

Received May 22, 2019, accepted June 5, 2019, date of publication June 17, 2019, date of current version July 9, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2923321

Machine Learning Tips and Tricks for Power Line Communications

ANDREA M. TONELLO¹, (Senior Member, IEEE), NUNZIO A. LETIZIA¹,
DAVIDE RIGHINI, (Student Member, IEEE), AND FRANCESCO MARCUZZI

Chair of Embedded Communication Systems, University of Klagenfurt, 9020 Klagenfurt, Austria

Corresponding author: Andrea M. Tonello (andrea.tonello@aau.at)

ABSTRACT A great deal of attention has been recently given to Machine Learning (ML) techniques in many different application fields. This paper provides a vision of what ML can do in Power Line Communications (PLC). We first and briefly describe classical formulations of the ML, and distinguish deterministic from statistical learning models with relevance to communications. We then discuss ML applications in PLC for each layer, namely, for characterization and modeling, for the development of physical layer algorithms, for media access control and networking. Finally, other applications of the PLC that can benefit from the usage of ML, as grid diagnostics, are analyzed. Illustrative numerical examples are reported to serve the purpose of validating the ideas and motivate future research endeavors in this stimulating signal/data processing field.

INDEX TERMS Machine learning, statistical learning, communications, power line communications, channel modeling, physical layer, MAC layer, network layer, grid diagnostics.

I. INTRODUCTION

Modern communication systems have reached a high degree of performance, meeting demanding requirements in numerous application fields. Significant progress in the analysis and design of communication systems has been rendered possible by the milestone work of Shannon [1] that provided a methodological approach to attack the challenge of reliably transmitting information through a given communication mean. Shannon's mathematical approach suggests to represent the system as a chain of blocks mathematically modeled, namely, the transmitter, the channel, and the receiver. The transmitter can be further divided into a source coder, a channel coder and a signal modulator. The channel is represented by a transfer function (in most cases considered linear time invariant, or time variant) and an additive noise term. Three generations of scientists and engineers grew up with this mathematical mindset which provided tools to acquire domain knowledge and use it to build a model for each block, so that the overall behavior becomes known. Such a framework intrinsically has the advantage that each block can be individually studied and optimized. We would refer to this approach as *physical* and *bottom-up*.

From an epistemological point of view, the *mathematical theory of communications* is based on knowledge coming

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

from *a priori* justifications and relying on intuitions and the nature of these intuitions, which is intrinsically what mathematics does. On the contrary, *a posteriori* knowledge is created by what is known from experience, therefore generated afterwards with an *empirical* and *top-down* approach. The immense contributions of I. Kant with his philosophy of transcendental aesthetics and logic [2], made a step further into the understanding and the definition of knowledge: knowledge of the structure of time and space and their relationships, is *a priori* knowledge; knowledge acquired from observations is *a posteriori* knowledge; most of our knowledge comes from the process of learning and observing phenomena, and without a priori knowledge it is impossible to reach the true knowledge. In this respect, Machine Learning (ML) [3], [4], which is the topic of this paper, can be considered an implementation by humans of techniques in machines to acquire knowledge from *a posteriori* observations of natural phenomena. The origin of success of ML relies on its ability to derive relations among phenomena and potentially discover the hidden (latent) state of a system, i.e. potentially provide an intrinsic *true* knowledge of the system. System identification and model based design through the aid of ML [5] constitute a first step to find undiscovered system properties via a mixed *a priori* - *a posteriori* learning approach, which, retrospectively, follows Kant's philosophical structure.

ML is indeed bringing new lymph in the domain of communication systems modeling, design, optimization,

and management. It provides a paradigm shift: rather than concentrating on a physical bottom-up description of the communication scheme, ML aims to learn and capture information from a collection of data, to derive the input-output relations of the system, or of a given task in the system. We would argue that a miraculous solution of communications challenges with ML does not yet exist. In addition, what is learned via ML tools is not necessarily representative of the physical reality, i.e., wrong beliefs about relations and dependencies among data may be generated. Consequently, the results have to be validated through the support of a probabilistic approach and an understanding of the system physics. But the path has been mapped out: ML offers a great deal of opportunities to research and design communication systems.

A. BUT WHAT ARE THE DOMAINS OF APPLICATION OF ML IN COMMUNICATIONS?

The applications of ML in communications are a multitude and cover all three fundamental protocol stacks: the physical layer, the MAC layer and the network layer. More specifically, the same applies to Power Line Communications (PLC) which is the technology that exploits the existing power deliver infrastructure to convey information signals [6]. When things become complex and a bottom-up model is difficult to derive or has too many uncertainties, ML can help, no doubt. This is particularly true in PLC since, despite the advances in channel and noise modeling [7], the communication media is still not fully understood and modeled especially when it comes to noise and interference. Consequently, the transceiver techniques designed so far might not be optimal [8]. Media access control and resource allocation in massive PLC networks (as smart metering ones) is an extremely complex task that can benefit from ML approaches [9]. The analysis of PLC signals exchanged among nodes can reveal properties of the grid status and detect anomalies in the cables, loads and generators, which is relevant for grid predictive maintenance [10], [11].

B. PAPER CONTRIBUTION

In this paper, we will discuss in detail the application of ML in PLC providing concrete insights (tips) of what can be done and with what ML tools (tricks). Several numerical examples are reported to validate the ideas and to stimulate further work in this research domain. To better understand ML, we start our short journey into ML for PLC by proving a compact introduction to ML with focus on applications in communications. This serves also the purpose of surveying, at the best of the authors knowledge, the existing literature on the topic and the initial studies conducted.

In detail, the paper is organized as follows. In Sec. II, ML fundamentals for both supervised and unsupervised learning are reported. Specific tools are described. They include artificial neural networks, and support vector machines (for supervised ML), and clustering, autoencoders, and generative networks (for unsupervised ML). Convolutional and

Recurrent Neural Networks as well as Reinforcement Learning are also briefly discussed. In Sec. III, we focus on PLC and ML for the characterization of the medium and its modeling through the use of a data driven, synthetic approach. In Sec. IV, ML for physical layer PLC is discussed while the MAC and network layers are considered in Sec. V. Other applications of ML for PLC are the topic of Sec. VI. The conclusions then follow.

II. MACHINE LEARNING BASICS FOR COMMUNICATIONS

Following the definition provided by Mitchell [12], ML algorithms can be categorized according to the learning process (the kind of experience E the machine has), the specific task T and the performance measure P . If we focus on the experience, learning is divided into *supervised*, *unsupervised* and *reinforcement* learning. According to different interpretations of the experience, one could distinguish between a *deterministic* or *probabilistic* interpretation/description of the system. Since in communication theory, signals are always studied as stochastic processes, we prefer to encapsulate ML tools into *statistical learning* tools, thus, under a stochastic/probabilistic framework.

Now, as discussed in the introduction, the paradigm shift in ML for communications is to frame the problem into an initial black-box description of the system and then learn and capture information from a collection of data so that a target task can be realized [13]. Examples of tasks, are system modeling, or more in general system identification, data detection, resource allocation, network management etc.

In the following, we summarize learning methods and tools and we report specific examples of ML based solutions already proposed in the literature using however a unified description approach.

A. SUPERVISED LEARNING

1) PRELIMINARIES AND DEFINITIONS

Let $(\mathbf{x}_i, \mathbf{y}_i) \sim p(\mathbf{x}, \mathbf{y})$, $i = 1, \dots, N$, be samples collected into a training set \mathcal{D} belonging to the joint distribution (pdf) $p(\mathbf{x}, \mathbf{y})$. Supervised learning, under a deterministic model, aims to find a mapping between all pairs of input-output vectors $(\mathbf{x}_i, \mathbf{y}_i)$, thus, an inferred function F that element-wise satisfies $\mathbf{y} = F(\mathbf{x})$. The ideal scenario would map unseen samples $\tilde{\mathbf{x}}$ into the right, initially unknown, label/target $\tilde{\mathbf{y}}$. As we want to address the problem under a stochastic/probabilistic approach, we state that probabilistic supervised learning tries to predict \mathbf{y} from \mathbf{x} by estimating $p(\mathbf{y}|\mathbf{x})$ under a *discriminative model* or by estimating the joint distribution $p(\mathbf{x}, \mathbf{y})$ under a *generative model*. Fig. 1 schematically distinguishes between deterministic and probabilistic approaches in ML.

If the outputs are continuous variables, we consider it as a *regression* problem, while if the targets are discrete, then we have a *classification* problem. A standard way to proceed during the learning process is to define a cost function C , namely a performance measure that evaluates the quality

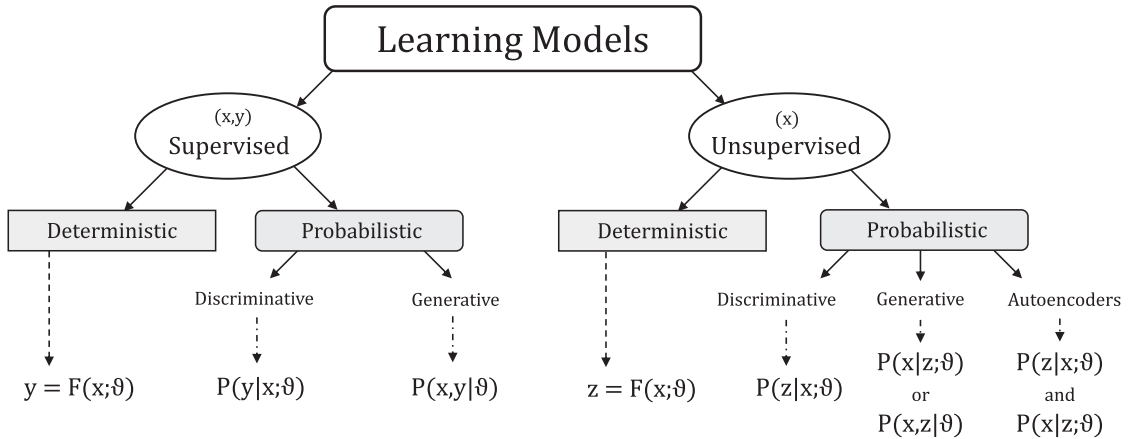


FIGURE 1. Taxonomy of learning models.

of our prediction \hat{y} . In most applications, we can rely only on the observed dataset \mathcal{D} and derive an empirical sample distribution since we do not have knowledge of the true joint distribution $p(\mathbf{x}, \mathbf{y})$. In particular, the goal of the training process is to minimize

$$C(\hat{\mathbf{y}}) = \mathbb{E}_{(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}}[\delta(\mathbf{y}, \hat{\mathbf{y}})] \quad (1)$$

where δ is a measure of distance between the wanted target \mathbf{y} and the prediction $\hat{\mathbf{y}}$, and \mathbb{E} denotes expectation.

2) TOOLS: NEURAL NETWORKS

Neural Networks (NNs) are among the most popular tools in this field since they are known being universal function approximators [14], they can be implemented in parallel on concurrent architectures and most importantly, they can be trained by backpropagation [15].

A feedforward neural network with L layers maps a given input $\mathbf{x}_0 \in \mathbb{R}^{D_0}$ to an output $\mathbf{x}_L \in \mathbb{R}^{D_L}$ by implementing a function $F(\mathbf{x}_0; \theta)$ where θ represents the parameters of the NN. To do so, the input is processed through L iterative steps

$$\mathbf{x}_l = f_l(\mathbf{x}_{l-1}; \theta_l), \quad l = 1, \dots, L \quad (2)$$

where $f_l(\mathbf{x}_{l-1}; \theta_l)$ maps the input of the l -th layer to its output. The most used layer is the fully connected one, whose mapping is expressed as

$$f_l(\mathbf{x}_{l-1}; \theta_l) = \sigma(\mathbf{W}_l \cdot \mathbf{x}_{l-1} + \mathbf{b}_l) \quad (3)$$

where $\sigma(\cdot)$ is the activation function while \mathbf{W}_l and \mathbf{b}_l are the parameters, weights and the biases, respectively. According to the specific application, several different types of layers and activation functions can be defined. Fig. 2 shows a general fully-connected architecture.

Defined a metric δ and a cost function C , the easiest and most classical algorithm to find the feasible set of parameters θ is the gradient descent which iteratively updates θ as $\theta_t = \theta_{t-1} - \eta \nabla C(\theta_{t-1})$ where η is the learning rate. Its popular variants are Stochastic Gradient Descent (SGD) and adaptive learning rates (Adam) [16]. Common choices for the cost

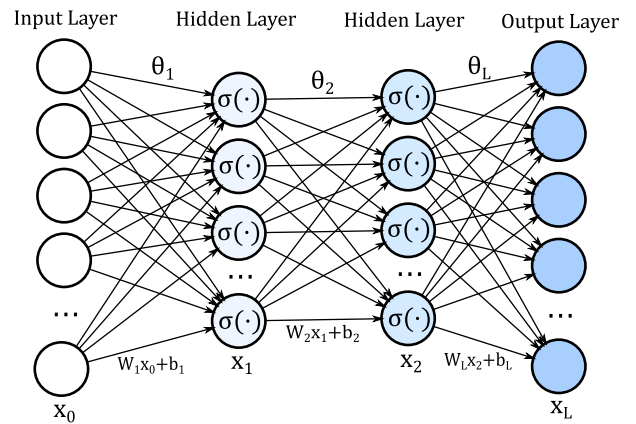


FIGURE 2. Structure of a fully connected neural network with 2 hidden layers.

function, in a deterministic model, are the mean squared error and categorical cross entropy, for which δ in (1) takes the form $\delta(\mathbf{y}, \hat{\mathbf{y}}) = \|\mathbf{y} - \hat{\mathbf{y}}\|^2$ and $\delta(\mathbf{y}, \hat{\mathbf{y}}) = -\mathbf{y} \cdot \log \hat{\mathbf{y}}$, respectively.

