-125 |
| | CENELEC C | 125-140 |
| | CENELEC D | 140-148.5 |
| US | FCC Part 15 for Carrier Current Devices | 9-490 |
| Japan | ARIB STD-T84 | 10-450 |
| Canada | ICES-006 | 0-535 |

1) identifying the backbone that connects the two end nodes;
2) simplifying the surrounding network topology into equivalent impedances;
3) splitting the backbone into transmission line portions;
4) computing each single portion's individual transfer function $H_b(f)$ as the ratio of the input and output voltage;
5) computing the overall transfer function as the scalar product of the individual $H_b(f)$'s.

In the following, we present more in detail how the noise generation works for our endeavor, and how the methods provides realistic values.

### A. NOISE AND ITS TOPOLOGY DEPENDENCE

To evaluate the real quality of a link in PLC network, we need to know the SNR between any pair of transmitters and receivers in the network. The aforementioned ratio is defined as:

$$SNR(f_k) = \frac{P_{T,i}(f_k) \left|H_{i,j}(f_k)\right|^2}{P_{N,j}(f_k)} \tag{4}$$

where $P_{T,i}(f_k)$ is the power of the signal emanated by transmitter $i$, $P_{N,j}(f_k)$ is the power of the noise perceived at the receiver port of index $j$ and $\left|H_{i,j}(f_k)\right|^2$ is the squared magnitude of the channel frequency response between the nodes $i$ and $j$ in the link. $f_k$ is the frequency at which the equation is considered.

From the noise generation standpoint, we treat our PLC network as an instantaneous snapshot of the whole system in absence of noise bursts. We assume that the most prominent noise sources are located at the terminal nodes that are also used for communication. All nodes are considered noise sources, although most of them are considered to be mild generators, such as private residences. Few nodes will act as intense noise generators, which model the presence of inverters, transformers and the such. The type of noise source determines the base noise level of a source: this is the flat white PSD noise profile that is generated and sensed by the generic node in case all the others are turned off. To determine the real noise level of a node network, the topology is used to compute the equivalent noise profile heard at a node from

other sources:

$$P_{N,i}(f_k) = P_{N,i}^{base}(f_k) + \sum_{j \neq i}^{Nodes} \left|H_{j,i}(f_k)\right|^2 P_{N,j}^{base}(f_k) \qquad (5)$$

In Equation (5), $P_{N,i}(f_k)$ represents the real level of noise at the specified frequency, while any $P_{N,i}^{base}(f_k)$ is the base level of noise as defined above, with $i \in Nodes$.

Our simple model proves to be validated by real measurements. A significant number of simulations were run on different networks (different profiles were considered based on the approaches described in Section II-A), with different configurations on the number of nodes acting as strong or mild noise sources, with different levels of base noise levels. Specifically, the mild noise sources were always generating a base noise flat PSD at $P_N(f) = -140 dBmW/Hz$, while the generic strong source had its level set randomly at a flat level extracted by a Gaussian process, with $P_N(f) \sim \mathcal{N}\left(\mu_{P_N}, \sigma_{P_N}\right)$, with $\mu_{P_N} = -65 dBmW/Hz$ and $\sigma_{P_N} = 10\, dBmW/Hz$.

By averaging the behavior of different networks and configurations, we found that our profiles agree with the ones presented in [28], which shows measurements of different networks from France about the perceived noise both at the end-node and at the transformer for the same spectrum we address here. The model we employ here, allows us to find the profiles presented in Figure 7 and 8 for different network configurations - these curves are obtained by averaging the channel responses in the logarithmic scale, as doing it in the linear one would highlight the few peak-values that might appear at any frequency and thus dominate the distribution. In the figures, opposite trends can be noticed: smaller networks produce higher noise at node ports and lower noise at the CCo. The high noise at the node ports is explained by the fact that smaller networks, by definition, implement shorter distances between the nodes, so when one is a noise source, it will have a stronger effect on its neighbors. At the same time, small size networks allow for fewer nodes to be deployed, thus the CCo, which is always separated by some distance from the nodes since the first LV line out of it is consistently empty, will hear cumulatively less noise. When compared with the data presented in [28], we can see that our values fall inside the same range. In [28], the authors measure the noise level in $dB\mu V/Hz$ by using an instrument with input impedance $Z = 50\Omega$, while our measurements are in $dBmW/Hz$. Considering that $P = V^2/R$, we can write that:

$$\begin{cases} M_{dB\mu V} = 20 \cdot \log_{10}\left(V \cdot 10^6\right) \\ M_{dBmW} = 10 \cdot \log_{10}\left(P \cdot 10^3\right) = 10 \cdot \log_{10}\left(\frac{V^2}{R} \cdot 10^3\right) \end{cases}$$

where $M_{dB\mu V}$ and $M_{dBmW}$ are respectively the measurements in the subscript-specified unit. This system can be used to find the equivalence between the two measurement systems as:

$$M_{dBmW} \simeq M_{dB\mu V} - 107\ dB$$

which shows that our simulated values fall inside the same range ($P_N \sim\in [-65, -40]\, dBmW/Hz$ for node ports, $P_N \sim\in [-72, -45]\, dBmW/Hz$ for the CCo port) as the ones measured on real deployments ($P_N \sim\in [20, 70]\, dB\mu V/Hz$ for node ports, $P_N \sim\in [0, 80]\, dB\mu V/Hz$ for the CCo port).

