

Physical-Layer Deep Learning: Challenges and Applications to 5G and Beyond

Francesco Restuccia, *Member, IEEE*, and Tommaso Melodia, *Fellow, IEEE*

Abstract—The unprecedented requirements of IoT networks have made fine-grained optimization of spectrum resources an urgent necessity. Thus, designing techniques able to extract knowledge from the spectrum in real time and select the optimal spectrum access strategy accordingly has become more important than ever. Moreover, 5G-and-beyond networks will require complex management schemes to deal with problems such as adaptive beam management and rate selection. Although deep learning has been successful in modeling complex phenomena, commercially-available wireless devices are still very far from actually adopting learning-based techniques to optimize their spectrum usage. In this paper, we first discuss the need for real-time deep learning at the physical layer, and then summarize the current state of the art and existing limitations. We conclude the paper by discussing an agenda of research challenges and how deep learning can be applied to address crucial problems in 5G-and-beyond networks.

I. INTRODUCTION

The wireless spectrum is undeniably one of nature's most complex phenomena. This is especially true in the highly-dynamic context of the Internet of Things (IoT), where the widespread presence of tiny embedded wireless devices seamlessly connected to people and objects will make spectrum-related quantities such as fading, noise, interference, and traffic patterns hardly predictable with traditional mathematical models. Techniques able to perform real-time fine-grained spectrum optimization will thus become fundamental to squeeze out any spectrum resource available to wireless devices.

There are a number of key issues – summarized in Figure 1 – that make existing wireless optimization approaches not completely suitable to address the spectrum challenges mentioned above. On one hand, *model-driven* approaches aim at (i) mathematically formalize the entirety of the network at different levels of the protocol stack, and (ii) optimize an objective function based on throughput, latency, jitter, and similar metrics. Although yielding optimal solutions, these approaches are usually NP-Hard, and thus, unable to be run in real time and

address spectrum-level issues. Moreover, they rely on a series of modeling assumptions (e.g., fading/noise distribution, traffic and mobility patterns, and so on) that may not always be valid in highly-dynamic IoT contexts. On the other hand, *protocol-driven* approaches consider a specific wireless technology (e.g., WiFi, Bluetooth or Zigbee) and attempt to heuristically change parameters such as modulation scheme, coding level, packet size, etc based on metrics computed in real time from pilots and/or training symbols. Protocol-driven approaches have the great advantage to be easily implementable. However, being heuristic in nature, they necessarily yield sub-optimal performance and are hardly applicable to other protocols beyond the one considered.

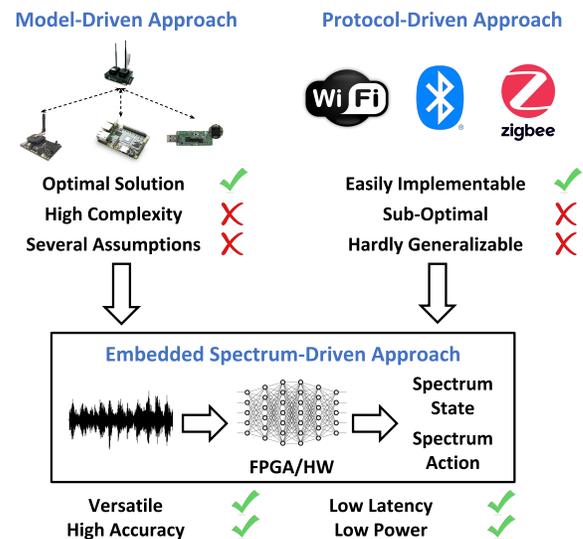


Fig. 1. Key issues in today's wireless optimization approaches.

Due to the above reasons, the wireless community has recently started to acknowledge that radically-novel propositions are needed to achieve both real-time and effective wireless spectrum optimization. This approach, called *spectrum-driven*, is rooted on this simple yet very powerful intuition; by leveraging real-time machine learning techniques implemented in the hardware portion of the wireless platform, we could design wireless systems that can *learn by themselves* the optimal spectrum actions to take given the current spectrum state, instead of relying on complex mathematical models and hard-coded protocols. Concretely speaking, the big picture

F. Restuccia and T. Melodia are with the Institute for the Wireless Internet of Things, Northeastern University, Boston, MA USA. This paper has been submitted for possible publication in the IEEE Communications Magazine. Authors emails: {frestuc, melodia}@northeastern.edu.

is to realize systems able to distinguish *on their own* different spectrum states (*e.g.*, based on noise, interference, channel occupation, and similar), and change their hardware and software fabric *in real time* to implement the optimal spectrum action (*e.g.*, modulation, coding, packet length, etc) to be performed according to the current spectrum state [1, 2].