In the context of probabilistic discriminative models, the fundamental learning criterion is the *maximum likelihood estimator* which finds a value of θ in a parameterized family of models $p(\mathbf{y}|\mathbf{x}; \theta)$ that is the most likely to have generated the observed data \mathcal{D} , formally

$$\theta = \arg \max_{\theta} p(\mathbf{y}|\mathbf{x}; \theta). \quad (4)$$

From the dataset samples and using the log-likelihood function, (4) can be empirically estimated as

$$\theta = \arg \max_{\theta} \sum_{i=1}^N \log p(\mathbf{y}_i|\mathbf{x}_i; \theta). \quad (5)$$

Note that the deterministic and probabilistic approaches collapse together when we consider Gaussian or categorical models since we will end up solving a least squares or a cross-entropy minimization problem, respectively. We leave the description of generative models for Sec. II-B since they

are typically and formally introduced under unsupervised learning, and then they have been extended to supervised versions. This is the reason why they can be cast in hybrid models.

NNs are only one of the several different tools introduced so far by the ML's community. Since they are excellent in finding patterns, in the context of PLC, as we will discuss, they can be used to find deterministic and probabilistic relationships between physical measured quantities such as the line impedance, the channel transfer function, the radiated power. Channel emulation can be realized with neural networks, an early example of which is [17]. NNs can also be used for prediction and network management purposes.

3) TOOLS: SUPPORT VECTOR MACHINES

Another relevant approach to supervised learning is the usage of a Support Vector Machine (SVM) [18], [19]. We briefly describe the SVM classifier since regression follows from the same idea with minor modifications. The SVM classifier predicts that a certain sample \mathbf{x}_i belongs to a class y_i if $\mathbf{w}^T \mathbf{x}_i - b$ is positive, where $\mathbf{w}^T \mathbf{x}_i - b = 0$ is the equation for the decision boundary. The aim is to maximize the distance $d = 2/\|\mathbf{w}\|$ between the supporting rescaled hyperplanes $\mathbf{w}^T \mathbf{x}_i - b = 1$ and $\mathbf{w}^T \mathbf{x}_i - b = -1$ (see Fig 3). This leads to the following dual Lagrangian formulation for the linear SVM

$$\mathcal{L}_d = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \quad (6)$$

where α_i are the Lagrange multipliers (non-negatives) and N the number of samples. When data are non linearly separable in the domain space, transforming them into an higher dimensional one using the mapping $\mathbf{x} \rightarrow \phi(\mathbf{x})$ increases the complexity of the classifier. A concept called *kernel trick* overcomes this problem; the key idea behind the kernel trick is that the inner product between \mathbf{x}_i and \mathbf{x}_j is similar to the inner product between $\phi(\mathbf{x}_i)$ and $\phi(\mathbf{x}_j)$. Moreover, since we are not interested in knowing ϕ but only the scalar product $\mathbf{x}_i \cdot \mathbf{x}_j$, such product is represented by the kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$. Consequently, we can make predictions using the function

$$f(\mathbf{x}) = b + \sum_i \alpha_i K(\mathbf{x}, \mathbf{x}_i), \quad (7)$$

which is a non-linear function w.r.t. \mathbf{x} but linear w.r.t. $\phi(\mathbf{x})$ and α . The most commonly used kernel is the radial basis function (RBF) kernel with expression

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^2}\right). \quad (8)$$

As an example, in the context of PLC, SVM was recently explored to monitor and detect cable degradations [20].

4) OTHER TOOLS

Another very simple non-probabilistic supervised learning algorithm is k -nearest neighbors. It is a non-parametric algorithm in the sense that, since there is no learning process,

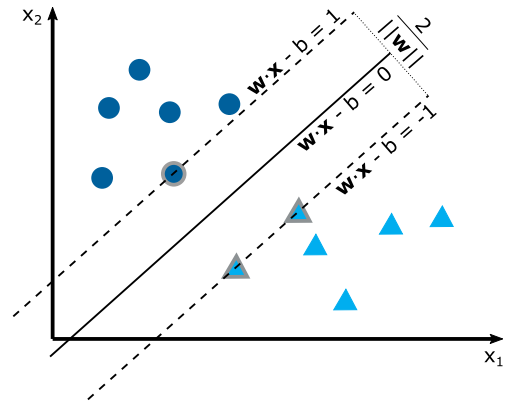


FIGURE 3. Decision boundaries and maximum separation for SVM framework.

at the test time we predict y_i for a given test input \mathbf{x}_i . The main idea is to find the k -nearest neighbors (for example using an Euclidean metric) to \mathbf{x}_i in the training set \mathbf{x} . The algorithm returns the average, if regression is considered, and the mode, if classification is considered, of the corresponding y , also called training class. This algorithm is very easy to implement but it becomes significantly slower as the number of samples increases.

Naive Bayes, decision trees, and random forest [21] are other popular examples of supervised learning algorithms.

5) APPLICATION OF SUPERVISED LEARNING TO COMMUNICATIONS

Supervised learning algorithms have been applied in communications scenarios with promising results with focus from physical layer to networking at the edge or at the cloud segment.

- *Physical Layer:* At the transmitter, the problem of maximizing energy efficiency has been addressed via resource allocation with deep neural networks [22]. At the receiver, several problems have been considered such as data detection and decoding [23], channel estimation and symbol detection in an OFDM system [24], linear codes decoding [25], decoding for non-linear channels like those in optical communications [26], modulation recognition [27]. In addition, also radio-localization using discriminant adaptive neural networks has been proposed in [28];
- *Link and Medium Access Control Layer:* feedback on the decodability of the received signal has been proposed using ML techniques in [29]; spectrum sensing and allocation in cognitive networks in presence of interfering devices has been done in [30]; interference management was considered in [31] and estimation of the number of active nodes in a wireless network testing various ML techniques was the objective of [32].
- *Network and Application Layers:* At the edge, caching popular contents in echo state networks exploiting a ML framework was investigated in [33]; at the cloud,

a network controller was designed for routing using a predictive approach [34]; lastly, a survey of ML techniques for traffic classification was offered in [35].

B. UNSUPERVISED LEARNING

1) PRELIMINARIES AND DEFINITIONS

Let $\mathbf{x}_i \sim p(\mathbf{x})$, $i = 1, \dots, N$, be samples collected into a training set \mathcal{D} belonging to the probability distribution function (PDF) $p(\mathbf{x})$. Unsupervised learning aims to find useful properties of the structure of a dataset \mathcal{D} , ideally inferring the true unknown distribution $p(\mathbf{x})$. Several different tasks are solved using unsupervised learning, roles such as: clustering, which divides the data into cluster of similar samples, feature extraction, which transforms data in a different latent space easier to handle and interpret, density estimation and generation/synthesis of new samples, whose objective is to learn, from data in \mathcal{D} , the distribution $p(\mathbf{x})$ and to produce new unseen samples from it. Contrarily to the supervised one, unsupervised learning does not have an unified accepted formulation. This is even clearer if we think that many unsupervised learning tasks require the introduction or the presence of an hidden variable \mathbf{z}_i for each sample, leading to the selection of different models under a probabilistic approach (as shown in Fig. 1):

- *Discriminative models*, where the latent code \mathbf{z}_i has to be extracted from data \mathbf{x}_i by defining a family $p(\mathbf{z}|\mathbf{x}; \theta)$ parameterized by a vector θ .
- *Autoencoders*, where \mathbf{x}_i is encoded into a latent variable \mathbf{z}_i so that recovering \mathbf{x}_i from \mathbf{z}_i is possible through a decoder. This model, as discussed in Sec. II-B.3, is very popular nowadays and can be interpreted also under a supervised learning approach, where \mathbf{x}_i is the input data, and \mathbf{x}_i is itself the corresponding target. In the classical deterministic approach, the problem consists of the parameterization of two functions, namely the encoder $\mathbf{z} = F(\mathbf{x}; \theta)$, and the decoder $\mathbf{x} = G(\mathbf{z}; \theta)$. In the probabilistic approach, the encoder has to model $p(\mathbf{z}|\mathbf{x}; \theta)$, while the decoder models $p(\mathbf{x}|\mathbf{z}; \theta)$.
- *Generative models*, where there exists an hidden \mathbf{z}_i that generates the observation \mathbf{x}_i . After a specification of a parameterized family $p(\mathbf{z}|\theta)$, the distribution of the observation can be rewritten as $p(\mathbf{x}|\theta) = \sum_{\mathbf{z}} p(\mathbf{z}|\theta)p(\mathbf{x}|\mathbf{z}; \theta)$.

For a fully and exhaustive description of the different models and the corresponding tools like Expectation Maximization (EM) and Evidence Lower Bound (ELBO), the interested reader is referred to [36]. We will focus only on clustering algorithms for discriminative models and autoencoders since they are now getting a lot of attention by communication researchers.

2) TOOLS: CLUSTERING

Clustering analysis includes different algorithms whose common task is to discover the hidden unknown structure and relationships between input data. Several algorithms are

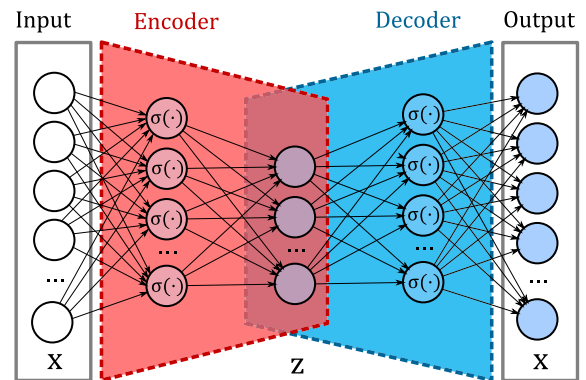


FIGURE 4. Autoencoder block with neural networks. A latent representation \mathbf{z} is learned by the encoder and used as input of the decoder for recovering \mathbf{x} .

available in the literature for this purpose. The most common and reliable ones are K-means, Hierarchical Clustering (HC), clustering using representatives (CURE), and Self Organizing Maps (SOM).

K-means is a clustering algorithm that allows to partition the input data in K sets, based on a certain metric exploiting expectation maximization algorithms. K-means clustering, considers a set of N data points in a D -dimensional space \mathbb{R}^D and an integer K . The problem is to determine a set of K points in \mathbb{R}^D , called centroids, so as to minimize the mean squared distance from each data point to its nearest centroid [37]. Usually, this clustering algorithm is solved efficiently with heuristic methods such as Lloyd’s.

HC algorithms are mainly classified into agglomerative methods (bottom-up methods) and divisive methods (top-down methods), depending on how the hierarchical dataset is formed [38]. Multiple implementations and variations of the HC algorithms are described in the literature [39]. Essentially, they follow a procedure where a tree of clusters called dendrogram is generated.

The CURE algorithm attempts to uncover the cluster shape using a collection of representative points from the main dataset [40]. This algorithm is efficient for large databases, and compared to K-means clustering is more robust to outliers and able to identify clusters with complex shapes.

SOM is a grid of map units [41]. Each unit is represented by a prototype vector that defines its space position. The units are connected to the adjacent ones to form a network. The number of map units corresponds to the final number of clusters, thus, a higher number of units corresponds to a higher accuracy of data separation. A training procedure is used to stretch the network and map the space of input data.

3) TOOLS: AUTOENCODERS

An autoencoder is a particular type of artificial neural network consisting of an encoding block which tries to learn a latent representation \mathbf{z} , typically in a lower-dimensional space, of the input variable \mathbf{x} , and a decoding block which reconstructs \mathbf{x} at the output exploiting the code \mathbf{z} (Fig. 4).

A typical learning formulation for deterministic autoencoders requires to solve

$$\theta_{\text{opt}} = \arg \min_{\theta} \delta(\mathbf{x}, G(F(\mathbf{x}; \theta_1); \theta_2)), \quad (9)$$

where δ is a distance measure, $\theta = (\theta_1, \theta_2)$, while F and G stand for the encoder and decoder function, respectively. When F and G are linear functions, we get the Principal Component Analysis (PCA). Given a set/matrix \mathbf{x} of N D -dimensional samples $\mathbf{x}_i \in \mathbb{R}^D$, PCA sets the encoder as $F(\mathbf{x}; \theta) = \mathbf{W}^T \mathbf{x}$ and the decoder as $G(\mathbf{z}; \theta) = \mathbf{W} \mathbf{z}$ where \mathbf{W} is the unknown parameter, a $D \times M$ matrix where M is the dimension of the latent space. If δ is the quadratic loss function, then (9) can be rewritten as

$$\mathbf{W}_{\text{opt}} = \arg \min_{\mathbf{W}} \sum_{i=1}^N \|\mathbf{x}_i - \mathbf{W} \mathbf{W}^T \mathbf{x}_i\|^2. \quad (10)$$

Let Σ be the sample covariance matrix of \mathbf{x} , then \mathbf{W} is given by the M principal eigenvectors of Σ . The geometric idea in PCA consists of finding a rotation of the domain space that aligns the principal axes of variance with the basis of the new representation latent space associated with \mathbf{z} . Under a probabilistic model, the resulting transform is a vector \mathbf{z} whose elements are mutually uncorrelated. PCA is broadly used also as a feature extractor, something which is very useful for the analysis of both the channel response and the noise in PLC, for instance.