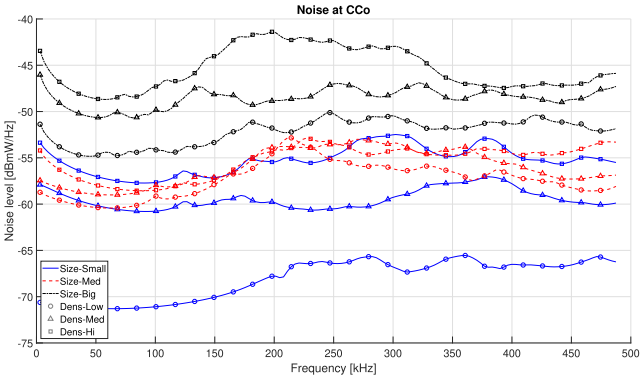



**FIGURE 7.** This graph presents the average level of noise for different types of networks measured at the CCo ports across the NB-PLC spectrum.

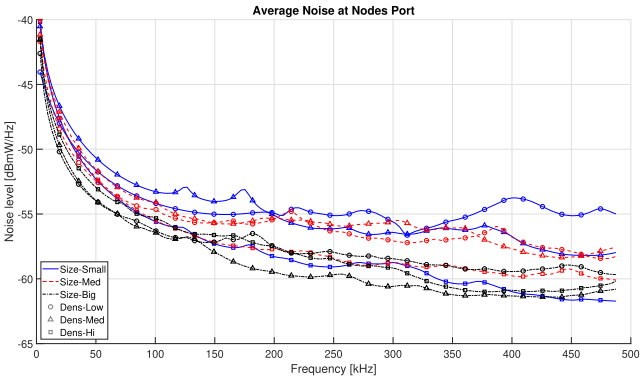



**FIGURE 8.** This graph presents the average level of noise for different types of networks measured at the generic node port across the NB-PLC spectrum.

## IV. PERFORMANCE FROM TOPOLOGY: ML APPROACH

Together, the random realistic-network topology generator Section II and the topology-based performance evaluation Section III, can be used to analyze metrics based on the appearance and structure of the network. In order to identify significant patterns and relations, a significantly large amount of data needs to be systematically processed and analyzed. For this purpose, we approach the problem from the applicative Machine Learning standpoint: ideally, we want a Deep Feedforward Network (DFN) to infer the magnitude of the channel's response and the SNR based solely on the observation of topological descriptors. This is done to answer the question "how precisely can we infer performance from topology?".

The approach here used follows the supervised learning paradigm [32]. This basically means that the learning machine is taught to solve a specific problem by being shown both inputs (topological descriptors) and outputs (performance metrics). The outputs - also referred to as labels - are deterministically computed through the method described in Section III. Specifically, we use DFNs in our approach.

The topological descriptors used here are many. Some of them are trivial while others proved to be of greater impact on the performance of the learning machine. In the following section we give details regarding the architecture of the used DFNs before moving onto performance inference evaluation.

### A. CHANNEL RESPONSE REGRESSION

The DFN paradigm described above can be trained to compute a desired value based on some input parameters. Our objective is to use only topological descriptors to compute (i) the magnitude of the channel response between two nodes of the network and (ii) the SNR of a link based on indicators of noise source distribution, with the model described in Section III-A (described in Section IV-B). In order to do this, we collect a large database of synthetically generated channel responses and connect each value to a set of topological values, as described in the following:

- The only non topological parameter used is the frequency at which to infer the considered metric.;
- The number of nodes in the network;
- Regarding the two nodes involved in any considered link: (i) electrical cable distance, (ii) Boolean values on whether they pertain to the same sector and LP, (iii) deciles of load density for both of them (method to compute this is described in Section II-B), (iv) distance of each node from its respective LP, (v) number of LV lines, LPs, nodes and branching points in each node's respective sector;
- Regarding the path between the two nodes: (i) number of LV line segments, (ii) electrical cable length of the first and last LV line segments of the path, (iii) number of LPs, branching points, empty LV line segments along the path.

Two of the above elements showed a strong impact on the final performance: (i) use of the density descriptor strongly raised correlation between real and inferred performance; (ii) use of information regarding the path between the nodes in the link reduced drastically (10x) the amount of data needed in training to achieve the best observed precision. Eventually, the amount of data points needed to train our architecture to infer performance was around 50 to 100 thousand elements.

