Illustration, istockphoto.com

TOWARD POLYMORPHIC INTERNET OF THINGS RECEIVERS THROUGH REAL-TIME WAVEFORM-LEVEL DEEP LEARNING

Francesco Restuccia and Tommaso Melodia *Institute for the Wireless Internet of Things Northeastern University, Boston, MA, USA*

Editors: Nicholas D. Lane and Xia Zhou

Excerpted from "PolymoRF: polymorphic wireless receivers through physical-layer deep learning" from *Proceedings of the Twenty-First International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing* (Mobihoc '20) with permission. https://dl.acm.org/doi/10.1145/3397166.3409132 ©ACM 2020

Wireless systems such as the Internet of Things (IoT) are changing the way we interact with the cyber and the physical world. As IoT systems become more and more pervasive, it is imperative to design wireless protocols that can effectively and efficiently support IoT devices and operations. On the other hand, today's IoT wireless systems are based on inflexible designs, which makes them inefficient and prone to a variety of wireless attacks. In this paper, we introduce the new notion of a deep learning-based polymorphic IoT receiver, able to reconfigure its waveform demodulation strategy itself in real time, based on the inferred waveform parameters. Our key innovation is the introduction of a novel embedded deep learning architecture that enables the solution of waveform inference problems, which is then integrated into a generalized hardware/software architecture with radio components and signal processing. Our polymorphic wireless receiver is prototyped on a custom-made software-defined radio platform. We show through extensive over-the-air experiments that the system achieves throughput within 87% of a perfect-knowledge Oracle system, thus demonstrating for the first time that polymorphic receivers are feasible.

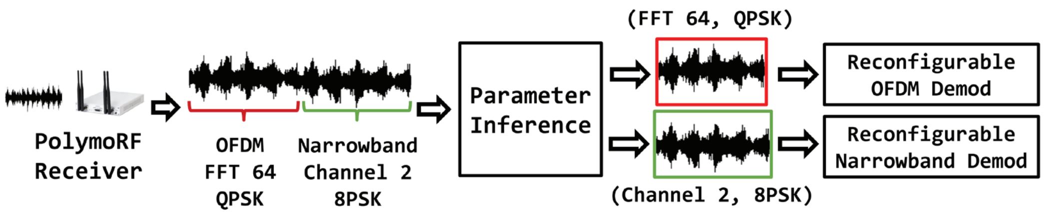

FIGURE 1. Example of a self-adaptive polymorphic receiver.

It has been forecast that more than 50 billion mobile devices will be soon connected to the Internet of Things (IoT) [1]. A clear side effect of this unprecedented growth is the potentially disruptive levels of interference that IoT devices will impose on each other [2]. Although Mitola and Maguire first envisioned the concept of "cognitive radios" 20 years ago [3], today's commercial wireless devices still use inflexible wireless standards such as Wi-Fi and Bluetooth – and thus, are still very far from being truly real-time reconfigurable. From a security perspective, another key issue is perhaps even more worrisome. It has been extensively demonstrated that jamming strategies targeting the inflexibility of key components of the wireless transmission, such as headers and pilots, can significantly decrease the system throughput while increasing the jammer stealthiness. For example, Clancy [4] demonstrated that pilot nulling attacks in OFDM systems can be up to 7.5dB more effective than traditional jamming. Moreover, Vo et al. [5] show that short bursts across carefully selected Wi-Fi sub-carriers can destroy more than 95% of Wi-Fi transmissions with an energy cost three orders of magnitude less than the communicating nodes.

Intuitively, the issues of existing communication systems could be addressed by allowing transmitters to dynamically switch parameters such as carrier frequency, FFT size, and symbol modulation without coordination with the receiver – in other words, *polymorphically* adapt to the transmitter's behavior. This will allow the transmitter (i) efficient spectrum occupation by using the most appropriate wireless scheme at any given moment, and (ii) change position of header and pilots over time, thus becoming less jamming-prone. Figure 1 shows an example of a polymorphic receiver able to infer the current transmitter's physical-layer scheme (e.g., OFDM vs narrowband) and the scheme's parameters (e.g., FFT size, channel, modulation), and then demodulate each portion of the signal.