When realized concretely, spectrum-driven optimization will realize the dream of a *cognitive radio* first envisioned more than 20 years ago by Mitola and Maguire [3]. The crude reality, however, is that so far no practical implementations of truly self-adaptive and self-resilient cognitive radios have been shown. Interestingly enough, the recent success of *physical-layer deep learning* in addressing problems such as modulation recognition [4], radio fingerprinting [5] and medium access control [6] has taken us many steps in the right direction [7, 8]. As we will explain later on, thanks to its unique theoretical and practical advantages, deep learning can really be a game-changer, especially when cast in the context of a real-time hardware-based implementation.

On the other hand, existing research has mostly focused on generating spectrum data and training models in the cloud. This, in turn, has left a number of key theoretical and system-level issues substantially unexplored. As deep learning was not conceived having the constraints and requirements of wireless communications in mind, it is still unclear what are the fundamental limitations of physical-layer deep learning and how far we can leverage its power to address ever more complex problems. For this reason, the key contribution of this paper is to propose an agenda of research opportunities to tackle existing and future challenges in this domain. Specifically, we first introduce the notion of physical-layer deep learning in Section II, and discuss the related requirements and challenges in III, as well as the existing state of the art. Next, we discuss future avenues of research in Section IV, as well as possible applications of deep learning to 5G-and-beyond networks in Section V. Finally, we draw conclusions in Section VI.

II. WHY PHYSICAL-LAYER DEEP LEARNING?

The rush of interest in deep learning from the wireless community is not without a reason. So far, learning-based techniques have been exceptionally successful in addressing classification and optimization problems where closed-form mathematical expressions are difficult or impossible to obtain [9]. Extensively employed in the computer vision and natural language processing domains, convolutional neural networks (CNNs) and recurrent neural networks (RNNs) are now being “borrowed” by wireless researchers to address handover and

power management in cellular networks, dynamic spectrum access, resource allocation/slicing/caching, video streaming, and rate adaptation, just to name a few [7].

The reader may wonder why traditional machine learning is not particularly apt to address real-time physical-layer problems. The answer lies in a combination of factors summarized in Figure 2, which also explains the key differences between the two approaches through an example. Overall, the pivotal advantage of deep learning is that it relieves from the burden of finding the right “features” characterizing a given wireless phenomenon. At the physical layer, this key advantage comes almost as a necessity for at least three reasons, which are discussed below.

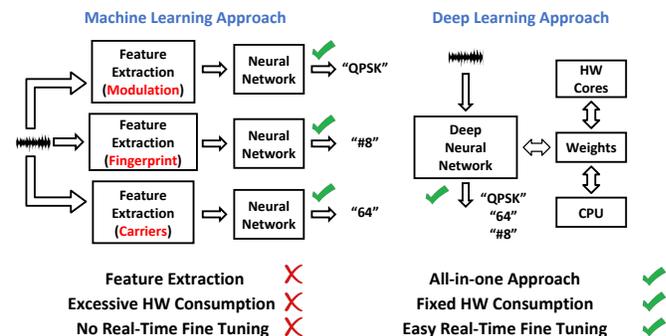


Fig. 2. Traditional vs Deep Learning for the Physical Layer.

Highly-Dimensional Feature Spaces. Classifying waveforms ultimately boils down to distinguishing small-scale *patterns* in the in-phase-quadrature (I/Q) plane, which may not be clearly separable in a low-dimensional feature space. The clearest example is the problem of radio fingerprinting, where we want to distinguish among hundreds (potentially thousands) of devices based on the unique imperfections imposed by the hardware circuitry. While legacy low-dimensional techniques can correctly distinguish up to a few dozens of devices [10], deep learning-based classifiers can scale up to hundreds of devices by learning extremely complex features in the I/Q space [11]. Similarly, O’Shea *et al.* [4] have demonstrated that on the 24-modulation dataset considered, deep learning models achieve on the average about 20% higher classification accuracy than legacy learning models under noisy channel conditions.