We trained different DFNs to experiment which configuration works best with the problem at hand, all of them using MATLAB's standard Levenberg-Marquardt backpropagation, which uses Mean Square Error as a loss metric and by default stops training after a maximum of 6 validation failures or 1000 epochs. All of our DFNs' trainings stopped because too many successive validation failures. Additionally, data division was done automatically and randomly by the training tool and backpropagation computations were done in MEX (which is an alternative to MATLAB with increased memory efficiency). The training of the neural networks was done on a computer with an Intel(R) Core(TM) i7-7500U CPU @ 2.70 GHz, 2904 MHz, 2 Cores and 4 Logical Processors; the fastest trainings with these settings were the ones of the 2-layer DFNs, which took around 2 hours to complete, while 5-layer DFNs would complete in roughly 30 hours. We tried different combinations of depths (2 to 5 layers) and activation functions (mostly combinations of ReLUs and sigmoids), but came to find that any DFN with more than 2 layers would lose performance due to overfitting and loss of generality. As a rule of thumb, any experimented DFN would have a smaller number of neurons in each subsequent layer.

Eventually, out of all the experimented features, we find that with an input layer with about 50 neurons, the best performance is reached with a 2-layer-DFN with the following characteristics: 1st layer, 100 neurons and ReLU activation function; 2nd layer, 50 neurons and sigmoid activation function.

The design of hidden units research area is very alive, with active discussions regarding definitive indications and guidelines for design. Because of this, our work also contributes in discussing a chosen taxonomy of DFNs and their performance on the problem at hand [32]. From an empirical point of view, deeper networks seem to result in better generalization for a wider variety of tasks. Being our task quite specific, we find that additional layers do not increase the performance of the estimator.

### B. SNR REGRESSION

With the following experiment, we want to push the boundary of the DFN capability of abstracting. The idea is to use topological information available exclusively at the transmitter side infer the SNR between the transmitter and a generic receiver in the network. In reference to Equation (4) in Section III-A, we observe that the value of transmitted power $P_{t,i}$ is always known to node $i$. In our model, we assume that any node is transmitting at the same exact power, thus this value is constant throughout the network and is not used for the DFN training. What the DFN will eventually compute based on topology is the channel gain to noise ratio measured in decibel $CNR^{dB}_{(i,j)}(f_k)$ between a transmitter $i$ and a receiver $j$:

$$CNR^{dB}_{(i,j)}(f_k) = 10 \cdot \log_{10}\left(\frac{|H_{i,j}(f_k)|^2}{P_{N,j}(f_k)}\right) \tag{6}$$

The noise model for the inference of the SNR was slightly simplified, limiting the number of strong noise sources per network to 3. Additionally, the mild noise sources all had a flat PSD with the same level:

$$P_{N,mild}(f) \sim \mathcal{N}\left(\mu_{P_{N,m}} = -130\frac{dBmW}{Hz}, \sigma_{P_N} = 3\frac{dBmW}{Hz}\right)$$

while the strong noise sources were set at the same level with the distribution:

$$P_{N,strong}(f) \sim \mathcal{N}\left(\mu_{P_{N,s}} = -65\frac{dBmW}{Hz}, \sigma_{P_N} = 3\frac{dBmW}{Hz}\right)$$

Two different experiments were run: (i) different DFNs were trained on data extracted from many simulations of random PLC networks, with the intent of teaching them to infer a link's SNR in an unseen network with 3 randomly selected strong noise source, and (ii) a DFN was trained on a unique

PLC network changing the position of the 3 strong noise sources and the PSD level of mild and strong noise sources. The input set of DFNs was the same as per the Channel Response case, with the addition of the following for each noise source:

- Distance from the receiver node;
- Boolean if same LP as receiver node;
- Boolean if same sector as receiver node.

## V. RESULTS

### A. CHANNEL RESPONSE MAGNITUDE

The performance of the regression DFNs was evaluated by means of different metrics, presented in the following, computed by evaluating $N_C = 5000$ random samples extracted from each one of 381 randomly generated networks that were not part of the training set. We define $\mathbf{X}$ as the set of data extracted from each network to infer the channel response, $\mathbf{T}$ as the corresponding desired outputs and $\mathbf{Y}$ as the inferred data when $\mathbf{X}$ is applied as an input to our neural network described in Section IV-A.

Mean Squared Error (MSE) is a well known measure, in this case applied to a predictor, which computes the average of the square of the difference between the original distribution values and the ones inferred by the predictor (our Neural Network):

$$MSE = \frac{1}{N_C}\sum_{i=1}^{N_C}(T_i - Y_i)^2 \tag{7}$$

This metric is chosen to give a reference performance value regarding the quality of the predicted values $\mathbf{Y}$. The MSE is the second moment of the error, and as such, is strictly positive, contains information both about the bias and the variance of the error and indicates better performance when it is close to 0. In our case, we can consider the bias to be negligible (cfr. with results about the CDF error), thus the MSE estimates very well the variance of the error $E_i = |T_i - Y_i|$.