Correlation is an indicator of linear dependence, but in most cases we are interested in representations where features have a different form of dependence. For example, in generative networks, Dinh [42], [43] proposed a non-linear deterministic transform of the data which maps them into a latent space of independent variables where the probability density results tractable. Another interesting and suitable type of autoencoder for communications is the Denoising AutoEncoder (DAE) [44]. The idea is to train a machine in order to minimize the following denoising criterion:

$$\mathcal{L}_{\text{DAE}} = \mathbb{E}_{\tilde{\mathbf{x}} \sim \mathcal{D}} [\delta(\mathbf{x}, G(F(\tilde{\mathbf{x}})))] , \quad (11)$$

where $\tilde{\mathbf{x}}$ is a stochastic corruption of \mathbf{x} . When we train a DAE using the expected quadratic loss and a corruption noise $\tilde{\mathbf{x}} = \mathbf{x} + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma I)$, Alain and Bengio [45] proved that the autoencoder recovers properties of the training density $p(\mathbf{x})$. They also showed that the DAE with a small noise corruption of variance σ^2 is similar to a Contractive AutoEncoder (CAE) with penalty coefficient $\lambda = \sigma^2$. The contractive autoencoder [46] is a particular form of regularized autoencoder which is trained in order to minimize the following reconstruction criterion:

$$\mathcal{L}_{\text{CAE}} = \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} \left[\delta(\mathbf{x}, G(F(\mathbf{x}))) + \lambda \left\| \frac{\partial F(\mathbf{x})}{\partial \mathbf{x}} \right\|_F^2 \right], \quad (12)$$

where $\|\mathbf{A}\|_F$ is the Frobenius norm. The idea behind CAE is that the regularization term attempts to make $F(\cdot)$ or $G(F(\cdot))$ as simple as possible, but at the same time the reconstruction

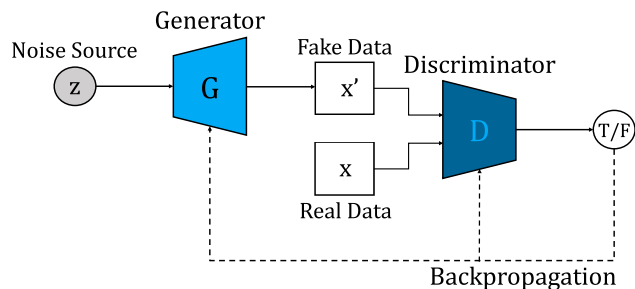


FIGURE 5. GAN framework in which the generator and discriminator are learned during the training process.

error must be small. Typically, autoencoders find a low-dimensional representation of the input vector \mathbf{x} at some intermediate level.

Now, in the domain of communications it has been proposed in [47] to learn a robust representation \mathbf{z} of the input \mathbf{x} in order to overcome channel perturbations (noise, fading, distortion, etc.). In this way the transmitted signal can be recovered with small probability of error exploiting the *redundancy* learned by the autoencoder. The findings in [45] are somehow remarkable when applied in a communications framework because they assert that corrupting the transmitted signal with some form of noise can be beneficial for the autoencoder in the reconstruction's phase.

4) TOOLS: GENERATIVE NETWORKS

Autoencoders can be studied also as generative models. Based on variational inference, Kingma and Welling introduced in [48] the concept of variational autoencoders. Denoted \mathbf{z} as the latent variable of the observed value \mathbf{x} for a parameter θ_1 , then $p(\mathbf{z}|\mathbf{x}; \theta_1)$ represents the intractable true posterior which can be approximated by a tractable one, $q(\mathbf{z}|\mathbf{x}; \phi)$, for a parameter ϕ . A probabilistic encoder produces $q(\mathbf{z}|\mathbf{x}; \phi)$ while a probabilistic decoder produces $p(\mathbf{x}|\mathbf{z}; \theta_2)$. Rather than outputting the code \mathbf{z} , the encoder outputs parameters describing a distribution for each dimension of the latent space. When the prior is assumed to be Gaussian, \mathbf{z} will consist of mean and variance. Tuning the parameters in the latent space, and passing the new latent samples through the decoder is a way to generate new data.

Finally, in the generative models framework, it is worth mentioning Generative Adversarial Networks (GAN) proposed by Goodfellow *et al.* [49]. The main idea is to train a pair of networks in competition with each other (Fig. 5): a generator model G that captures the data distribution and a discriminator model D that distinguishes if a sample is a true sample coming from real data rather than a fake one coming from data generated by G . The training procedure for G is to maximize the probability of D making a mistake. GANs can be thought as a minimax two-player game which will end when a Nash equilibrium point is reached. Given an input noise vector \mathbf{y} with distribution $p_{\text{noise}}(\mathbf{y})$, the map into data space is achieved through $G(\mathbf{y}; \theta_{\text{gen}})$. Defining the following

value function

$$V(G, D) = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{y} \sim p_{\text{noise}}(\mathbf{y})}[1 - \log D(G(\mathbf{y}))], \quad (13)$$

it was shown that the generator implicitly learns the true distribution since the equilibrium is reached when $p_{\text{gen}} = p_{\text{data}}$.

This generation technique has several applications in communication theory like stochastic channel modeling and channel coding [50]. As we will discuss in the following (Sec. III-B), it has also application in PLC due to the ability of generating synthetic data from a given distribution, thus, modeling for example the PLC channel response or noise.

5) APPLICATION TO COMMUNICATIONS

Herein, we review some recent applications of unsupervised learning to communication systems for different layers of the protocol stack.

- *Physical Layer:* Autoencoders for deep learning were firstly interpreted as a communication system in [47]. The application of unsupervised learning to end-to-end communications can be divided according to the presence or not of an a priori knowledge of the channel model for the training part. If the channel model is available, the usage of autoencoders with deep learning can help to design an optimal encoding/decoding procedure. For instance applications are found in optical fiber communications [51], in AWGN channels with unreliable feedback [52], sparse code multiple access schemes where deep neural networks are used to learn a decoding strategy to minimize bit-error rate [53]. When Channel State Information (CSI) is available, channel charting [54] for localization exploiting CSI has been realized using autoencoders. If the channel model is unknown, several new approaches have been recently proposed. In order to design a radio communication system with autoencoders and to overcome the lack of channel knowledge, a two-phase training strategy was followed in [55]. For the same objective, [56] and [57] utilized stochastic perturbation and policy gradients techniques, respectively. Generative models can be used to generate samples of a given communication channel, in particular GANs have been exploited in [50] and [58].
- *Link and Medium Access Control Layer:* Spectrum sensing exploiting GANs and resource management for LTE were studied in [59] and [60], respectively.
- *Network and Application Layers:* Clustering algorithms are the fundamental tools to be used in the network and application layers. Routing can be improved via clustering [61], while in the application layer, identification of clustering communities, which is a social networks goal, has been considered in [62].

C. OTHER LEARNING SCHEMES

In the following sections, we briefly discuss Convolutional Neural Networks (CNNs), Recurrent Neural

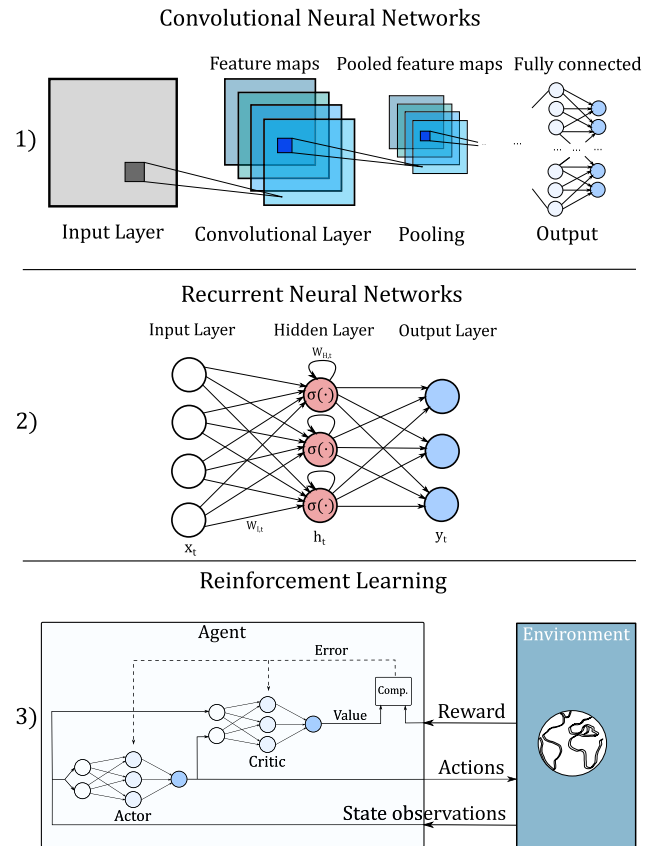


FIGURE 6. Graphical learning models of: 1) Convolutional Neural Network with one convolutional layer; 2) Recurrent Neural Network with simple feedback loop; 3) Reinforcement learning setting for actor-critic method.

Networks (RNNs) and the Reinforcement Learning (RL) approach to offer an exhaustive overview of other ML tools that may have application in PLC.

1) CONVOLUTIONAL NEURAL NETWORKS

Convolutional Neural Networks (CNNs) (Fig. 6.1) are able to capture the spatial and temporal dependencies of data, and for this reason they find application in image and document recognition [63], medical image analysis [64], natural language processing [65], and more in general pattern recognition. CNNs are Multi Layer Perceptrons (MLP) (Sec. II-A.2) with a regularization approach since they consist of multiple convolutional layers to ensure the translation invariance characteristics. In particular, given an input data matrix \mathbf{I}_i , the feature map \mathbf{F}_j is obtained as

$$\mathbf{F}_j = \sigma \left(\sum_{i=1}^C \mathbf{I}_i * \mathbf{K}_{i,j} + \mathbf{B}_j \right) \quad (14)$$

namely, through the superposition of C layers, e.g., $C = 3$ for RGB images, each comprising a convolution between the input matrix \mathbf{I}_i and a kernel matrix $\mathbf{K}_{i,j}$, plus an additive bias term \mathbf{B}_j , and a final application of non-linear activation function $\sigma(\cdot)$, typically, a *sigmoid*, or *tanh*, or *ReLU*. Each set of kernel matrices represents a filter that extracts local features.

To control the problem of over fitting, the dimension of data and features to be extracted is reduced by pooling layers. Finally, fully-connected layers are used to extract semantic information from features.

2) RECURRENT NEURAL NETWORKS

Recurrent Neural Networks (RNNs) are a class of supervised and unsupervised ML tools that are capable of representing time dependencies. Since they introduce the notion of time into the model, they have been successfully applied to tasks such as speech [66] and handwriting recognition [67], health care [68], and machine translation [69]. Bearing in mind the architecture of the NNs presented in Sec. II-A.2, RNNs can be seen as feedforward NNs made of artificial neurons with one or more feedback loops (Fig. 6.2).

If we consider a simple RNN made of the input layer, an hidden layer, and the output layer, then the hidden layer or memory of the system \mathbf{h}_t , can be obtained as

$$\mathbf{h}_t = \sigma(\mathbf{W}_{I,t} \cdot \mathbf{x}_t + \mathbf{W}_{H,t} \cdot \mathbf{h}_{t-1} + \mathbf{b}_t) \quad (15)$$

where \mathbf{x}_t is a real-valued input vector available at time t , $\sigma(\cdot)$ is the activation function, while $\mathbf{W}_{I,t}$, $\mathbf{W}_{H,t}$ and \mathbf{b}_t are the weight matrix connecting the input to the hidden layer, and the weight matrix connecting the previous hidden layer to the current one and the biases, respectively. At time t , the hidden layer \mathbf{h}_t is influenced by the current input \mathbf{x}_t but also by the previous hidden state \mathbf{h}_{t-1} , such that the output \mathbf{y}_t depends on the evolution over time of the hidden layer.

Several different architectures for RNNs have been proposed such as Long Short-Term Memory (LSTM) [70] and Bidirectional RNNs [71]. For further details and a wide survey on RNNs, the interested reader is referred to [72] and [73], respectively.

3) REINFORCEMENT LEARNING

Reinforcement Learning (RL) addresses the problem of an *agent* learning to act in a dynamic *environment* by finding the best sequence of actions that maximizes a *reward* function. The basic idea is that the agent explores the interactive environment. According to the observation experience it gets, it changes his actions in order to receive higher rewards. RL finds several applications, from robots control [74], to games such as Atari [75], and Go [76].

Basic RL can be modeled as a Markov Decision Process (MDP). Let S_t be the observation (or state) provided to the agent at time t . The agent reacts by selecting an action A_t to obtain from the environment the updated reward R_{t+1} , the discount γ_{t+1} , and the next state S_{t+1} . In particular, the agent-environment interaction is formalized by a tuple $\langle \mathcal{S}, \mathcal{A}, T, r, \gamma \rangle$ where \mathcal{S} is a finite set of states, \mathcal{A} is a finite set of actions, $T(s, a, s') = P[S_{t+1} = s' | S_t = s, A_t = a]$ is the transition probability from state s to state s' under the action a , $r(s, a) = \mathbb{E}[R_{t+1} | S_t = s, A_t = a]$ is the reward function, and $\gamma \in [0, 1]$ is a discount factor. To find out which actions are good, the agent builds a *policy*, i.e., a map $\pi : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ that defines the probability

of taking an action a when the state is s . If we denote with $G_t = \sum_{k=0}^{\infty} \left(\prod_{i=1}^k \gamma_{t+i} \right) R_{t+k+1}$ the discount return, then the goal of the agent is to maximize the expected discount return (*value*) $q^\pi(s, a) = \mathbb{E}_\pi[G_t | S_t = s, A_t = a]$, by finding a good policy $\pi(s, a)$. When the state or action sets are defined in high-dimensional spaces, the policy π and the value q can be represented by deep neural networks (Sec. II-A.2).

With some more detail, there are essentially three different RL algorithms [77], [78]: a) policy based methods, when the agent, given the observation as input, optimizes the policy π without using a value function q ; b) value based methods, when the agent, given the observation and the action as inputs, learns a value function q ; c) actor critic methods, where a *critic* measures how good the action taken is (value-based), and an *actor* controls the behaviour of the agent (policy-based) (Fig. 6.3).