Novelty and Contribution. This paper's key innovation is to finally bridge the gap between the extensive theoretical research on cognitive radios and the associated system-level challenges, by demonstrating that inference-based wireless communications are indeed feasible on off-the-shelf embedded devices. The main purpose of this work is to provide a blueprint for next-generation wireless receivers, where their radio hardware and software are not protocol-specific, but instead spectrum-driven and adaptable on-the-fly to different waveforms. Specifically, in this paper, we design a novel learning architecture called *RFNet*, specifically and carefully tailored for the embedded RF domain. Our key intuition in RFNet is to arrange I/Q samples to form an "image" that can be effectively analyzed by the convolutional layers. This operation produces high-dimensional representations of small-scale transition in the I/Q complex plane. We integrate RFNet into a generalized hardware/software architecture with radio components and signal processing. We prototype our system on a ZYNQ-7000 system-on-chip (SoC) and analyze its performance on a scheme where the transmitter can switch among 3 FFT sizes and 3 symbol modulation schemes without explicit notification to the receiver. A demo video where the transmitter switches FFT size every 0.5s is available at https://youtu.be/5vf_pb0nvKk. We believe ours is the first demonstration of real-time OFDM reconfigurability without explicit transmitter/receiver coordination. Experiments show that the system achieves at least 87% of the throughput of a perfect-knowledge – and thus, unrealistic – Oracle OFDM system, thus proving the feasibility of polymorphic receivers.

BACKGROUND AND CHALLENGES

Learning-based radios are envisioned to be able to automatically infer the current spectrum status in terms of occupancy [6], interference [7] and malicious activities [8]. Most of the existing work is based on low-dimensional machine learning [9], which requires the cumbersome manual extraction of very complex, ad hoc features from the waveforms. For this reason, deep learning has been proposed as a viable alternative to traditional learning techniques [10]. The key problem of RF modulation recognition through deep learning has been extensively investigated [11–13]. The seminal work by O'Shea et al. [11] and Karra et al. [14] proposed ConvNets-based to address the issue. However, the authors do not address the issue of what to do with the inferred RF information. Conversely, Kulin et al. present in [13] a framework for end-to-end wireless deep learning, where a use case on dynamic spectrum access is provided. The above work proposes models leveraging a significant number of parameters, thus ultimately not applicable to real-time RF settings. Recently, [15] has demonstrated the need for real-time hardware-based RF deep learning. However, the main limitation of [15] is that it focuses on the learning aspect only, ultimately not addressing the problem of connecting real-time inference with receiver reconfigurability.

Doing away with explicit coordination and inflexible physical layers is the first step toward wireless receivers able to self-adapt to demodulate many waveforms with a single radio interface [16, 17]. Yet, despite their compelling necessity, these wireless receivers do not exist today. Achieving this goal required us to address a set of key research challenges summarized below.

(1) Keeping Up with the Transmitter. A crucial aspect is the real-time parameter inference. In practical systems, however, transmitters may choose to switch its parameter configuration in the order of milliseconds (e.g., frequency hopping, rate adaptation). For example, if the transmitter chooses to switch modulation every 100ms, the learning model should run in (much) less than 100ms to predict the parameters and morph the receiver into a new configuration. It was shown in [15] that CPU latency is several orders of magnitude greater than what is required to sustain realistic sampling rates from the RF interface. Thus, we need hardware-based designs to implement low-latency knowledge extraction techniques.