Another measure of performance of our predictor is given by the Mutual Information (MI) of the sample population $\mathbf{T}$, which is the desired output for the predictor, and the sample population $\mathbf{Y}$, which is the actual output. MI is classically intuitively described as a measure of how much the knowledge about one random variable tells us about another one, although its formal formulation is:

$$I(T;Y) = \int_t\int_y p_{T,Y}(t,y)\log\left(\frac{p_{T,Y}(t,y)}{p_T(t)p_Y(y)}\right)dtdy \tag{8}$$

where $p_T(t)$ and $p_Y(y)$ are the marginal distributions. The MI between $\mathbf{T}$ and $\mathbf{Y}$ can also be seen as the reduction in the entropy of $\mathbf{T}$ (or $\mathbf{Y}$) when knowledge of $\mathbf{Y}$ (or $\mathbf{T}$) is available - basically, the measurable reduction in uncertainty. MI is a strictly non negative value, which can be null only in the case of independent distributions.

In Figure 9, we can observe the performance of a few of our DFNs on the batch of 381 randomly generated networks, which are qualitatively similar to the ones used for training (same physical attributes for cables, same distribution for load impedance, same topological distributions…), although never seen before by the Neural Network. This graph is used to show the performance when the number of hidden layers varies. The aforementioned performance is expressed by means of the CDF of the MSE, basically showing how our predictor behaves on a representative range of different networks. From the graph, one can observe that, when the predictor has three or five hidden layers, roughly 90% of the cases present a variance of less than 9, which in turn corresponds to an average standard deviation of $\pm 3$ (or $\pm 30$ $dB$). In other words, in 9 networks out of 10, these predictors can guess the value of the squared absolute value of the channel response within 3 orders of magnitude for 67% of the cases. This is related to the fact that the distribution of the logarithm of said absolute value is roughly normal ( [29] mentions that the magnitude of the channel responses in a network is log-normally distributed). When the predictor only contains two hidden layers, then 90% of the cases present an MSE smaller than 8. In linear scale, this means that the error is around 66% of the ones in the three/five layers cases.

The MSE is also evaluated for the CENELEC frequency spectrum (3 kHz to 487.5 kHz). This result is reported in Figure 10: the MSE is clearly stronger for the higher end of the spectrum. This is due to the fact that while the DFN approach is able to pick up the dependency between attenuation and frequency, it is hardly capable to understand relations between frequency, topology and phase of a signal. This in turn creates higher errors at higher frequencies due to a harder-to-predict influence of the signal phase on the overall attenuation.

An additional relation that was explored and is worth mentioning, is that larger networks tend to accrue higher MSEs.

Figure 11 shows the CDF of $I(T;Y)$ over the whole dataset. The MI-performance roughly follows a normal distribution indicating (as expected also form the MSE-performance) that the distributions of desired and inferred data are not independent. Contrary to the MSE case, the MI seems to be independent from the number of nodes in the network. This indicates that our ML approach is scalable and generalizes physical properties well.

The value of the MI shows us that there is a correlation between the deterministic and the inferred performances of the network. When considering the distributions of $\mathbf{T}$ and $\mathbf{Y}$, it's possible to see a strong increase (about 10-15x) in the MI values. This basically shows that our DFNs are able to infer the performance characteristics of PLC networks simply from their topology. DFNs can thus be used as hybrid top-down/bottom-up network characteristics generators: they start from physical descriptors of networks and generate representative values while being quick.

### B. PREDICTION OF SNR

Initially, we herein show the results regarding the DFNs trained to infer the SNR in unseen networks. For this stage

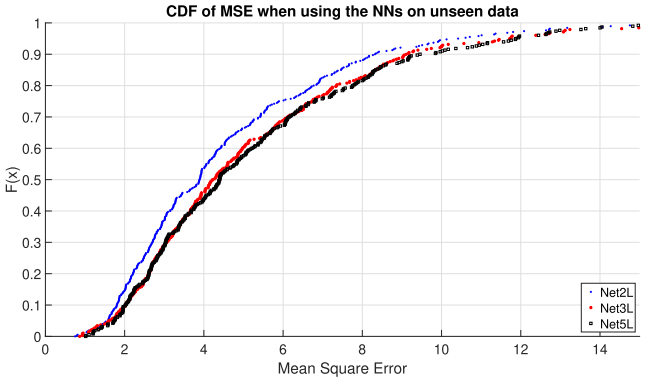



**FIGURE 9.** This graph presents the CDF of the MSE for three predictors with different numbers of hidden layers. They are tested on 381 different networks, which were never seen in training phase. The high number of newly observed networks gives us an idea of the average performance of our predictors.