4) APPLICATION TO COMMUNICATIONS

In [79], CNNs naively extract features and use them for radio modulation recognition on time series of radio signals. A similar approach is adopted in [80] where CNNs automatically detect and identify frequency domain signatures for the IEEE 802.x standard compliant technologies. RNNs have been firstly introduced as channel equalizers in [81], while recently in [82], LSTM RNNs have been applied for network traffic anomaly detection. Dynamic spectrum assignment in PLC has been investigated in [83] under a RL approach. Due to the dynamic environment, RL finds several applications in communications and networking, mostly covered in [84].

III. ML FOR PLC MEDIUM CHARACTERIZATION AND MODELING

In the following sections. We will mostly highlight the relevance and applications of the supervised and unsupervised ML tools described in Sec. II-A and Sec. II-B in this context. Reinforcement learning will also be briefly discussed in the context of resource allocation in PLC.

We start from medium characterization since this is a key topic in PLC. It includes the analysis of the channel response (in time and frequency domain), the line impedance, the noise and interference [7].

A. STATISTICAL LEARNING

Up to date studies, have focused on the statistical characterization of the PLC channel and noise. For what matters the channel, datasets have been collected in several scenarios, such as in-home [85]–[88], outdoor-access [89], [90] and in-vehicle (cars, ships, planes) [91]–[95]. A great deal of knowledge has been acquired (mostly for the broadband channel in the in-home scenario) and some consensus has been reached on the main characteristics. For instance, the frequency response and the average channel gain have a log-normal distribution [96], [97]. Periodic channel time variations are exhibited especially in the narrow band spectrum, or for frequencies up to 10 MHz [98]. Correlation is

TABLE 1. Features description.

ID	Feature name	Equation	Description
1	maxAbs	$\max_j \{ s_j \}$	maximum of the absolute values
2	sum	$\sum_{j=1}^N s_j$	sum of the samples
3	sum2	$\sum_{j=1}^N s_j^2$	sum of the squared samples
4	std	$\sqrt{\frac{1}{N-1} \sum_{j=1}^N s_j - \mu ^2}$	standard deviation of the samples
5	skew	$\frac{\frac{1}{N} \sum_{j=1}^N (s_j - \mu)^3}{\left(\sqrt{\frac{1}{N} \sum_{j=1}^N (s_j - \mu)^2} \right)^2}$	skewness of the samples
6	kurt	$\frac{\frac{1}{N} \sum_{j=1}^N (s_j - \mu)^4}{\left(\sqrt{\frac{1}{N} \sum_{j=1}^N (s_j - \mu)^2} \right)^2}$	kurtosis of the samples
7	pears	$r_{CH1,2} = \frac{\sum_{j=1}^N (s_{CH1,j} - \mu_{CH1})(s_{CH2,j} - \mu_{CH2})}{\sqrt{\sum_{j=1}^N (s_{CH1,j} - \mu_{CH1})^2 \sum_{j=1}^N (s_{CH2,j} - \mu_{CH2})^2}}$	Pearson correlation of the samples
8	dist	$\ s_{CH1} - s_{CH2}\ = \sqrt{\sum_{j=1}^N s_{CH1,j} - s_{CH2,j} ^2}$	distance between data of channel 1 and 2
9	dCor	[106]	distance correlation
10	ent	[107]	entropy approximation
11	diffEnt	[107]	entropy approximation of CH1-CH2
12	sumEnt	[107]	entropy approximation of CH1+CH2
13	fPeak	—	find the number of samples over a certain voltage level
14	fEnFr	Exploiting the power spectral density (PSD) estimated via Burg's method. [108]	find the energy of samples in a certain frequency range
15	fdist	—	find the distance in samples between the two highest peaks
16	corrStd	—	standard deviation of the correlation of two channels
17	corrSkew	—	skewness of the correlation of two channels
18	corrKurt	—	kurtosis of the correlation of two channels

In this table, s_j stands for the sample of index j , μ is the samples mean, and N is the number of samples.

exhibited among the multi-conductor channels in MIMO setups [99].

Noise has a very complex nature in PLC since it comprises not only thermal noise components but also (and mostly) components generated by active loads and electronic circuits via direct coupling and conduction, e.g., harmonics and emissions from power converters, or inductive/EM coupling especially at high frequencies [100]. Noise is extremely heterogeneous and temporal dependent. As such, its modeling is challenging. The existing description relies on a physical phenomena classification into stationary, cyclostationary, impulsive, and interference from radio transmissions that couple in the lines [101]. Only initial results for the description of the multi-conductor noise have been reported [102]–[105].

As in all communication systems, the accurate knowledge of the medium is a prerequisite for the design of a reliable and energy efficient communication technology. ML can bring new insights on the medium characterization. Firstly, it can be used to analyze and extract features. Then, to classify data from measurements. The definition of the features that contain the information necessary to separate the data into

classes is not a trivial step in the noise classification process. Aiming at obtaining a statistical understanding of the medium, the features include energy, moments of multiple order, approximated entropy and correlation. A comprehensive list of features that can be considered, is reported in Tab. 1. The knowledge about the PLC physical properties aids a lot to identify the most relevant features. For instance, the entropy is useful to recognize narrow band noise signals that are characterized by a sinusoidal behavior. The a priori knowledge that electromagnetic coupling effects among conductors may exist, suggests to study the correlation between noise traces in multiple conductor networks.

Classification tools can then be applied to determine the most prominent features and the significant number of classes. An example is PCA (Sec. II-B.3) analysis which is useful to reduce the dimensions of large datasets of variables and determine the most important features in the data.

Now, let us consider multi-conductor PLC noise. The exploitation of unsupervised learning allows the identification of noise classes. For this task, in [109], a library of features was created for a dataset of multi-conductor

narrow band (NB) noise traces obtained via measurements in the band 3-500 KHz in an office environment. Different methods were tested to extract the features (listed in Tab. 1), and then provide a classification of the noise. Features such as the distance correlation ($dCor$), the Person correlation ($pears$), or the differential entropy ($diffEnt$) are useful to highlight the relations between the noise traces. SOM [41] provided the best results among the tested algorithms. Finally, for each class a statistical analysis of the associated traces was carried out. An example of results obtained with this classification approach is shown in Fig. 7. Herein, the energy of the noise voltage trace on the first pair of conductors (neutral-ground, namely $V_{CH1}(t)$) is related to the energy of the noise trace on the second pair of conductors (neutral-phase, namely $V_{CH2}(t)$). The identified cluster PDFs are also reported in the legend of Fig. 7.

This data driven approach can classify and discover features in a more general way than the conventional approach that starts from a physical description of the phenomena generating noise, and essentially it partitions the noise into stationary, cyclostationary synchronous/asynchronous with the mains frequency, and impulsive [110].

Another example is the analysis of the line impedance and the verification of whether it has a relation with the channel response [88], [111], [112]. This is a relevant question since should this be the case, then the forward link (transmitter-receiver) channel can be inferred from the measurement of the line impedance at the transmitter node [112]. In principle, the answer to this question could be obtained with a physical analysis of the lines and loads using Transmission Line (TL) and circuit theory. Some results following this approach have been recently obtained and proved that such a dependence does not exist if not at very low frequencies and for particular circumstances where the cables and wiring structures have conductors that do not have per-unit-length cross impedance terms [112]. However, one can object that the proof has been obtained under the assumption of transverse electromagnetic propagation for which TL holds true. Therefore, in general a non-linear, complex and time dependent relation may exist. Here is where ML comes into place: to discover dependencies.

Another example is the analysis of the dependence of the line impedance at the transmitter and at the receiver. Should such a dependence exist, then physical layer security techniques could be applied to exploit such common information for the generation of secure keys without the (insecure) exchange of information between sender and recipient [113].

B. STATISTICAL-SYNTHETIC MODELING

Once the medium is characterized, next, modeling kicks in. A number of relevant results have been obtained for modeling the linear periodically time variant channel response [87]. They include so called bottom-up approaches and top-down approaches. In the former case, TL theory is applied to a deterministic or statistical description of the network topology which includes the network graph, cable types, cable

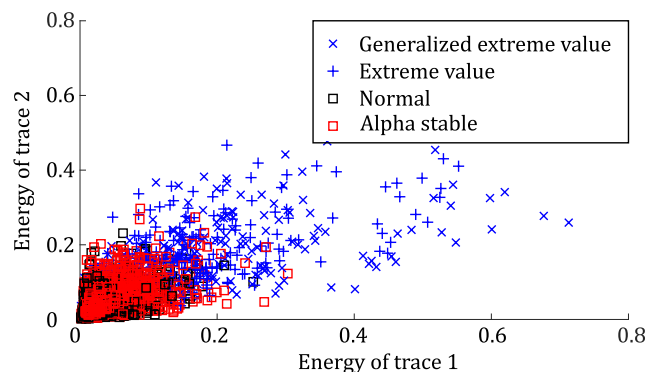


FIGURE 7. Example of noise data analyzed with SOM. The clusters are labeled with the associated PDF. The energy of the trace pairs 1 and 2 are normalized w.r.t. to their maximum.

lengths, and loads [114]. In the latter case, a mixed physical and phenomenological description of the channel response is given with the use of a parametric model derived from an abstract description of the link which accounts for multipath propagation, transmission-reflection effects, cable attenuation with length and frequency [115], [116].

Deterministic or stochastic fitting of the model with data from measurements was done to derive either a deterministic model or a random channel model. More recently, it was observed that if we had an accurate statistical description of the channel response, then a purely phenomenological model could be derived [117]. Essentially, such a model would be able to generate synthetic data that follow the observed statistics, totally abstracting from the physical interpretation of propagation of EM signals along the wires. This is essentially what ML approaches are nowadays trying to do when used as generators of *fake* data in image and sound applications. Indeed the question is how to statistically analyze data and, in a more challenging way, how to generate synthetically such distributions. The work in [117] proposed an approach which, with now an eye on ML techniques, can be further and significantly improved. For instance, GANs are an innovative tool to generate synthetic data following the same statistics of a given dataset. In particular, for communications, stochastic channel synthesis is a relevant topic since, for instance, new coding and modulation schemes require realistic channels to be effectively tested. Sec. II-B.4 describes the full methodology to design and train generative networks for this purpose. As an example, we report in Fig. 8 the result of using a GAN for the generation of the broad band PLC channel frequency response. The neural network architecture is trained with a dataset of 1000 measured single-input-single-output channel transfer functions with bandwidth of 100 MHz obtained from measurements in the in-home environment [97]. Fig. 8 shows that the synthetically generated channel frequency responses look very consistent with the measured ones and span the large dynamic range of attenuation from 10 to 90 dB.

A similar approach can be adopted on other domains, for example, GANs along with RNNs (Sec. II-C) are

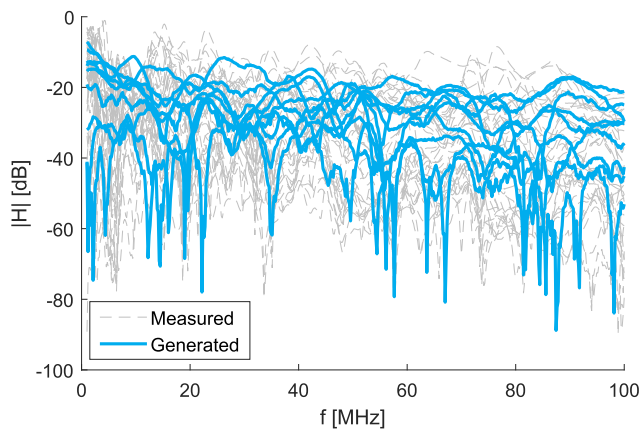


FIGURE 8. Broad-Band PLC channels measured and generated with GAN.

promising tools to generate PLC synthetic noise while taking into account time dependence. In addition, another area where ML can provide help is modeling electro-magnetic-interference and mode conversions occurring along the medium on the signal [118]. Radiation in PLC occurs at the network discontinuities and its radiation pattern depends on the length of each branch. The electromagnetic radiation of the cables can be accounted for in a per-unit-length equivalent circuit model using a series of resistances [119]. An unsupervised optimization approach (using the techniques in Sec. II-B) can be followed to obtain the optimal parameters. Alternatively, starting from an initial set of known parameters obtained for a given measurement dataset, supervised learning (using the techniques in Sec. II-A) provides a methodology for the generalization and realization of a statistical channel generator that accounts for radiation effects.

IV. ML FOR PHYSICAL LAYER PLC

PLC physical layer design [8] encompasses the tasks of designing the channel coder, the modulator and the associated algorithms at the receiver side, which includes synchronization, channel estimation, detection/equalization, interference/noise mitigation, channel decoding. In addition, precoding at the transmitter side in the form of power allocation, bit allocation, precoding, and interference alignment are also *tasks* classically relegated to the physical layer [120]–[122].

Physical layer PLC design was a vivid activity which brought innovation (partly derived from the results firstly obtained by the wireless communications community) in the form of:

- Analysis and optimization of concatenated codes, bit-interleaved codes, LDPC, Turbo codes, Fountain codes and associated decoding strategies taking into account the assumed peculiar amplitude statistics of noise, e.g., Middleton distributed [8, Sec. 5.8].
- Development of single carrier modulation (coded FSK, CDMA in first generation PLC devices), UWB modulation, multicarrier modulation (OFDM and filter

bank modulation in second generation PLC devices) [8, Sec. 5.2-5.5].

- Precoding design taking into account the periodically time variant behavior of the medium so that the signal-to-noise ratio exhibits a periodic behavior over mains cycles [9, Sec. 6], [120].
- MIMO techniques (also adopted by standards) and selection combining schemes including hybrid approaches that merge PLC links with radio links to improve diversity [121], [123].

Although most of the above mentioned aspects are common to any communication system, the PLC channel specificities introduce several challenges, e.g., transmission is more band limited than in other systems, up-to-now a PSD constraint rather than a total power constraint has been imposed at the transmitter, spectrum management is relevant for coexistence, prediction of channel and noise is also relevant and may be enabled with new understanding of the medium.

The question is whether ML can bring new *lymph* in PLC PHY layer design. We envision a number of ways forward as detailed below.