All-in-One Approach. The second key advantage of deep learning is that automatic feature extraction allows the system designer to reuse the same deep learning architecture – and thus, the same hardware circuit – to address different learning problems. Critically, this allows not only to save hardware resources, but also

to keep both latency and energy consumption constant, which are highly-desirable features in embedded systems design and are particular critical in wireless systems, as explained in Section II. Figure 2 shows an example where a learning system is trained to classify modulation, number of carriers and fingerprinting. While deep learning can concurrently recognize the three quantities (provided that the network has been appropriately trained), traditional learning requires different feature extraction processes for each of the classification outputs. This, in turn, increases hardware consumption and hinders fine-tuning of the learning model.

Real-Time Fine Tuning. Optimization strategies cannot be considered one-size-fits-all in today’s highly-dynamic spectrum environment. This is no exception for deep learning-based algorithms. Indeed, recent research [5, 11] has shown that the wireless channel makes it highly unlikely to deploy deep learning algorithms that will function without periodic fine-tuning of the weights. This makes the deep learning classification system *time-varying*, which is one of the main challenges of modern machine learning [12] and discussed in Section II. While deep learning “easily” accomplishes this goal by performing batch gradient descent on fresh input data [13], the same is not true for traditional machine learning algorithms, where tuning feature extraction algorithms can be extremely challenging since it would require to completely change the circuit itself.

III. PHYSICAL-LAYER DEEP LEARNING: REQUIREMENTS AND CHALLENGES

The target of this section is to discuss existing system-level challenges in physical-layer deep learning as well as the state of the art in addressing these issues. For a more detailed compendium of the state of the art, the interested reader can take a look at the comprehensive survey of the topic by Zhang *et al.* [8].

To ease the reader into the topic, we summarize at a very high level the main components and operations of a learning-based wireless device in Figure 3. The core feature that distinguishes learning-based devices is that digital signal processing (DSP) decisions are driven by deep neural networks (DNNs). In particular, in the receiver (RX) DSP chain the incoming waveform is first received and placed in an I/Q buffer (step 1). Then, a portion of the I/Q samples are forwarded to the RX DNN (step 2), which produces an inference that is used to reconfigure the RX DSP logic (step 3). For example, if a QPSK modulation is detected instead of BPSK, the RX demodulation strategy is reconfigured accordingly. Finally, the incoming waveform is released from the

I/Q buffer and sent for demodulation (step 4). At the transmitter’s side, the I/Q samples are sent to the RX DNN and to the TX DNN to infer the current spectrum state (*e.g.*, spectrum-wide noise/interference levels). As soon as the inference is produced and the TX DSP logic is changed (step 6), the TX’s buffered data is released (step 7), processed by the TX DSP logic and sent to the wireless interface (step 8).

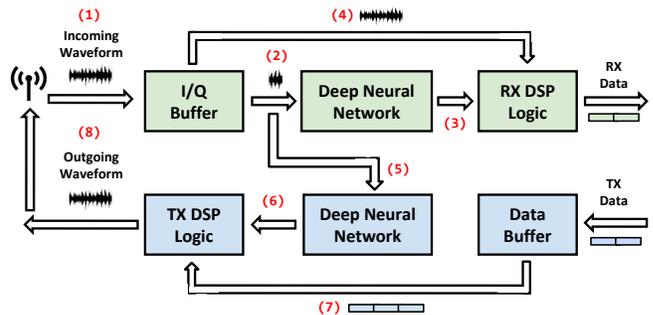


Fig. 3. Main operations in a learning-based wireless device.

We identify three core challenges in physical-layer deep learning, which are discussed below.

A. Addressing Latency and Space Constraints

Domains such as computer vision usually do not have extreme requirements in terms of maximum latency or number of weights of a deep learning model. For example, when we are uploading a picture on a social network, we do not expect a face recognition algorithm that automatically “tags” us and our friends to run under a given number of milliseconds. Moreover, software developers are usually not concerned of how depth the model is or how many parameters it uses, as long as it gives the needed accuracy performance. Very different, however, is the case of physical-layer deep learning, where digital signal processing (DSP) constraints and hardware limitations have to be heeded – in some cases, down to the clock cycle level.