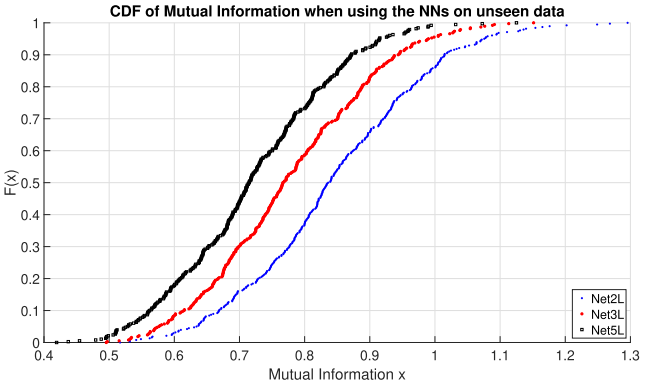



**FIGURE 11.** This graph presents the CDF of the MI for the predictors tested on 381 different networks, which were never seen in training phase. The high number of newly observed networks gives us an idea of the average performance of our predictors. The trend shows that an increase in the number of hidden layer corresponds to a loss of MI.

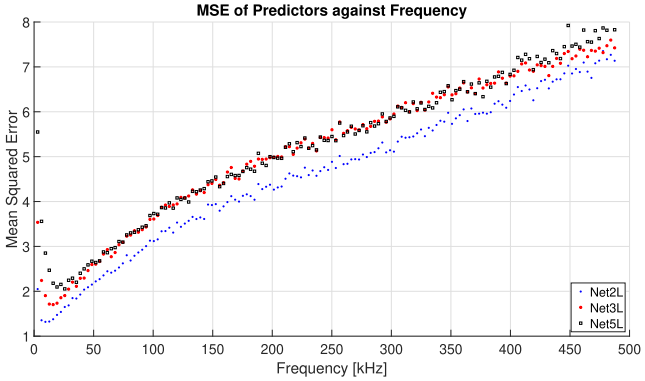



**FIGURE 10.** In this graph, the MSE is reported as a function of frequency. As mentioned above, the analyzed spectrum covers the entirety of the CENELEC bands, from 3 kHz to 487.5 kHz. There is a clear relation between the expected MSE and the frequency, as higher frequencies imply higher MSE. Considering any fixated distance, higher frequencies imply stronger attenuation on the travelling signals and also shorter wave lengths, which imply faster oscillating sinusoidal variations of the signals. This results in harder-to-predict amplitudes based on the low resolution of topological data. Also, a smaller number of hidden layers shows better performance.

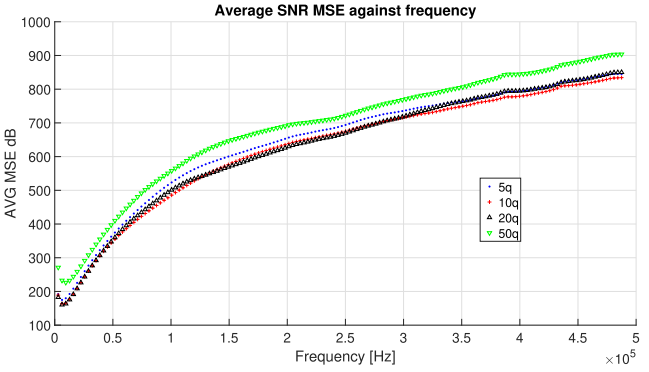



**FIGURE 12.** This graph shows the performance of our DFN trained on inferring the SNR of a link on unseen PLC networks. The different markers identify different DFNs who were trained with different number of quantiles for density description. The DFNs that used 50 quantiles shows the worst performance, followed by the one with 5 quantiles, with 20 and the one with 10 is the one presenting best performance. This one identifies the best compromise between under and over fitting.

we want to show the effect of a different number of density quantiles when inferring the SNR. Four different DFNs were used which comprised either 5, 10, 20 or 50 density quantiles (cfr. Section II-B), both for the transmitter and receiver nodes in input. As can be observed in Figure 12, when the number of quantiles is low or high (5 or 50) the DFNs tend to be less correct in the inference. This might be due to absence of enough information or overfitting. Nevertheless, a usage of 10 or 20 quantiles seems to be the local optima for this specific problem. Unseen PLC networks are nevertheless too hard to predict correctly - lower frequency show an average squared error around 200, which translates into an average uncertainty of $\pm 14$ *dB*, while the higher end of the spectrum shows up to an $\pm \sim 30$ *dB* uncertainty.

Eventually, a DFN was trained to infer link SNR between node pairs from a single network, for different unseen configurations of strong and mild noise sources. The network used in this procedure is a medium sized network containing 202 nodes. Performance was studied exclusively with a DFN using 10 quantiles for density, which presented improved performance w.r.t the unseen networks case. In fact the lower end of the spectrum shows a reduced uncertainty of $\pm 7$ *dB* on average, while the end spectrum around $\pm 16$ *dB*.

The second DFN was not only trained on a single network and tested on the same, but was also trained with a larger dataset than the first one inferring on unseen PLC networks. The former was trained on 400 thousand points of dataset, while the latter on about 150 thousand. While it is clear that larger datasets offer a more precise training, smaller ones can also be effective for an approximate inference of the network's performance, while avoiding overfitting problems.