Focusing at the receiver and at its tasks, ML can be used for estimation problems such as channel estimation and synchronization. For instance, assuming a certain parametric model of the channel, the parameters can be estimated via learning and inference. Similarly training of the equalizer can also be done via learning from field data. More in general, the receiver task, i.e. data detection, can be seen as a classification problem. It can be solved with supervised learning, namely using NNs (Sec. II-A.2) or SVMs (Sec. II-A.3), where the received signal is the input and the known transmitted bits are the output during the training process.

If we now look at the full communication system, a fascinating idea is to treat it as an autoencoder [47] (Sec. II-B.3). Let $s \in \mathcal{M} = \{1, 2, \dots, M\}$ be a message out of M possible ones and let $F : \mathcal{M} \mapsto \mathbb{R}^n$ be the transform realized by the encoding stage that maps the input message s into the transmitted signal $\mathbf{x} = F(s)$. This first block can be considered as the transmitter block which generates the waveform $\mathbf{x} = F(s)$. After it, the middle block of the autoencoder, learns the transform made by the channel and stochastically described by the conditional probability density function $p(\mathbf{y}|\mathbf{x})$ where \mathbf{y} corresponds to the received signal. Lastly, the decoding block (receiver) takes \mathbf{y} as input and estimates the transmitted message s by computing the transform $G : \mathbb{R}^n \mapsto \mathcal{M}$. Each of these blocks can be implemented using NNs so that their chain constitutes a deep neural network which, trained end-to-end, reconstructs the input message at the output. Since the input is the message s and the desired output is the same reconstructed message s , this leads to a classification task where the cross-entropy is a natural choice for the cost function. The end-to-end learning process enables the autoencoder to find a robust representation of the input signal which can lead to the identification of optimal coding and decoding schemes for stochastic channels.

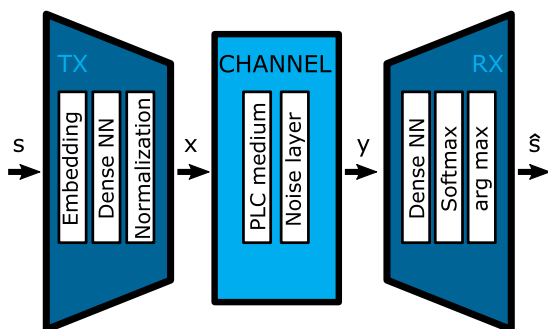


FIGURE 9. Representation of a PLC system as an autoencoder.

The autoencoder may find a solution that outperforms existing modulation and coding methods. In fact, the autoencoder approach works without any assumption on the channel and is able to recognize, in principle, its dynamics without the need to rely on the current view of the PLC channel as a linear periodically time-variant system [87].

An autoencoder for application in PLC, schematically described in Fig. 9, has been implemented using TensorFlow libraries [124]. The transmitter and receiver blocks were modeled as a cascade of multiple fully connected layers. The channel block was designed with a first layer that embeds the channel response (medium) and a second additive layer of Gaussian noise. An arbitrary channel response for the simulation was taken from the dataset described in [97] consisting of several channel realizations with bandwidth of 100 MHz. The autoencoder takes as input symbols with cardinality 4 and 16 (2 or 4 bits), respectively referred to as 4-autoencoder, and 16-autoencoder. The performance of the autoencoder (in terms of symbol error rate) has been compared to that attained by an impulsive 4-PAM or 16-PAM modulation scheme with an ideal matched filter receiver [125]. The results are shown in Fig. 10 as a function of the Signal-to-Noise Ratio (SNR) per bit.

It is interesting to notice that the 4-autoencoder performs 1 dB worse than the 4-PAM scheme in the considered SNR range which is also due to the fact that the 4-PAM scheme assumes perfect knowledge of the channel. However, by increasing the input signal cardinality the autoencoder manifests very good performance exceeding that of 16-PAM for all the considered SNRs. Furthermore, at a given SNR level we note an autoencoder cliff effect, i.e., a significant increase of the SER curve slope. This can be justified if we think that the autoencoder finds a more suitable non-linear coding and decoding scheme when it has a larger hyperspace to search in.

Another bunch of applications of ML for PLC is in the domain of optimization problems that arise at the transmitter side, namely, PHY layer resource allocation. Let us think for instance to the water filling problem that aims at distributing optimally power across the available spectrum [9]. Instead of pursuing a physical model based approach, we may want to train a NN to provide the power allocation as the result of

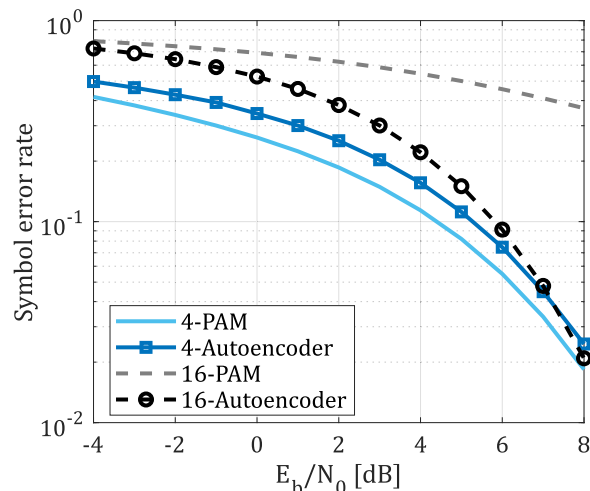


FIGURE 10. SER versus E_b/N_0 for the autoencoder of Fig. 9 and the M-PAM modulation scheme.

observing data from the field and defining a cost function which can be the bit-error rate or throughput. It should be noted that with a bottom-up approach this task can be solved by identifying the system state (channel response and noise), perform classification and estimation, and then dynamically determine the optimal allocation for the specific identified system state. However, a physical agnostic neural network can embed in one shot all these tasks and efficiently provide the solution. Of course, how to train the network and how complex the network has to be is a research question especially in PLC where the system state can have abrupt changes due to network topology changes, load changes, and noise generators changes. In addition, another approach based on reinforcement (Sec. II-C) learning can be exploited as done in [83].

V. ML FOR MAC AND NETWORK LAYER PLC

Media access control and networking can also enjoy the use of ML. State-of-the-art PLC systems use contentious based media access techniques, i.e., CSMA, Aloha, flooding, as well as adaptive TDMA and FDMA [9]. A universally optimal solution has not yet been found especially because of the heterogeneity of applications, services, traffic requirements, variability of medium conditions and network configurations. Let us think for instance of a typical automatic metering application in the access network managed by the concentrator at the MV-LV sub-station, in contrast to a PLC in-building network managed by cell coordinators, or a street light control application, in contrast to a PLC backhauling solution for 5G wireless micro-cells. Although chip sets and transceivers (both NB and BB) have been developed to serve as enabling communication devices for the plethora of above applications, and although they can provide point-to-point connectivity, the optimization of their performance in a massive networked environment is still a challenge. This is because multiple layers need to be optimized and so far limited data was available to learn how to exploit the wide

flexibility of parameters at both the PHY and MAC layers. Things get even more complex when cooperation is required, i.e., when relaying and routing has to be implemented to offer full coverage in large massive networks [126], [127]. In brief, cross layer optimization, resource allocation, spectrum management and scheduling etc. are challenging tasks in PLC networks, as the results coming from field trials and large deployments are telling us. The dynamics of the channel and traffic, the abrupt network shut-downs and the need for fast reconfiguration, render these tasks extremely demanding if a layer by layer approach is considered, as traditionally done. This is because the traditional approach requires modeling each layer and function, which is prohibitively complex. Here is where ML comes into place.

ML and more in general artificial intelligence is envisioned as a relevant tool to develop a more abstract and synthetic model of the network behavior and to realize the network orchestration tasks [128]. The network behavior has to be learned via the acquisition of real data sets where events, actions and results are labeled to create time series that can train optimization algorithms. In essence the orchestration of the network can be seen as a control problem where data observers serve as sensors, a global cost function is then formulated and minimized via learning from field data. To the best of our knowledge, efforts in applying ML techniques in routing problems have been carried out mostly for wireless networks and specifically Mobile Ad-hoc Networks (MANETs), where topology information is not available to nodes and the data-driven nature of the application poses significant challenges in terms of adaptation to run-time changes and scalability [129]–[132]. Most studies also show that classical solutions, such as Bellman-Ford, Ad-hoc On-demand Distance Vector (AODV), Destination-Sequenced Distance-Vector (DSDV), Low-Energy Adaptive Clustering Hierarchy (LEACH), shortest path and the such, perform poorer than ML approaches when it comes to achievable throughput, end-to-end delay and time required to converge to a better solution when run-time changes occur. The most widespread approach to tackle these problems is reinforcement learning [133], where cost functions are defined in order to switch the behavior of a router based on environmental and link conditions (Sec. II-C). This is done by implementing a rewarding (or penalizing) system, which influences the probability of an element to make a specific choice, that keeps track of past states of the network and uses them to predict the possible future changes in the traffic load. The main disadvantage of RL is its computational demand, which weighs on every node of the network when implemented in a distributed kind of architecture. In the following, we present a different, computationally lighter approach, where supervised learning is used to synthetically train neural networks to implement a decentralized smart routing solution that is able to react to changes in PLC networks by using readily available information.

In more detail, the envisioned application aims at using the power line distribution network as a fronthaul infrastructure for a small cell radio network. In [134] the small cell radio

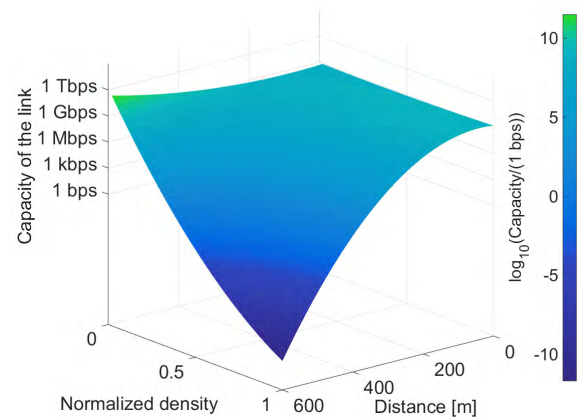


FIGURE 11. Inferred average capacity of an end-to-end PLC link based on easily-observable topology parameters.

network and the underlying power line infrastructure are brought together in a joint paradigm and the capacity of the PLC fronthaul is analyzed through a bottom-up emulation tool [114], [135]. A regression approach is developed to tackle the problem of determining the capacity of an end-to-end power line link based solely on geometrical/topological properties of the network, for instance the density of nodes (radio cells) in the overall service area, and the distance between the communicating nodes.

With the approach followed, it is possible to infer dependencies of the channel capacity and routing performance on a number of parameters (Fig. 11), e.g., topology type, density of nodes (branches), cable type, loads, electrical distance, etc. The channel model used to generate synthetic data is a TL model where the topology, loads, cable electrical characteristics etc. are included. It generates frequency selective channel responses for any pair of nodes in the network. Distance here refers to the length of the power line backbone that connects two considered nodes. In detail, Fig. 11 shows the capacity inferred through regression of two nodes pertaining to a PLC network when the communication is implemented in the broad-band spectrum (2-86 MHz). The curve shows an aberration for low densities and high distances: capacity does not actually increase with distance for low densities. This is due to the fact that regression is implemented through a simulator that first deploys loads (small cells) on the territory and then tries to connect them in a way that power line is minimized. This means that for low densities, high distances are never achieved, thus the regression algorithm does not have data for that specific density-distance subset.

The capacity inferred through regression enables a ML approach that can be used by nodes of the network to infer the best position of a global router to extend coverage and maximize capacity, given that the topology is known at every point (Fig. 12). This information can be set by the manager of the power line infrastructure, otherwise, whenever a new node is added to the network, the topology can be reconstructed by a self-discovery mechanism. With every node being able

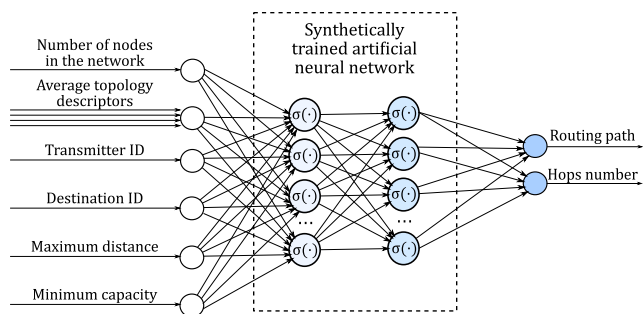


FIGURE 12. NN used for determining routing.

to infer quickly and easily the capacity of the links in the network, a simple ML-based routing algorithm can easily increase capacity, as shown in Fig. 13, where the capacity gained with ML routing is compared to the optimal solution, the latter achieved with full knowledge of topology and an exhaustive solution of the optimization problem.