The first critical issue is running the model quickly enough to avoid overflowing the I/Q buffer and/or the data buffer (see Figure 3). Just to provide the reader with some figures, consider that an incoming waveform sampled at 40MHz (*e.g.*, a WiFi channel) will generate a data stream of 160MB/s, provided that each I/Q sample is stored in a 4-byte word. With an I/Q buffer of 1MB – very large if we consider that it is implemented in hardware – the DNN must run with a latency less than 6.25ms to avoid buffer overflow. The latency becomes 6.25 μ s if we consider a more realistic buffer of 1kB.

Moreover, the second issue is that the channel and the transmitter’s behavior themselves may require the

DNN to run with very little latency. For example, if the channel coherence time is 10ms, the DNN should run with latency *much less* than 10ms to make meaningful inference. Similarly, if the DNN performs modulation recognition every 1ms, the DNN has to run with latency much less than 1ms if it wants to detect modulation changes. The examples clearly show that lower DNN latency implies (i) higher admissible sampling rate of the waveform, and thus, higher bandwidth of the incoming signal; (ii) higher capability of analyzing fast-varying channels and waveforms.

Finally, hardware resource utilization is a spinous issue. Nowadays, deep learning models usually have millions of parameters (*e.g.*, AlexNet has some 60M weights) or perhaps also tens of millions, *e.g.*, VGG-16, with about 138M. Obviously, it is hardly feasible to entirely fit these models into the hardware fabric of even the most powerful embedded devices currently available. Moreover, it is not feasible to run them from the cloud and transfer the result to the platform due to the additional delay involved. Therefore, physical-layer deep learning model have also to be relatively small to be feasibly implemented on embedded devices.

1) **State of the art:** The first work to propose a systematic investigation into the above issues is [2]. In this paper, the authors propose *RFLearn*, a hardware/software framework to integrate a Python-level CNN into the DSP chain of a radio receiver. The framework is based on high-level synthesis (HLS) and translates the software-based CNN to an FPGA-ready circuit. The advantage is that through HLS, the constraints on accuracy, latency and power consumption can be tuned based on the application. *RFLearn*'s performance and design cycle were evaluated on a custom FPGA-defined radio. As a practical case study, the authors train several models to address the problem of modulation recognition. Latency and power consumption are compared to a software-based implementation, which is shown to perform respectively 17x and 15x worse than *RFLearn*. Moreover, it is shown that accuracy of over 90% can be achieved with a model of only about 30k parameters.

The same authors integrate deep reinforcement learning (DRL) techniques at the transmitter's side by proposing *DeepWiERL* [1], a general-purpose, hybrid software/hardware DRL framework to support the training and real-time execution of state-of-the-art DRL algorithms on top of embedded devices. Moreover, *DeepWiERL* includes a novel supervised DRL model selection and bootstrap (S-DMSB) technique that leverages HLS and transfer learning to orchestrate a neural network architecture that decreases convergence time and satisfies

application and hardware constraints. The work is the first to prove the feasibility of real-time DRL-based algorithms on a wireless platform, showing superior performance with respect to software-based systems. Results also indicates that S-DMSB may improve convergence time and reward by respectively 6x and 45%.

B. Designing Features and Addressing Stochasticity

It is very well understood what deep neural networks (DNNs) actually learn as discriminating features in computer vision applications. For example, the first layers in convolutional neural networks (CNNs) are trained to detect small-scale "edges" (*i.e.*, contours of eyes, lips, etc), which become more and more complex as the network gets deeper (*i.e.*, mouth, eyes, hair type, etc) [9]. CNNs have also the advantage of being *shift-invariant*: convolved weights in each layer detect patterns in arbitrary positions in the sequence, and a max-pool layer passes the presence of a signal to a higher layer. This is precisely the property that makes these networks excellent at detecting, *e.g.*, an object or a face in an image, irrespective of where it occurs.