The topology data used in this experiment is rather bare: the network description fed to the DFNs contains a few geometrical distances and node-density information. Any electrical descriptor (type of cables, load impedance, signal travelling speed…) is purposedly omitted to understand better what weight the topology has on network performance, which turns out to be quite important. The results shown above

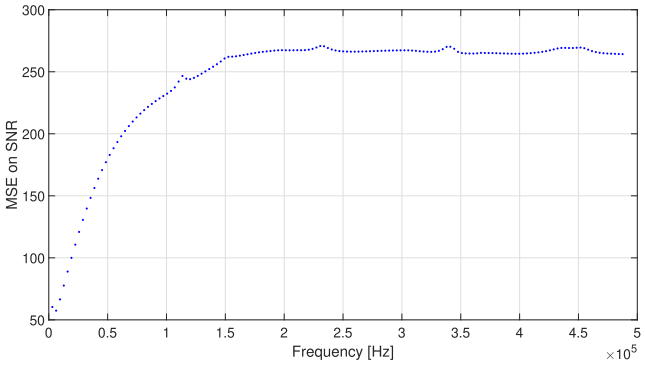



**FIGURE 13.** This graph shows the performance of our DFN trained on inferring the SNR of a link on a single, specific PLC network. Density was described with 10 quantiles. This configuration shows an improved performance w.r.t. to the unseen networks case - an unchanging topology reduces the degrees of freedom with which the topological parameters can change (e.g., number of nodes is always the same). By limiting the topology variability, we obtain more precise results.

confirm a strong correlation between the desired data and the inferred one. The possibility to quickly infer channel quality and SNR in PLC networks through ML approaches enables fast assessment of control-requirements (routing, switching, etc…) in the design phase of a network or even before a simple topology refactoring (addition or subtraction of one or more nodes, failure of a line section, etc…).

The employment of a bottom-up model to generate realistic data for ML training shows that: (i) the topology description of the PLC network is a strong element in predicting the performance in terms of link metrics and (ii) that the topology-based ML approach is perfectly capable of reproducing the correct distributions from which realistic Channel Frequency Responses can be extracted, similarly to a top-down based method. In this sense, we achieve convergence between the two approaches.

## VI. CONCLUSION

In this article, we presented a novel random electrical grid generator based on the observation of real deployments. We used this model together with a bottom-up channel generator to determine realistic channel frequency responses and noise spectral densities. Overall, the generator allows to create realistic and location dependent SNR values. Machine Learning was then used to train a neural network capable of predicting link quality (SNR) from simple grid topology parameters. Our results highlight the effectiveness of designing the NN properly to infer the target quantities (SNR and signal magnitude attenuation due to the channel), the possibility of implementing a quick hybrid bottom-up/top-down algorithm based on topology for realistic channel response and SNR generation. In turn, this allows us to show that PLC network performance strongly depends on grid topology. The design, planning and deployment of next generation power line communication networks in support of the smart grid will rely on the dependencies between topology and performance as it has been shown by the contribution of this paper.

## VII. ACKNOWLEDGMENT

The authors wish to acknowledge the cooperation with the G3-PLC Alliance in the Project "G3-PLC Networks Data Learning, in 2022," and the fruitful discussions with Cedric Lavenu about the topology of power line access networks.

## REFERENCES

[1] F. Bouhafs, M. Mackay, and M. Merabti, "Links to the future: Communication requirements and challenges in the smart grid," *IEEE Power Energy Mag.*, vol. 10, no. 1, pp. 24–32, Jan./Feb. 2012, doi: 10.1109/MPE.2011.943134.

[2] Y. Yan, Y. Qian, H. Sharif, and D. Tipper, "A survey on smart grid communication infrastructures: Motivations, requirements and challenges," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 1, pp. 5–20, 1st Quart., 2012, doi: 10.1109/SURV.2012.021312.00034.

[3] G. Habault, M. Lefrancois, F. Lemercier, N. Montavont, P. Chatzimisios, and G. Z. Papadopoulos, "Monitoring traffic optimization in a smart grid," *IEEE Trans. Ind. Informat.*, vol. 13, no. 6, pp. 3246–3255, Dec. 2017, doi: 10.1109/TII.2017.2742584.

[4] V. C. Gungor, D. Sahin, T. Kocak, S. Ergut, C. Buccella, C. Cecati, and G. P. Hancke, "A survey on smart grid potential applications and communication requirements," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 28–42, Feb. 2013, doi: 10.1109/TII.2012.2218253.

[5] L. Lampe, A. M. Tonello, and T. Swart, *Power Line Communications: Principles, Standards and Applications From Multimedia to Smart Grid*, 2nd ed. Hoboken, NJ, USA: Wiley, 2016, doi: 10.1002/9781118676684.