On the other hand, we can also implement routing by considering the number of hops as the metric to be minimized, while imposing a constraint on the minimum capacity to be achieved. In addition, PLC can be used as an access network infrastructure for narrowband applications [136]. Sensor networks, which usually communicate via wireless technologies, can be supported by a PLC infrastructure where this solution has better penetration than a wireless one. A popular protocol for routing in these networks is the Lightweight On-demand Ad-hoc Distance-vector (LOADng) protocol developed by G3-PLC [136]. This protocol relies on information tables in order to connect all nodes in the network; topology is discovered by broadcasting route request packets, and routes are determined by minimizing a cost function. Asymmetric links, burst traffic and noise, as well as nodes with limited resources on power, computation and memory offer constraints that can be solved with lightweight neural networks. As shown in Fig. 12, we can train our learning machine to devise the optimal number of routers and the best path for a message to reach any point in the network, both in downlink and uplink. The training set is herein generated synthetically by means of the aforementioned emulation tool, thus a big number of random network samples can be used to let the NN learn the underlying relations between topology and the optimal routing strategy. Specifically, the NN scheme accepts as inputs the source-destination IDs, the number of nodes in the network, the maximum distance between these nodes, the minimum capacity required to consider a link as functioning, and some other geometrical descriptors. The scheme outputs the number of routers required to achieve connectivity and the best path for connection. The basic idea of this implementation is that the nodes do not need to know the topology of the network, but only the few parameters referenced in Fig. 12. Fig. 14 reports the percentage of correct guessed routing paths, i.e., the ratio of routing paths obtained by ML that are equal to the ones obtained with the exhaustive solution of the optimization problem and the perfect knowledge of the

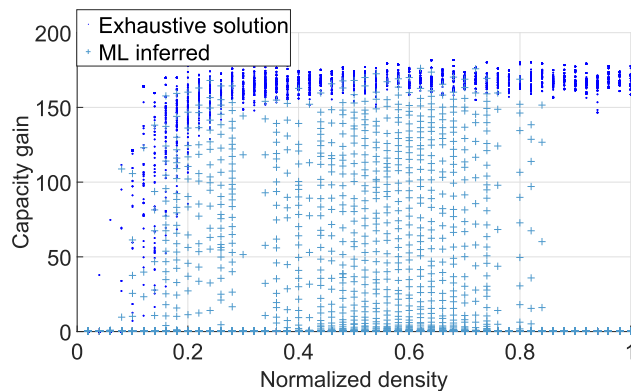


FIGURE 13. Capacity gain using ML routing in a PLC backhauling cellular network as a function of the radio cells density in the area.

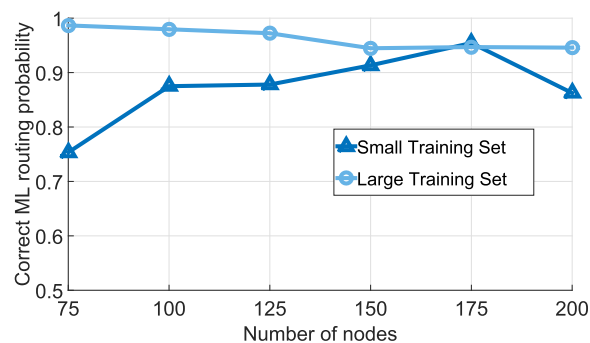


FIGURE 14. Probability of determining the optimal routing scheme via NN as a function of the number of nodes and training set dimension in a metering PLC network.

link capacity. The results show a high probability of success. Furthermore, it is worth noticing that the NN was trained with network topologies with a number of nodes ranging from 100 to 175. To prevent overfitting, an initial dropout layer [137] (drop rate $p = 0.5$) was also added. The set used for testing the performance of the NN is nevertheless larger, i.e., networks with less than 100 or more than 175 nodes are simulated in order to assess the capabilities of our approach to find routing solutions for unobserved situations during the training stage. This explains the somewhat lower performance of ML routing w.r.t. the optimal routing for a number of nodes smaller than 100 and larger than 175.

In conclusion, based on the results presented, ML for networking in PLC is promising. Further investigations should be done: representative datasets have to be generated, supervised and unsupervised ML techniques have to be compared, and appropriate training algorithms have to be developed.

VI. OTHER APPLICATIONS

PLC is not a mere communication technology. The power grid is a fully interconnected system where all electrical phenomena affect the entire network. Moreover, it is influenced by external physical events. Hence, PLC nodes can be used to sense the grid itself as well as the surrounding environment. In other words, PLC transceivers can act as probes for grid diagnostics by analyzing the electromagnetic

field and the data traffic in the PLC frequency bands [11]. Possible applications of PLC for grid sensing are topology reconstruction [138], [139] and anomalies detection, i.e., fault detection, cable aging, load identification [140]–[142]. The former application enables grid operators to better the knowledge about the grid configuration, the status of switches and feeders which is not always complete especially in the LV part of the access network. The knowledge of the network topology is also important for networking aspects since it allows to locate nodes, and implement for instance geo-routing algorithms. The latter application enables grid operators to better monitor the grid status, malfunctions etc. which in turn enables predictive maintenance of the grid and therefore grants a better energy delivery service. Some recent results have provided a better understanding of the propagation of high frequency signals (as PLC ones) in multi-conductor transmission lines [141]. Essentially, it has been shown that we can relate the state of the grid, cables and loads to several electrical quantities that can be measured and that provide information to detect the presence of anomalies. The results offer an analytic framework described by a set of complex but extremely informative functions. Nonetheless, diagnosis of malfunctions ends up into an estimation problem or classification problem [140]. In this context, [142] has already provided results on the usage of ML for cable degradation detection, for instance.

To provide a numerical result, we consider the problem of detecting anomalies in a distribution grid by the analysis of high frequency PLC signals. In [141] it was shown that for this objective, the line admittance at the nodes (\mathbf{Y}_{in}), the reflection coefficient (ρ_{in}), and the transfer function (\mathbf{H}) between pairs of nodes can be exploited. Supervised learning (Sec. II-A) can be very powerful here; once several sampled signals \mathbf{x} are stored with the respective labels y , NNs or SVMs can learn the deterministic relationship $y = F(\mathbf{x})$ (which can be arbitrarily complex) and predict new unknown labels for different sensed signals. Here, we consider a simple neural network approach. A dataset was realized based on a single topology realization with 20 nodes and average node length of 700 m. Anomalies occurring on the grid have been added to obtain 10000 realizations of a perturbed grid, specifically, load impedances changes, concentrated faults and distributed faults according to the models described in [141].

The training process was conducted using 50% of the data, while testing was done using the other unbiased half part. We used only one hidden layer with 100 neurons and a softmax layer to work with probabilities as output. Input signals are the ratios between the measurement of the electrical parameters \mathbf{Y}_{in} , ρ_{in} and \mathbf{H} at a given time, and the reference value of the respective signal in the band 4.3 – 500 kHz with 4.3 kHz sampling frequency. The reference values were obtained when there was no anomaly. The output signals from the classifier are 4 different class indicators according to the particular type of fault: 1. unperturbed, 2. load impedance change, 3. concentrated fault, 4. distributed fault. The obtained results are similar for all three types of electrical

Predicted class	1	1217 24.3%	64 1.3%	14 0.3%	30 0.6%	91.8% 8.2%
	2	0 0.0%	1123 22.5%	10 0.2%	275 5.5%	79.8% 20.2%
	3	0 0.0%	53 1.1%	1229 24.6%	58 1.2%	91.7% 8.3%
	4	0 0.0%	38 0.8%	0 0.0%	889 17.8%	95.9% 4.1%
		100% 0.0%	87.9% 12.1%	98.1% 1.9%	71.0% 29.0%	89.2% 10.8%
		1	2	3	4	
		Real class				

FIGURE 15. Confusion matrix (accuracy of the prediction) for 4-classes anomaly detection using the reflection coefficient as the main input signal.

signals considered, so we present here only the accuracy values of the classifier when using the reflection coefficient. Furthermore, both the case of load impedances that stay constant (at 2 kΩ) or that are randomly variable are considered.

TABLE 2. Anomaly detection accuracy with constant and variable load impedances.

Accuracy		Classes		
		Fault Detection	All 4 Classes	3 Classes
Load Imped.	Constant	87.8%	87.8%	100 %
	Variable	98.7%	89.2%	87.1%

The first column of Tab. 2 reports the ability to detect the presence of a fault and this is achieved through an accuracy of 87.8% when the load impedances are constant, and 98.7% when they change, respectively. The second column shows the ability to detect and classify the type of anomaly: the average accuracy is 87.8% when the load impedances are constant, and 89.2% when they change, respectively. It is rather interesting to notice that, when the load impedances are constant, the neural network is not able to fully distinguish between the unperturbed network (class 1) and the distributed fault event (class 4). This suggests to remove from the data the last type of anomaly, and study the ability of the neural network to classify the remaining first 3 classes. This is presented in the third column, where it is shown that the accuracy reaches 100% when the load impedances are constant, and 87.1% when they change, which means that most of the uncertainty is introduced by the last class. More details are offered by Fig. 15, which reports the confusion matrix of 4 classes anomaly detection with variable loads. In this case,

the confusion matrix shows a conflict between class 2 and class 4, in particular the NN erroneously tends to predict (around 1/3 of the times) class 2 when the correct class is instead the 4th. The results are promising and encourage further work for tuning the NN hyper-parameters.

VII. CONCLUSIONS

This paper has provided an overview of ML as a signal/data processing tool to improve the knowledge and the performance of PLC systems. A clear distinction among deterministic and probabilistic ML formulations has been made. Several new ideas of application of ML in PLC have been presented in the domain of medium characterization, statistical modeling, PHY layer, MAC layer and grid diagnostics. The paper has discussed what ML supervised and unsupervised tools can be used for the problem at hand. The concepts have been supported by numerical results using synthetic or real datasets. They have provided encouraging evidence that ML has a role in PLC. Despite its infancy, in this application domain, ML stimulates an all new era of research activities in PLC. Several future research directions have been envisioned, among which the exploitation of domain knowledge of PLC to better extract features via ML tools. For instance, the a priori physical knowledge of the channel (through parametric models) offers a guidance to better learn and characterize it. Furthermore, how to train, validate and test the machine (which is a core topic in ML), and how to compare the ML tools opens the door to several research endeavors. The starting point in PLC will be the development of representative data sets. This has to be followed by a statistical learning approach where the validity of results is measured in a probability framework.

REFERENCES

- [1] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, Jul./Oct. 1948. [Online]. Available: <https://ieeexplore.ieee.org/document/6773024>
- [2] I. Kant, *Critique of Pure Reason* (The Cambridge Edition of the Works of Immanuel Kant). New York, NY, USA: Cambridge Univ. Press, 1998.
- [3] C. M. Bishop, *Pattern Recognition and Machine Learning* (Information Science and Statistics). Berlin, Germany: Springer-Verlag, 2006.
- [4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.
- [5] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: MIT Press, 2012.
- [6] L. Lampe, A. M. Tonello, and T. G. Swart, Eds., *Power Line Communications: Principles, Standards and Applications from Multimedia to Smart Grid*. Hoboken, NJ, USA: Wiley, 2016.
- [7] F. J. Cañete, K. Dostert, S. Galli, M. Katayama, L. Lampe, M. Lienard, S. Mashayekhi, D. G. Michelson, M. Nassar, R. Pighi, A. Pinomaa, M. Raugi, A. M. Tonello, M. Tucci, and F. Versolatto, *Channel Characterization*. Hoboken, NJ, USA: Wiley, 2016, ch. 2, pp. 8–177. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118676684.ch2>
- [8] K. Dostert, M. Girotto, L. Lampe, R. Raheli, D. Rieken, T. G. Swart, A. M. Tonello, A. J. H. Vinck, and S. Weiss, *Digital Transmission Techniques*. Hoboken, NJ, USA: Wiley, 2016, ch. 5, pp. 261–385. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118676684.ch5>
- [9] J. A. Cortés, S. D' Alessandro, L. P. Do, L. Lampe, R. Lehnert, M. Noori, and A. M. Tonello, *Medium Access Control and Layers Above in PLC*. Hoboken, NJ, USA: Wiley, 2016, ch. 6, pp. 386–448. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118676684.ch6>
- [10] A. N. Milioudis, G. T. Andreou, and D. P. Labridis, "Enhanced protection scheme for smart grids using power line communications techniques—Part I: Detection of high impedance fault occurrence," *IEEE Trans. Smart Grid*, vol. 3, no. 4, pp. 1621–1630, Dec. 2012.
- [11] A. M. Tonello, "Why you should consider the uptake of PLC tech," *ESI Africa-Utility Maintenance Focus*, vol. 3, pp. 34–36, Jul. 2017.
- [12] T. M. Mitchell, *Machine Learning*, 1st ed. New York, NY, USA: McGraw-Hill, 1997.
- [13] O. Simeone, "A very brief introduction to machine learning with applications to communication systems," *IEEE Trans. Cogn. Commun. Netw.*, vol. 4, no. 4, pp. 648–664, Dec. 2018. doi: [10.1109/TCCN.2018.2881442](https://doi.org/10.1109/TCCN.2018.2881442).
- [14] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Netw.*, vol. 2, no. 5, pp. 359–366, 1989.
- [15] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Parallel distributed processing: Explorations in the microstructure of cognition," in *Learning Internal Representations by Error Propagation*, vol. 1, D. E. Rumelhart and J. L. McClelland, Eds. Cambridge, MA, USA: MIT Press, 1986, pp. 318–362. [Online]. Available: <http://dl.acm.org/citation.cfm?id=104279.104293>
- [16] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, Dec. 2014. [Online]. Available: <https://arxiv.org/abs/1412.6980>
- [17] Y.-T. Ma, K.-H. Liu, and Y.-N. Guo, "Artificial neural network modeling approach to power-line communication multi-path channel," in *Proc. Int. Conf. Neural Netw. Signal Process.*, Jun. 2008, pp. 229–232.
- [18] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proc. 5th Annu. Workshop Comput. Learn. Theory (COLT)*, New York, NY, USA, 1992, pp. 144–152. [Online]. Available: <http://doi.acm.org/10.1145/130385.130401>
- [19] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995. [Online]. Available: <https://doi.org/10.1023/A:1022627411411>
- [20] Y. Huo, G. Prasad, L. Atanackovic, L. Lampe, and V. C. M. Leung, "Grid surveillance and diagnostics using power line communications," in *Proc. IEEE Int. Symp. Power Line Commun. Appl. (ISPLC)*, Apr. 2018, pp. 1–6.
- [21] T. K. Ho, "Random decision forests," in *Proc. 3rd Int. Conf. Document Anal. Recognit. (ICDAR)*, vol. 1, Washington, DC, USA, Aug. 1995, pp. 278–282. [Online]. Available: <http://dl.acm.org/citation.cfm?id=844379.844681>
- [22] A. Zappone, M. Di Renzo, M. Debbah, T. T. Lam, and X. Qian, "Model-aided wireless artificial intelligence: Embedding expert knowledge in deep neural networks towards wireless systems optimization," *CoRR*, Aug. 2018. [Online]. Available: <https://arxiv.org/abs/1808.01672>
- [23] T. Gruber, S. Cammerer, J. Hoydis, and S. T. Brink, "On deep learning-based channel decoding," in *Proc. 51st Annu. Conf. Inf. Sci. Syst. (CISS)*, Baltimore, MD, USA, Mar. 2017, pp. 1–6. doi: [10.1109/CISS.2017.7926071](https://doi.org/10.1109/CISS.2017.7926071).
- [24] H. Ye, G. Y. Li, and B.-H. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 1, pp. 114–117, Feb. 2018.
- [25] E. Nachmani, E. Marciano, L. Lugosch, W. J. Gross, D. Burshtein, and Y. Be'ery, "Deep learning methods for improved decoding of linear codes," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 119–131, Feb. 2018.
- [26] A. Argyris, J. Bueno, and I. Fischer, "Photonic machine learning implementation for signal recovery in optical communications," *Sci. Rep.*, vol. 8, no. 1, May 2018, Art. no. 8487.
- [27] N. E. West and T. O'Shea, "Deep architectures for modulation recognition," in *Proc. IEEE Int. Symp. Dyn. Spectr. Access Netw. (DySPAN)*, Mar. 2017, pp. 1–6.
- [28] S. H. Fang and T. N. Lin, "Indoor location system based on discriminant-adaptive neural network in IEEE 802.11 environments," *IEEE Trans. Neural Netw.*, vol. 19, no. 11, pp. 1973–1978, Nov. 2008.
- [29] N. Strodthoff, B. Göktepe, T. Schierl, C. Hellge, and W. Samek, "Enhanced machine learning techniques for early HARQ feedback prediction in 5G," *CoRR*, Jul. 2018. [Online]. Available: <https://arxiv.org/abs/1807.10495>
- [30] V. K. Tumuluru, P. Wang, and D. Niyato, "A novel spectrum-scheduling scheme for multichannel cognitive radio network and performance analysis," *IEEE Trans. Veh. Technol.*, vol. 60, no. 4, pp. 1849–1858, May 2011.
- [31] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, "Learning to optimize: Training deep neural networks for interference management," *IEEE Trans. Signal Process.*, vol. 66, no. 20, pp. 5438–5453, Oct. 2018.