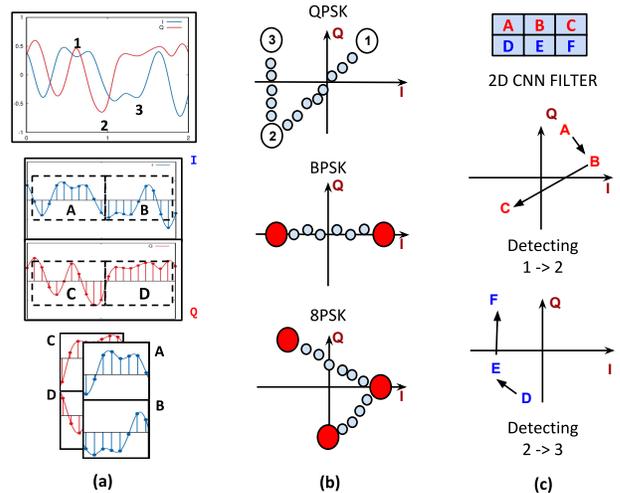


Fig. 4. (a) Construction of an input tensor of size $10 \times 10 \times 2$ from I/Q waveform; (b) Examples of transitions in the I/Q complex plane corresponding to QPSK, BPSK and 8PSK; (c) Example of a 2×3 filter of a CNN trained for BPSK vs QPSK recognition.

In the wireless domain, however, CNNs do not operate on images but on I/Q samples, implying that further investigations are needed to construct the input tensor from the I/Q samples. To make an example, Figure 4(a) shows the approach based on two-dimensional (2D) convolution proposed in [2]. Specifically, input tensors are constructed by "stacking up" H rows of W consecutive I/Q samples. To further clarify why the input tensor was constructed this way, Figure 4(b) shows examples of transitions in the I/Q complex plane corresponding to

QPSK, BPSK, and 8PSK. Particularly, we also show the transitions corresponding to the points (1) to (3) in the upper side of Figure 4(a). Figure 4(b) clearly depicts that different modulation waveforms present different transition patterns in the I/Q plane. For example, the transitions between (1, 0) and (-1, 0) peculiar to BPSK do not appear in QPSK, which presents a substantially different constellation. This can constitute a unique “signature” of the signal that can eventually be learned by the CNN filters. To point out how CNN filters can distinguish different I/Q patterns, Figure 4(c) shows an example of a 2x3 filter in the first layer of a CNN trained for BPSK vs QPSK modulation recognition. Specifically, the first row of the filter (*i.e.*, A, B, C) detects I/Q patterns where the waveform transitions from the first to the third quadrant, while the second row (*i.e.*, D, E, F) detects transitions from the third to the second quadrant.

The above and similar CNN-based approaches [4], although demonstrated to be effective, do not fully take into account that a physical-layer deep learning system is inherently *stochastic* in nature; Figure 5 summarizes the main sources of randomness. The first one is the unavoidable noise and fading that is inherent to any wireless transmission. We point out that although channel statistics could be stationary in some cases, and therefore could theoretically be learned, (i) these statistics cannot be valid in every possible network situation; (ii) a CNN cannot be trained on all possible channel distributions and related realizations; (iii) a CNN is hardly re-trainable in real-time due to its sheer size.

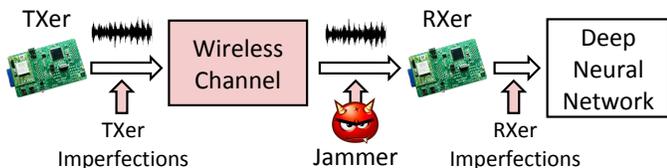


Fig. 5. Source of randomness in physical-layer deep learning.

The second factor to consider is adversarial action (*i.e.*, jamming), which may change the received signal significantly and usually, in a totally unpredictable way. The third factor is the unavoidable imperfections hidden inside the RF circuitry of off-the-shelf radios (*i.e.*, I/Q imbalance, frequency/sampling offsets, and so on). This implies that signal features can (and probably will in most cases) change over time, in some cases in a very significant way.

1) *State of the art*: The issue of stochasticity of physical-layer deep learning has been mostly investigated in the context of radio fingerprinting. This technique leverages hardware imperfections embedded in the transmitter’s DSP circuitry to uniquely identify radio

devices without relying on slow, energy-hungry cryptography. It has been shown that deep learning algorithms can outperform traditional feature-based algorithms in identifying large populations of devices [11].

Very recently, in [11] the authors proposed a large-scale investigation of the impact of the wireless channel on the accuracy of CNN-based radio fingerprinting algorithms. Specifically, the authors collected more than 7TB of wireless data obtained from 20 bit-similar wireless devices over the course of 10 days in different wireless environments. They also report results on a 400GB government dataset containing thousands of WiFi and ADS-B transmissions. The authors show that the wireless channel decreases the accuracy from 85% to 9% and from 30% to 17% in the experimental and government dataset, respectively. However, another key insight brought by the evaluation is that waveform equalization can increase the accuracy by up to 23%.