[6] *Narrowband Orthogonal Frequency Division Multiplexing Power Line Communication Transceivers for G3-PLC Networks*, document G9903, 2021.

[7] S. Galli and T. Lys, "Next generation narrowband (under 500 kHz) power line communications (PLC) standards," *China Commun.*, vol. 12, no. 3, pp. 1–8, Mar. 2015, doi: 10.1109/CC.2015.7084358.

[8] F. Marcuzzi and A. M. Tonello, "Artificial intelligence based routing in PLC networks," in *Proc. IEEE Int. Symp. Power Line Commun. Appl. (ISPLC)*, Manchester, Apr. 2018, pp. 1–6, doi: 10.1109/ISPLC.2018.8360241.

[9] F. Marcuzzi and A. M. Tonello, "Artificial-intelligence-based performance enhancement of the G3-PLC LOADng routing protocol for sensor networks," in *Proc. IEEE Int. Symp. Power Line Commun. Appl. (ISPLC)*, Praha, Czech Republic, Apr. 2019, pp. 1–6, doi: 10.1109/ISPLC.2019.8693390.

[10] F. Marcuzzi and A. M. Tonello, "Radio access network backhauling using power line communications," in *Broadband Communications Networks—Recent Advances and Lessons From Practice*, Sep. 2018, doi: 10.5772/intechopen.73390.

[11] O. G. Hooijen, "On the relation between network-topology and power line signal attenuation," in *Proc. IEEE Int. Symp. Power Line Commun. Appl. (ISPLC)*, Tokyo, Japan, Dec. 1998, pp. 45–56.

[12] T. Esmailian, F. Kschischang, and P. G. Gulak, "In-building power lines as high-speed communication channels: Channel characterization and a test channel ensemble," *Int. J. Commun. Syst.*, vol. 16, no. 5, pp. 381–400, 2003, doi: 10.1002/dac.596.

[13] P. Mlynek, J. Misurec, M. Koutny, and P. Silhavy, "Two-port network transfer function for power line topology modeling," *Radioengineering*, vol. 21, pp. 356–363, Apr. 2012.

[14] A. M. Tonello and F. Versolatto, "Bottom-up statistical PLC channel modeling—Part I: Random topology model and efficient transfer function computation," *IEEE Trans. Power Del.*, vol. 26, no. 2, pp. 891–898, Apr. 2011.

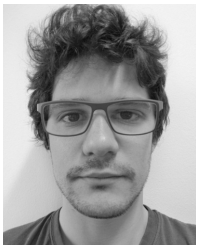
[15] J. Le, C. Wang, W. Zhou, Y.-Y. Liu, and W. Cai, "A novel PLC channel modeling method and channel characteristic analysis of a smart distribution grid," *Protection Control Modern Power Syst.*, vol. 2, no. 1, p. 14, Dec. 2017, doi: 10.1186/s41601-017-0044-2.

[16] A. M. Tonello, F. Versolatto, B. Béjar, and S. Zazo, "A fitting algorithm for random modeling the PLC channel," *IEEE Trans. Power Del.*, vol. 27, no. 3, pp. 1477–1484, Jul. 2012.

[17] M. Zimmermann and K. Dostert, "A multipath model for the powerline channel," *IEEE Trans. Commun.*, vol. 50, no. 4, pp. 553–559, Apr. 2002, doi: 10.1109/26.996069.

[18] A. Pittolo and A. M. Tonello, "A synthetic statistical MIMO PLC channel model applied to an in-home scenario," *IEEE Trans. Commun.*, vol. 65, no. 6, pp. 2543–2553, Jun. 2017, doi: 10.1109/TCOMM.2017.2677938.