- [32] D. Del Testa, M. Danieleto, G. M. Di Nunzio, and M. Zorzi, "Estimating the number of receiving nodes in 802.11 networks via machine learning techniques," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1–7.
- [33] M. Chen, W. Saad, C. Yin, and M. Debbah, "Echo state networks for proactive caching in cloud-based radio access networks with mobile users," *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 3520–3535, Jun. 2017.
- [34] Y. Wang, M. Martonosi, and L.-S. Peh, "A supervised learning approach for routing optimizations in wireless sensor networks," in *Proc. 2nd Int. Workshop Multi-Hop Ad Hoc Netw., Theory Reality (REALMAN)*, New York, NY, USA, 2006, pp. 79–86. doi:10.1145/1132983.1132997.
- [35] T. T. Nguyen and G. Armitage, "A survey of techniques for Internet traffic classification using machine learning," *IEEE Commun. Surveys Tuts.*, vol. 10, no. 4, pp. 56–76, 4th Quart., 2008.
- [36] O. Simeone, "A brief introduction to machine learning for engineers," *Found. Trends Signal Process.*, vol. 12, nos. 3–4, pp. 200–431, 2018.
- [37] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient K-means clustering algorithm: Analysis and implementation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 881–892, Jul. 2002.
- [38] M. Roux, "A comparative study of divisive and agglomerative hierarchical clustering algorithms," *J. Classification*, vol. 35, no. 2, pp. 345–366, Jul. 2018. doi:10.1007/s00357-018-9259-9.
- [39] K. Bade and A. Nurnberger, "Personalized hierarchical clustering," in *Proc. IEEE/WIC/ACM Int. Conf. Web Intell. (WI)*, Dec. 2006, pp. 181–187.
- [40] J. Leskovec, A. Rajaraman, and J. D. Ullman, *Mining of Massive Datasets*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2014.
- [41] J. Vesanto and E. Alhoniemi, "Clustering of the self-organizing map," *IEEE Trans. Neural Netw.*, vol. 11, no. 3, pp. 586–600, May 2000.
- [42] L. Dinh, D. Krueger, and Y. Bengio, "NICE: Non-linear independent components estimation," *CoRR*, Oct. 2014. [Online]. Available: <https://arxiv.org/abs/1410.8516>
- [43] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real NVP," *CoRR*, May 2016. [Online]. Available: <https://arxiv.org/abs/1605.08803>
- [44] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. 25th Int. Conf. Mach. Learn. (ICML)*, New York, NY, USA, 2008, pp. 1096–1103. doi:10.1145/1390156.1390294.
- [45] G. Alain and Y. Bengio, "What regularized auto-encoders learn from the data generating distribution," *CoRR*, Nov. 2012. [Online]. Available: <https://arxiv.org/abs/1211.4246>
- [46] S. Rifai, P. Vincent, X. Müller, X. Glorot, and Y. Bengio, "Contractive auto-encoders: Explicit invariance during feature extraction," in *Proc. 28th Int. Conf. Mach. Learn. (ICML)*, 2011, pp. 833–840.
- [47] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, Dec. 2017.
- [48] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," *CoRR*, Dec. 2013. [Online]. Available: <https://arxiv.org/abs/1312.6114>
- [49] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, vol. 2. Cambridge, MA, USA: MIT Press, 2014, pp. 2672–2680. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2969033.2969125>
- [50] T. J. O'Shea, T. Roy, N. West, and B. C. Hilburn, "Physical layer communications system design over-the-air using adversarial networks," in *Proc. 26th Eur. Signal Process. Conf. (EUSIPCO)*, Sep. 2018, pp. 529–532.
- [51] B. Karanov, M. Chagnon, F. Thouin, T. A. Eriksson, H. Bülow, D. Lavery, P. Bayvel, and L. Schmalen, "End-to-end deep learning of optical fiber communications," *J. Lightw. Technol.*, vol. 36, no. 20, pp. 4843–4855, Oct. 15, 2018.
- [52] H. Kim, Y. Jiang, S. Kannan, S. Oh, and P. Viswanath, "Deepcode: Feedback codes via deep learning," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, Montreal, QC, Canada, Dec. 2018, pp. 9436–9446.
- [53] M. Kim, N.-I. Kim, W. Lee, and D.-H. Cho, "Deep learning-aided SCMA," *IEEE Commun. Lett.*, vol. 22, no. 4, pp. 720–723, Apr. 2018.
- [54] C. Studer, S. Medjkouh, E. Gönültaş, T. Goldstein, and O. Tirkkonen, "Channel charting: Locating users within the radio environment using channel state information," *IEEE Access*, vol. 6, pp. 47682–47698, 2018.
- [55] S. Dörner, S. Cammerer, J. Hoydis, and S. ten Brink, "Deep learning based communication over the air," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 132–143, Feb. 2018.
- [56] V. Raj and S. Kalyani, "Backpropagating through the air: Deep learning at physical layer without channel models," *IEEE Commun. Lett.*, vol. 22, no. 11, pp. 2278–2281, Nov. 2018.
- [57] F. A. Aoudia and J. Hoydis, "End-to-end learning of communications systems without a channel model," in *Proc. 52nd Asilomar Conf. Signals, Syst., Comput.*, Oct. 2018, pp. 298–303.
- [58] H. Ye, G. Y. Li, B.-H. F. Juang, and K. Sivanesan, "Channel agnostic end-to-end learning based communication systems with conditional GAN," *CoRR*, Jul. 2018. [Online]. Available: <https://arxiv.org/abs/1807.00447>
- [59] K. Davaslioglu and Y. E. Sagduyu, "Generative adversarial learning for spectrum sensing," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.
- [60] U. Challita, L. Dong, and W. Saad, "Proactive resource management for LTE in unlicensed spectrum: A deep learning perspective," *IEEE Trans. Wireless Commun.*, vol. 17, no. 7, pp. 4674–4689, Jul. 2018.
- [61] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Comput. Commun.*, vol. 30, nos. 14–15, pp. 2826–2841, Oct. 2007.
- [62] E. Abbe, A. S. Bandeira, and G. Hall, "Exact recovery in the stochastic block model," *CoRR*, May 2014. [Online]. Available: <https://arxiv.org/abs/1405.3267>
- [63] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
- [64] Q. Li, W. Cai, X. Wang, Y. Zhou, D. D. Feng, and M. Chen, "Medical image classification with convolutional neural network," in *Proc. 13th Int. Conf. Control Automat. Robot. Vis. (ICARCV)*, Dec. 2014, pp. 844–848.
- [65] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proc. 25th Int. Conf. Mach. Learn. (ICML)*, New York, NY, USA, 2008, pp. 160–167. doi:10.1145/1390156.1390177.
- [66] A. Graves, A.-R. Mohamed, and G. E. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Vancouver, BC, Canada, May 2013, pp. 6645–6649. doi:10.1109/ICASSP.2013.6638947.
- [67] A. Graves and J. Schmidhuber, "Offline handwriting recognition with multidimensional recurrent neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds. Red Hook, NY, USA: Curran Associates, 2009, pp. 545–552.
- [68] E. Choi, M. T. Bahadori, A. Schuetz, W. F. Stewart, and J. Sun, "Doctor AI: Predicting clinical events via recurrent neural networks," in *Proc. 1st Mach. Learn. Healthcare Conf.*, vol. 56, F. Doshi-Velez, J. Fackler, D. Kale, B. Wallace, and J. Wiens, Eds. Los Angeles, CA, USA: PMLR, Aug. 2016, pp. 301–318.
- [69] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, May 2015, pp. 1–15.
- [70] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. doi:10.1162/neco.1997.9.8.1735.
- [71] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.
- [72] Z. C. Lipton, J. Berkowitz, and C. Elkan, "A critical review of recurrent neural networks for sequence learning," *CoRR*, May 2015. [Online]. Available: <http://arxiv.org/abs/1506.00019>
- [73] H. Salehinejad, S. Sankar, J. Barfett, E. Colak, and S. Valaee, "Recent advances in recurrent neural networks," *CoRR*, Feb. 2018. [Online]. Available: <https://arxiv.org/abs/1801.01078>
- [74] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proc. 32nd Int. Conf. Mach. Learn.*, vol. 37, F. Bach and D. Blei, Eds. Lille, France: PMLR, Jul. 2015, pp. 1889–1897. [Online]. Available: <http://proceedings.mlr.press/v37/schulman15.html>
- [75] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. A. Riedmiller, "Playing Atari with deep reinforcement learning," *CoRR*, Dec. 2013. [Online]. Available: <https://arxiv.org/abs/1312.5602>