To address the issue of stochasticity, the *DeepRadioID* system [5] was recently proposed. The core intuition behind *DeepRadioID* is to leverage finite impulse response filters (FIRs) to be computed at the receiver’s side. These FIRs compensate current channel conditions by being applied at the transmitter’s side. In this way, small-scale modifications can strengthen the fingerprint without compromising the throughput significantly. The authors formulated a *Waveform Optimization Problem* (WOP) to find the optimum FIR for a given CNN. Since the FIR is tailored to the specific device’s hardware, it is shown that an adversary is not able to use a stolen FIR to imitate a legitimate device’s fingerprint. The *DeepRadioID* system was evaluated with a testbed of 20 bit-similar SDRs, as well as on two datasets containing transmissions from 500 ADS-B devices and by 500 WiFi devices. Experimental results show that *DeepRadioID* improves the fingerprinting accuracy by 27% with respect to the state of the art.

IV. PHYSICAL-LAYER DEEP LEARNING: THE WAY AHEAD

We now present an agenda of research opportunities in the field of physical-layer deep learning. Figure 6 summarizes the challenges discussed below.

A. Evaluating Physical-Layer Learning at Scale

So far, physical-layer deep learning techniques have been validated in controlled, lab-scale environments and with a limited number of wireless technologies. Although some noticeable efforts have been done to produce large-scale datasets in the area of radio fingerprinting, other physical-layer learning problems (*e.g.*,

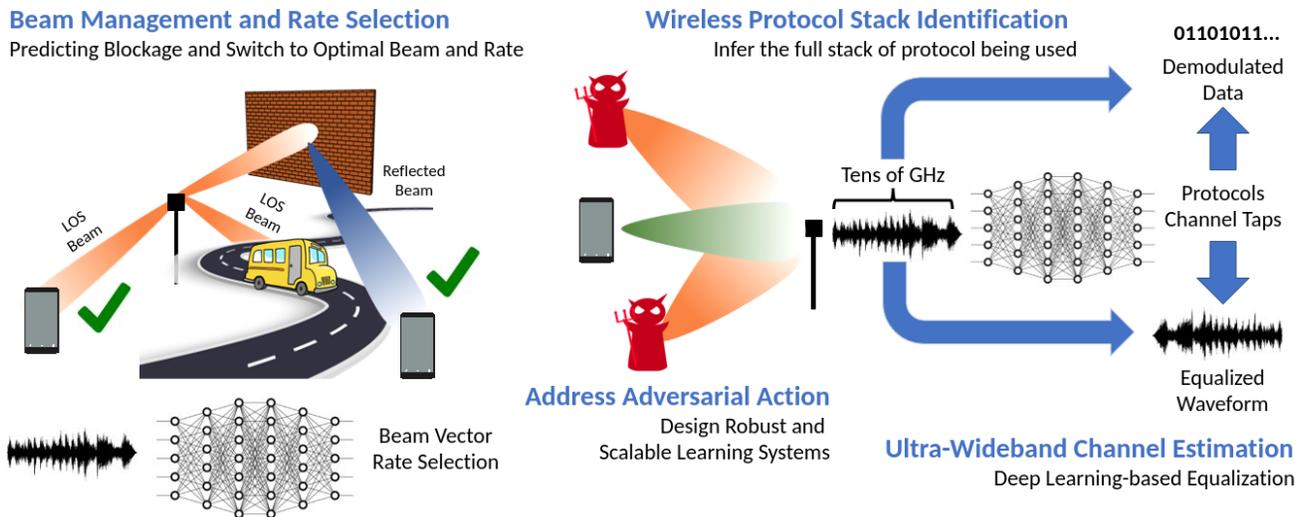


Fig. 6. Summary of Main Research Challenges in Physical-Layer Deep Learning and Applications to 5G-and-beyond Networks.

modulation recognition) have been clearly left behind. For this reason, the research community desperately needs large-scale experimentation to really understand whether these techniques can be applied in realistic wireless ecosystems where hundreds of nodes, protocols and channels will necessarily coexist.