[19] D. Righini, N. A. Letizia, and A. M. Tonello, "Synthetic power line communications channel generation with autoencoders and GANs," in *Proc. IEEE Int. Conf. Commun., Control, Comput. Technol. Smart Grids (SmartGridComm)*, Beijing, China, Oct. 2019, pp. 1–6, doi: 10.1109/SmartGridComm.2019.8909700.
[20] J. P. A. Leon, F. J. Rico-Novella, and L. J. De La Cruz Llopis, "Predictive traffic control and differentiation on smart grid neighborhood area networks," *IEEE Access*, vol. 8, pp. 216805–216821, 2020, doi: 10.1109/ACCESS.2020.3041690.
[21] A. M. Tonello, N. A. Letizia, D. Righini, and F. Marcuzzi, "Machine learning tips and tricks for power line communications," *IEEE Access*, vol. 7, pp. 82434–82452, 2019.
[22] F. Marcuzzi, A. M. Tonello, and N. A. Letizia, "Discovering routing anomalies in large PLC metering deployments from field data," in *Proc. IEEE Int. Symp. Power Line Commun. Appl. (ISPLC)*, Malaga, Spain, May 2020, pp. 1–6.
[23] A. Sanchez and W. Rivera, "Big data analysis and visualization for the smart grid," in *Proc. IEEE Int. Congr. Big Data (BigData Congress)*, Honolulu, HI, USA, Jun. 2017, pp. 414–418, doi: 10.1109/BigDataCongress.2017.59.
[24] D. V. Nga, O. H. See, D. N. Quang, C. Y. Xuen, and L. L. Chee, "Visualization techniques in smart grid," *Smart Grid Renew. Energy*, vol. 3, no. 3, pp. 175–185, 2012, doi: 10.4236/sgre.2012.33025.
[25] D. Hartmann, K. Wolter, and T. Krauss, "ICT resilience simulations in small confined smart distribution grids," in *Proc. Intelec 35th Int. Telecommun. Energy Conf., Smart Power Efficiency*, Hamburg, Germany, 2013, pp. 1–6.
[26] M.-Y. Zhai, "Transmission characteristics of low-voltage distribution networks in China under the smart grids environment," *IEEE Trans. Power Del.*, vol. 26, no. 1, pp. 173–180, Jan. 2011, doi: 10.1109/TPWRD.2010.2067228.
[27] A. M. Tonello and T. Zheng, "Bottom-up transfer function generator for broadband PLC statistical channel modeling," in *Proc. IEEE Int. Symp. Power Line Commun. Appl.*, Mar. 2009, pp. 7–12, doi: 10.1109/ISPLC.2009.4913395.
[28] I. Elfeki, S. Jacques, I. Aouichak, T. Doligez, Y. Raingeaud, and J.-C. Le Bunetel, "Characterization of narrowband noise and channel capacity for powerline communication in France," *Energies*, vol. 11, no. 11, p. 3022, Nov. 2018.
[29] S. Galli, A. Scaglione, and Z. Wang, "For the grid and through the grid: The role of power line communications in the smart grid," *Proc. IEEE*, vol. 99, no. 6, pp. 998–1027, Jun. 2011, doi: 10.1109/JPROC.2011.2109670.
[30] B. Tan and J. Thompson, "The parameters for powerline channel modeling," 2012, *arXiv:1203.3245*.
[31] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature*, vol. 521, pp. 436–444, May 2015, doi: 10.1038/nature14539.
[32] I. B. Y. Goodfellow and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
[33] F. Passerini and A. M. Tonello, "On the exploitation of admittance measurements for wired network topology derivation," *IEEE Trans. Instrum. Meas.*, vol. 66, no. 3, pp. 374–382, Mar. 2017, doi: 10.1109/TIM.2016.2636478.
[34] H. Li, C. Dan, B. Huaixiang, and L. Shurong, "Topology discovery algorithm based on ant colony algorithm of power line carrier sensor network," in *Proc. Int. Conf. Commun. Softw. Netw.*, Macau, 2009, pp. 102–105, doi: 10.1109/ICCSN.2009.45.



**FRANCESCO MARCUZZI** received the Laurea Magistrale degree in electronic engineering from the University of Udine, Italy, in 2016. He is currently undertaking higher education courses in the topics of software engineering and machine learning. He was a Researcher at the University of Klagenfurt. In more recent years, he was a software engineer, developing machine learning solutions for logistic applications. He published in different conferences on the topic of power line communications. His related research interests include applied machine learning and routing algorithms.

**ANDREA M. TONELLO** (Senior Member, IEEE) received the Laurea degree (summa cum laude) in electrical engineering and the Ph.D. degree in electronics and telecommunications from the University of Padova, Padua, Italy, in 1996 and 2002, respectively. From 1997 to 2002, he was at the Bell Laboratories-Lucent Technologies, Whippany, NJ, USA, first as a member of the Technical Staff. He was then promoted to Technical Manager and appointed as the Managing Director of the Bell Laboratories, Italy. In 2003, he joined the University of Udine, Udine, Italy, where he became an aggregate Professor, in 2005, and an Associate Professor, in 2014. He then moved at the University of Klagenfurt, Austria, where he is currently a Professor of embedded communication systems. He has also a post as a part-time Associate Professor with the University of Udine, since the end of 2020. He received several awards, including the Bell Laboratories Recognition of Excellence Award (1999), the Distinguished Visiting Fellowship from the Royal Academy of Engineering, U.K. (2010), the IEEE Distinguished Lecturer Award from VTS (2011–2015) and from COMSOC (2018–2019), the Italian Full Professor Habilitation (2013), the Chair of Excellence from Carlos III Universidad, Madrid (2019–2020). He was also a co-recipient of ten best paper awards. He was the general chair or the TPC co-chair of several conferences. He was the Chair of the IEEE COMSOC Technical Committee (TC) on power line communications (2014–2018). He is currently the Chair of the TC on smart grid communications. He also served as the Director of Industry outreach in the board of governors of IEEE COMSOC (2019–2020). He serves/ed as an Associate Editor of several journals, including IEEE Transactions on Vehicular Technology, IEEE Transactions on Communications, IEEE Access, and *IET Smart Grid*.