- [76] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis, "Mastering the game of go without human knowledge," *Nature*, vol. 550, pp. 354–359, Oct. 2017.
- [77] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, 1st ed. Cambridge, MA, USA: MIT Press, 1998.
- [78] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Process. Mag.*, vol. 34, no. 6, pp. 26–38, Nov. 2017.
- [79] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *Engineering Applications of Neural Networks*, C. Jayne and L. Iliadis, Eds. Cham, Switzerland: Springer, 2016, pp. 213–226.
- [80] N. Bitar, S. Muhammad, and H. H. Refai, "Wireless technology identification using deep convolutional neural networks," in *Proc. IEEE 28th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Oct. 2017, pp. 1–6.
- [81] G. Kechriotis, E. Zervas, and E. S. Manolakos, "Using recurrent neural networks for adaptive communication channel equalization," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 267–278, Mar. 1994.
- [82] B. J. Radford, L. M. Apolonio, A. J. Trias, and J. A. Simpson, "Network traffic anomaly detection using recurrent neural networks," *CoRR*, Mar. 2018. [Online]. Available: <https://arxiv.org/abs/1803.10769>
- [83] B. Nikfar, G. Bumiller, and A. J. H. Vinck, "An adaptive pursuit strategy for dynamic spectrum assignment in narrowband PLC," in *Proc. IEEE Int. Symp. Power Line Commun. Appl. (ISPLC)*, Apr. 2017, pp. 1–6.
- [84] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, and D. I. Kim, "Applications of deep reinforcement learning in communications and networking: A survey," *CoRR*, Oct. 2018. [Online]. Available: <https://arxiv.org/abs/1810.07862>
- [85] M. Tlich, A. Zeddham, F. Moulin, and F. Gauthier, "Indoor power-line communications channel characterization up to 100 MHz—Part I: One-parameter deterministic model," *IEEE Trans. Power Del.*, vol. 23, no. 3, pp. 1392–1401, Jul. 2008.
- [86] S. Galli, "A simple two-tap statistical model for the power line channel," in *Proc. ISPLC*, Mar. 2010, pp. 242–248.
- [87] J. A. Cortés, F. J. Cañete, L. Díez, and J. L. G. Moreno, "On the statistical properties of indoor power line channels: Measurements and models," in *Proc. IEEE Int. Symp. Power Line Commun. Appl.*, Apr. 2011, pp. 271–276.
- [88] F. Versolatto and A. M. Tonello, "PLC channel characterization up to 300 MHz: Frequency response and line impedance," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2012, pp. 3525–3530.
- [89] M. Babic, "D5: Pathloss as a function of frequency, distance and network topology for various LV and MV European powerline networks," Deliverable D5, EU Project Opera, Mar. 2005.
- [90] A. M. Tonello, F. Versolatto, and C. Tornelli, "Analysis of impulsive uwb modulation on a real MV test network," in *Proc. IEEE Int. Symp. Power Line Commun. Appl.*, Apr. 2011, pp. 18–23.
- [91] M. Mohammadi, L. Lampe, M. Lok, S. Mirabbasi, M. Mirvakili, R. Rosales, and P. van Veen, "Measurement study and transmission for in-vehicle power line communication," in *Proc. IEEE Int. Symp. Power Line Commun. Appl.*, Mar./Apr. 2009, pp. 73–78.
- [92] M. Antoniali, M. De Piantè, and A. M. Tonello, "PLC noise and channel characterization in a compact electrical car," in *Proc. IEEE 17th Int. Symp. Power Line Commun. Appl.*, Mar. 2013, pp. 29–34.
- [93] M. Antoniali, A. M. Tonello, M. Lenardon, and A. Qualizza, "Measurements and analysis of PLC channels in a cruise ship," in *Proc. IEEE Int. Symp. Power Line Commun. Appl.*, Apr. 2011, pp. 102–107.
- [94] P. Pagani, R. Hashmat, A. Schwager, D. Schneider, and W. Bäschlin, "European MIMO PLC field measurements: Noise analysis," in *Proc. IEEE Int. Symp. Power Line Commun. Appl.*, Mar. 2012, pp. 310–315.
- [95] A. Pittolo, M. De Piantè, F. Versolatto, and A. M. Tonello, "In-vehicle power line communication: Differences and similarities among the in-car and the in-ship scenarios," *IEEE Veh. Technol. Mag.*, vol. 11, no. 2, pp. 43–51, Jun. 2016.
- [96] S. Galli, "A simplified model for the indoor power line channel," in *Proc. IEEE Int. Symp. Power Line Commun. Appl.*, Dresden, Germany, Mar./Apr. 2009, pp. 13–19.
- [97] A. M. Tonello, F. Versolatto, and A. Pittolo, "In-home power line communication channel: Statistical characterization," *IEEE Trans. Commun.*, vol. 62, no. 6, pp. 2096–2106, Jun. 2014.
- [98] F. J. C. Corripio, J. A. C. Arrabal, L. D. del Rio, and J. T. E. Munoz, "Analysis of the cyclic short-term variation of indoor power line channels," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 7, pp. 1327–1338, Jul. 2006.
- [99] L. T. Berger, A. Schwager, P. Pagani, and D. M. Schneider, "MIMO power line communications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 106–124, 1st Quart., 2015.
- [100] D. Righini and A. M. Tonello, "Noise determinism in multi-conductor narrow band PLC channels," in *Proc. IEEE Int. Symp. Power Line Commun. Appl. (ISPLC)*, Apr. 2018, pp. 1–6.
- [101] M. Zimmermann and K. Dostert, "Analysis and modeling of impulsive noise in broad-band powerline communications," *IEEE Trans. Electromagn. Compat.*, vol. 44, no. 1, pp. 249–258, Feb. 2002.
- [102] N. Shlezinger and R. Dabora, "On the capacity of narrowband PLC channels," *IEEE Trans. Commun.*, vol. 63, no. 4, pp. 1191–1201, Apr. 2015.
- [103] A. Pittolo, A. M. Tonello, and F. Versolatto, "Performance of MIMO PLC in measured channels affected by correlated noise," in *Proc. 18th IEEE Int. Symp. Power Line Commun. Appl.*, Glasgow, U.K., Mar./Apr. 2014, pp. 261–265.
- [104] M. Antoniali, F. Versolatto, and A. M. Tonello, "An experimental characterization of the PLC noise at the source," *IEEE Trans. Power Del.*, vol. 31, no. 3, pp. 1068–1075, Jun. 2016.
- [105] M. Elgenedy, M. Sayed, N. Al-Dhahir, and R. C. Chabaan, "Cyclostationary noise mitigation for SIMO powerline communications," *IEEE Access*, vol. 6, pp. 5460–5484, 2018.
- [106] G. J. Székely, M. L. Rizzo, and N. K. Bakirov, "Measuring and testing dependence by correlation of distances," *Ann. Statist.*, vol. 35, no. 6, pp. 2769–2794, Dec. 2007. doi: [10.1214/009053607000000505](https://doi.org/10.1214/009053607000000505).
- [107] J. S. Richman and J. R. Moorman, "Physiological time-series analysis using approximate entropy and sample entropy," *Amer. J. Physiol.-Heart Circulatory Physiol.*, vol. 278, no. 6, pp. H2039–H2049, 2000.
- [108] D. A. Clifton, *Machine Learning for Healthcare Technologies*, 1st ed. Edison, NJ, USA: IET, 2016.
- [109] D. Righini and A. M. Tonello, "Automatic clustering of noise in multi-conductor narrow band PLC channels," in *Proc. IEEE Int. Symp. Power Line Commun. Appl. (ISPLC)*, Apr. 2019, pp. 1–6.
- [110] B. Han, V. Stoica, C. Kaiser, N. Otterbach, and K. Dostert, "Noise characterization and emulation for low-voltage power line channels between 150 KHz and 10 MHz," KIT, Karlsruhe, Germany, Tech. Rep. KITopen-ID: 1000056745, 2016.
- [111] D. Liang, H. Guo, and T. Zheng, "Impedance tracking and estimation using power line communications," in *Proc. IEEE Int. Symp. Power Line Commun. Appl. (ISPLC)*, Apr. 2019, pp. 1–6.
- [112] M. De Piantè and A. M. Tonello, "Channel and admittance dependence analysis in PLC with implications to power loading/savings," in *Proc. IEEE Int. Symp. Power Line Commun. Appl. (ISPLC)*, Apr. 2019, pp. 1–6.
- [113] F. Passerini and A. M. Tonello, "Physical layer key generation for secure power line communications," *CoRR*, Sep. 2018. [Online]. Available: <https://arxiv.org/abs/1809.09439>
- [114] 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.
- [115] A. M. Tonello and F. Versolatto, "Bottom-up statistical PLC channel modeling—Part II: Inferring the statistics," *IEEE Trans. Power Del.*, vol. 25, no. 4, pp. 2356–2363, Oct. 2010.
- [116] M. Zimmermann and K. Dostert, "A multipath model for the powerline channel," *IEEE Trans. Commun.*, vol. 50, no. 4, pp. 553–559, Apr. 2002.
- [117] 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.
- [118] H. Hirsch, M. Koch, N. Weling, and A. Zeddham, *Electromagnetic Compatibility*. Hoboken, NJ, USA: Wiley, 2016, ch. 3, pp. 178–222. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118676684.ch3>
- [119] D. Righini, F. Passerini, and A. M. Tonello, "Modeling transmission and radiation effects when exploiting power line networks for communication," *IEEE Trans. Electromagn. Compat.*, vol. 60, no. 1, pp. 59–67, Feb. 2018.
- [120] S. Katar, B. Mashburn, K. Afkhamie, H. Latchman, and R. Newman, "Channel adaptation based on cyclo-stationary noise characteristics in PLC systems," in *Proc. IEEE Int. Symp. Power Line Commun. Appl.*, Mar. 2006, pp. 16–21.

- [121] J.-Y. Baudais and M. Crussière, "Linear precoding for multicarrier and multicast PLC," in *MIMO Power Line Communications: Narrow and Broadband Standards, EMC, and Advanced Processing* (Devices, Circuits, and Systems), L. T. Berger, A. Schwager, P. Pagani, and D. Schneider, Eds. Boca Raton, FL, USA: CRC Press, 2014, pp. 493–530. [Online]. Available: <https://hal.archives-ouvertes.fr/hal-00927043>
- [122] M. J. Rahman and L. Lampe, "Interference alignment for MIMO power line communications," in *Proc. IEEE Int. Symp. Power Line Commun. Appl. (ISPLC)*, Mar./Apr. 2015, pp. 71–76.
- [123] M. Sayed, G. Sebaali, B. L. Evans, and N. Al-Dhahir, "Efficient diversity technique for hybrid narrowband-powerline/wireless smart grid communications," in *Proc. IEEE Int. Conf. Smart Grid Commun. (SmartGrid-Comm)*, Nov. 2015, pp. 1–6.
- [124] M. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," *CoRR*, Mar. 2016. [Online]. Available: <https://arxiv.org/abs/1603.04467>
- [125] A. M. Tonello, "Wideband impulse modulation and receiver algorithms for multiuser power line communications," *EURASIP J. Adv. Signal Process.*, vol. 2007, no. 1, 2007, Art. no. 096747.
- [126] G. Bumiller, L. Lampe, and H. Hrasnica, "Power line communication networks for large-scale control and automation systems," *IEEE Commun. Mag.*, vol. 48, no. 4, pp. 106–113, Apr. 2010.
- [127] S. D'Alessandro and A. M. Tonello, "On rate improvements and power saving with opportunistic relaying in home power line networks," *EURASIP J. Adv. Signal Process.*, vol. 2012, no. 1, p. 194, Sep. 2012. doi: [10.1186/1687-6180-2012-194](https://doi.org/10.1186/1687-6180-2012-194).
- [128] A. Bradai, K. Singh, T. Ahmed, and T. Rasheed, "Cellular software defined networking: A framework," *IEEE Commun. Mag.*, vol. 53, no. 6, pp. 36–43, Jun. 2015.
- [129] S. Khodayari and M. J. Yazdanpanah, "Network routing based on reinforcement learning in dynamically changing networks," in *Proc. 17th IEEE Int. Conf. Tools Artif. Intell. (ICTAI)*, Nov. 2005, p. 5 and 366.
- [130] Y. Lv and Y. Zhu, "Routing algorithm based on swarm intelligence," in *Proc. 10th World Congr. Intell. Control Automat.*, Jul. 2012, pp. 47–50.
- [131] G. Kumar and J. Singh, "Energy efficient clustering scheme based on grid optimization using genetic algorithm for wireless sensor networks," in *Proc. 4th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT)*, Jul. 2013, pp. 1–5.
- [132] N. Guizani and J. Fayn, "Framework for intelligent message routing policy adaptation," in *Proc. IEEE Int. Conf. Autonomic Comput.*, Jul. 2015, pp. 235–238.
- [133] K.-L. A. Yau, H. G. Goh, D. Chieng, and K. H. Kwong, "Application of reinforcement learning to wireless sensor networks: Models and algorithms," *Computing*, vol. 97, no. 11, pp. 1045–1075, 2015.
- [134] F. Marcuzzi and A. M. Tonello, "Artificial intelligence based routing in PLC networks," in *Proc. IEEE Int. Symp. Power Line Commun. Appl. (ISPLC)*, Apr. 2018, pp. 1–6.
- [135] F. Marcuzzi and A. M. Tonello, "Stochastic geometry for the analysis of small radio cells and PLC back-hauling," in *Proc. 11th Int. ITG Conf. Syst., Commun. Coding (SCC)*, Feb. 2017, pp. 1–6.
- [136] 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)*, Apr. 2019, pp. 1–6.
- [137] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2627435.2670313>
- [138] M. O. Ahmed and L. Lampe, "Power line communications for low-voltage power grid tomography," *IEEE Trans. Commun.*, vol. 61, no. 12, pp. 5163–5175, Dec. 2013.
- [139] 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.
- [140] F. Passerini and A. M. Tonello, "Smart grid monitoring using power line modems: Anomaly detection and localization," *IEEE Trans. Smart Grid*, to be published.
- [141] F. Passerini and A. M. Tonello, "Smart grid monitoring using power line modems: Effect of anomalies on signal propagation," *IEEE Access*, vol. 7, pp. 27302–27312, 2019.
- [142] L. Förstel and L. Lampe, "Grid diagnostics: Monitoring cable aging using power line transmission," in *Proc. IEEE Int. Symp. Power Line Commun. Appl. (ISPLC)*, Apr. 2017, pp. 1–6.



ANDREA M. TONELLO (M'00–SM'12) received the Laurea degree (*summa cum laude*) in electrical engineering and the Ph.D. degree (by research) in electronics and telecommunications from the University of Padova, Padua, Italy, in 1996 and 2002, respectively. From 1997 to 2002, he was with Bell Labs-Lucent Technologies, Whippany, NJ, USA, first as a Member of the Technical Staff. He was promoted to a Technical Manager and appointed as the Managing Director of the Bell Labs, Italy

Division. 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 is currently a Professor with the Chair of Embedded Communication Systems, University of Klagenfurt, Klagenfurt, Austria. He received several awards, including the Bell Labs Recognition of Excellence Award, in 1999, the Distinguished Visiting Fellowship from the Royal Academy of Engineering, U.K., in 2010, the IEEE Distinguished Lecturer Award from VTS, from 2011 to 2015 and from COMSOC, from 2018 to 2019, the Italian Full Professor Habilitation, in 2013, the Chair of Excellence from Carlos III Universidad, Madrid, from 2019 to 2020. He was also a co-recipient of nine best paper awards. He was the Chair of the IEEE Communications Society Technical Committee on Power Line Communications. He is an Associate Editor of several journals, including the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, the IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE ACCESS, and *IET Smart Grid*.



NUNZIO A. LETIZIA received the Laurea Magistrale degree (*summa cum laude*) in electrical engineering from the University of Udine, Italy, in 2018. He is currently pursuing the Ph.D. degree with the University of Klagenfurt, Austria. During his studies, he received a full scholarship from the Scuola Superiore dell'Università degli studi di Udine. His research interests include mathematics and physics to communication engineering, signal processing, and statistical learning.



DAVIDE RIGHINI (S'17) received the Laurea Magistrale degree in electronic engineering from the University of Udine, Italy, in 2016. He is currently a University Assistant with the NES Institute, University of Klagenfurt, Austria. His research interests include modeling, characterization and measurement of cyber physical systems including sensor, and communication technologies.



FRANCESCO MARCUZZI received the Laurea Magistrale degree in electronic engineering from the University of Udine, Italy, in 2016. He is currently pursuing the Ph.D. degree with the University of Klagenfurt, Austria, where he is currently a University Assistant. His research interests include networking techniques, routing, powerline communication networks, communication systems, and applied machine learning.

• • •