To bring physical-layer deep learning to the next level, we are going to need “wireless data factories” able to generate I/Q data at unseen scale. The newly-developed *Platforms for Advanced Wireless Research* (PAWR) will be fundamental in addressing the above challenge [14]. The PAWR program will develop four platforms to be shared among the wireless research community, with each platform conceived to enable at-scale experimentation by supporting the technical diversity, geographical extension, and user density representative of a small city or community. The platforms will enable sub-6, millimeter-wave and drone experimentation capabilities in a multitude of real-world scenarios. Alongside PAWR, the Colosseum network emulator [15] will be soon open to the research community and provide us with unprecedented data collection opportunities. Originally developed to support DARPA’s spectrum collaboration challenge in 2019, Colosseum can emulate up to 256x256 4-tap wireless channels among 128 software-defined radios. Users can create their own wireless scenarios and thus create “virtual worlds” where learning algorithms can be truly stressed to their full capacity.

B. Addressing Wireless Adversarial Learning

Up until now, researchers have focused on improving the accuracy of the physical-layer deep learning model, without heeding security concerns. Indeed, prior work in computer vision has shown that the accuracy of a deep learning model can be significantly compromised

by crafting adversarial inputs. The first kind of attack is called *targeted*, where given a valid input, a classifier and a target class, it is possible to find an input close to the valid one such that the classifier is “steered” toward the target class. More recently, researchers have demonstrated the existence of *universal perturbation vectors*, such that when applied to the majority of inputs, the classifier steers to a class different than the original one. On the other hand, the time-varying nature of the channel could compromise adversarial attempts. Moreover, the received waveforms still need to be decodable and thus cannot be extensively modified. Therefore, additional research is needed to fill the gap between AML and the wireless domain and demonstrate *if, when, and how* adversarial machine learning (AML) is concretely possible in practical wireless scenarios.

V. PHYSICAL-LAYER DEEP LEARNING:

APPLICATIONS TO 5G-AND-BEYOND NETWORKS

The millimeter (mmWave) and Terahertz (THz) spectrum bands have become the *de facto* candidates for 5G-and-beyond communications. To fully unleash the power of these bands, mmwave/THz systems will operate with *ultra-wide spectrum bands* – in the order of several, perhaps tens of gigahertz. With spectrum bands becoming extremely wide, pilot-based channel estimation could not result to be the best strategy – both from an efficiency and effective standpoint. Indeed, frequently transmitting pilots for the whole bandwidth can lead to severe loss of throughput. A neural network, in combination with techniques such as compressive sensing, could be trained to infer the channel directly based on the I/Q samples, without requiring additional pilots. One possible strategy could be to leverage the packet headers or trailers as source of reference I/Q data to train the learning model.

Moreover, next-generation networks will necessarily require fast and fine-grained optimization of parameters at all the layers of the protocol stack. Radios will thus need to be extremely spectrum-agile, meaning that wireless protocols should be used *interchangeably* and according to the current spectrum circumstances. For example, OFDM could be the best strategy at a given moment in time, yet subsequently (*i.e.*, milliseconds later) a narrowband transmission could prove to be more effective. To demodulate incoming waveforms transmitted with different strategies, it becomes necessary to *infer* the waveform type – and thus, the wireless protocol stack being used – before feeding it to the DSP logic. It is yet to be understood whether physical-layer I/Q samples can be used to infer the whole stack of a wireless protocol.

Another major challenge of mmWave and THz communications is the severe path and absorption loss (*e.g.*, oxygen at 60 GHz). Moreover, mmWave and THz carriers cannot penetrate physical obstacles such as dust, rain, snow, and other opaque objects (people, building, transportation vehicles), making them highly susceptible to blockage. This key aspect will require high directionality of antenna radiations (*i.e.*, beamforming), which will increase the transmission range but also introduce the compelling need for proactive beam steering and rate adaptation techniques. Deep learning could be utilized to design prediction techniques that can infer in real-time an incoming blockage in a beam direction and thus proactively “steer” the beam toward another direction.

VI. CONCLUSIONS

The unprecedented scale and complexity of today’s wireless systems will necessarily require a completely new networking paradigm, where protocols and architectures will rely on data-driven techniques able to achieve fine-grained real-time spectrum optimization. In this paper, we have provided an overview of physical-layer deep learning and the state of the art in this topic. We have also introduced a roadmap of exciting research opportunities, which are definitely not easy to tackle but that if addressed, will take physical-layer deep learning to the next step in terms of capabilities. We hope that this paper will inspire and spur significant wireless research efforts in this timely and exciting field in the years to come.

REFERENCES

[1] F. Restuccia and T. Melodia, “DeepWiERL: Bringing Deep Reinforcement Learning to the Internet of Self-Adaptive Things,” *IEEE INFOCOM*, 2020.

[2] —, “Big Data Goes Small: Real-Time Spectrum-Driven Embedded Wireless Networking through Deep Learning in the RF Loop,” in *IEEE INFOCOM*, 2019.

[3] J. Mitola and G. Q. Maguire, “Cognitive Radio: Making Software Radios More Personal,” *IEEE Personal Communications*, vol. 6, no. 4, pp. 13–18, Aug 1999.

[4] T. J. O’Shea, T. Roy, and T. C. Clancy, “Over-the-Air Deep Learning Based Radio Signal Classification,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 168–179, Feb 2018.

[5] F. Restuccia, S. D’Oro, A. Al-Shawabka, M. Belgiovine, L. Angioloni, S. Ioannidis, K. Chowdhury, and T. Melodia, “Deep-RadioID: Real-Time Channel-Resilient Optimization of Deep Learning-based Radio Fingerprinting Algorithms,” in *ACM MobiHoc*, 2019, pp. 51–60.

[6] O. Naparstek and K. Cohen, “Deep Multi-user Reinforcement Learning for Distributed Dynamic Spectrum Access,” *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 310–323, 2019.

[7] J. Jagannath, N. Polosky, A. Jagannath, F. Restuccia, and T. Melodia, “Machine Learning for Wireless Communications in the Internet of Things: A Comprehensive Survey,” *Ad Hoc Networks*, vol. 93, p. 101913, October 2019.

[8] C. Zhang, P. Patras, and H. Haddadi, “Deep Learning in Mobile and Wireless Networking: A Survey,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2224–2287, Third Quarter 2019.

[9] Y. LeCun, Y. Bengio, and G. Hinton, “Deep Learning,” *Nature*, vol. 521, no. 7553, p. 436, 2015.

[10] T. D. Vo-Huu, T. D. Vo-Huu, and G. Noubir, “Fingerprinting Wi-Fi Devices Using Software Defined Radios,” in *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks*. ACM, 2016, pp. 3–14.

[11] A. Al-Shawabka, F. Restuccia, S. D’Oro, T. Jian, B. C. Rendon, N. Soltani, J. Dy, K. Chowdhury, S. Ioannidis, and T. Melodia, “Exposing the Fingerprint: Dissecting the Impact of the Wireless Channel on Radio Fingerprinting,” *Proc. of IEEE Conference on Computer Communications (INFOCOM)*. Available at <http://tinyurl.com/RadioINFOCOM>, 2020.

[12] G. Ditzler, M. Roveri, C. Alippi, and R. Polikar, “Learning in Nonstationary Environments: A Survey,” *IEEE Computational Intelligence Magazine*, vol. 10, no. 4, pp. 12–25, 2015.

[13] F. Chollet, *Deep Learning with Python*. Manning Publications Co., 2017.

[14] “Platforms for Advanced Wireless Research (PAWR),” <https://advancedwireless.org>, 2020.

[15] “The Colosseum Network Emulator,” <https://www.colosseum.net>, 2020.

Francesco Restuccia (M’16) is an Associate Research Scientist at the Institute for the Wireless Internet of Things, Northeastern University, Boston, USA. His research interests lie in the design of protocols and architectures for embedded wireless systems and embedded artificial intelligence. Dr. Restuccia has published extensively in these areas and serves as TPC member and reviewer for top IEEE and ACM venues.

Tommaso Melodia (F’18) is the William Lincoln Smith Chair Professor at Northeastern University. He is the Director of Research for the PAWR Project Office. Prof. Melodia’s research focuses on modeling, optimization, and experimental evaluation of wireless systems. He is the Editor in Chief of Computer Networks, an Associate Editor for the IEEE Transactions on Wireless Communications, the IEEE Transactions on Mobile Computing, and the IEEE Transactions on Biological, Molecular, and Multi-Scale Communications.