(2) Creating Learning Architectures for the Embedded RF Domain. Recent advances in RF deep learning [11–14, 18, 19] have demonstrated that convolutional neural networks (ConvNets) may be applied to analyze RF data without feature extraction and selection algorithms [20–23]. Moreover, ConvNets present a number of characteristics (discussed in Section 4) that make them particularly desirable from a hardware implementation perspective. However, these solutions cannot be applied to implement real-time poly-morphic wireless communications – existing art [11, 18] utilizes general-purpose architectures with a very high number of parameters, requiring hardware resources and latency that go beyond what is acceptable in the embedded domain. This crucial issue

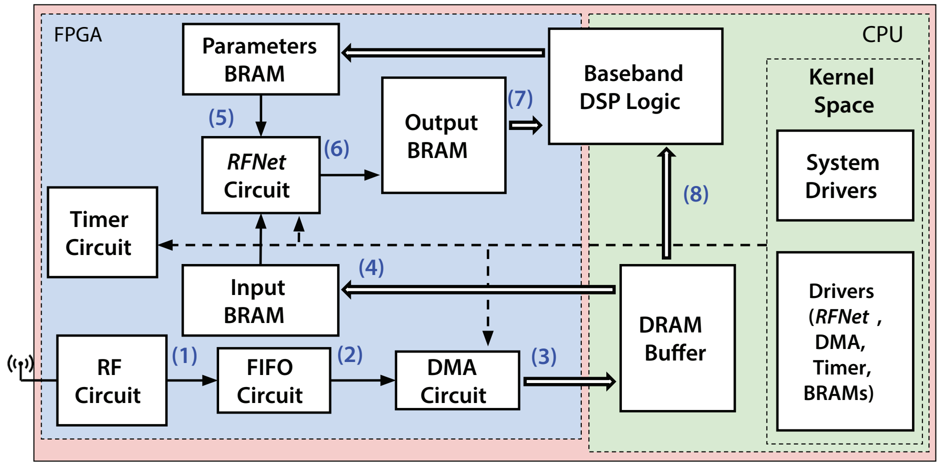

FIGURE 2. Modules and operations.

calls for novel, RF-specific, real-time architectures. We are not aware of learning systems tested in a real-time wireless environment and used to implement inference-based wireless systems.

(3) System-level Feasibility of Polymorphic Platforms. It is yet to be demonstrated whether polymorphic platforms are feasible and effective. This is not without a reason – from a system perspective, it required us to tightly interconnect traditionally separated components, such as CPU, RF front-end, and embedded operating system/kernel, to form a seamlessly running low-latency learning architecture closely interacting with the RF components and able to adapt at will its hardware and software based on RF-based inference. Furthermore, since polymorphic wireless systems are subject to inference errors, we need to test its performance against a perfect-knowledge (thus, ideal and not implementable) system.

SYSTEM OVERVIEW

The primary operations performed by our polymorphic IoT receiver platform are summarized in Figure 2. In a nutshell, the system can be considered as a full-fledged learning-based software-defined radio architecture where both the inference system and the demodulation strategy can be morphed into new configurations at will.

We provide a walk-through of the main operations with the help of Figure 2. Although for simplicity we refer to specific hardware equipment and circuits in our explanation, we point out that the building blocks of our platform design (BRAMs, DMA, FIFOs, etc.) can be implemented in any commercially available FPGA platform. We assume the transmitter may transmit by choosing among a discrete set of physical-layer parameters, which are known at the receiver's side. Physical-layer parameters may be changed at will by the transmitter but not before a minimum switching time. For the sake of generality, in this paper we will not assume any particular strategy in the transmitter's parameter choice, which can be driven by a series of factors (including anti-jamming strategy, noise avoidance, throughput optimization, and so on) that will be considered as out of the scope of this paper, whose main focus is instead on the receiver's side.

(1) Reconfigurable Radio Front-end. The RF signal is received (step 1) through a reconfigurable RF front-end. In our prototype, we used an AD9361 [24] radio interface, which supports frequency range between 70 MHz to 6.0 GHz and channel bandwidth between 200 kHz to 56 MHz. We chose the AD9361 because it is commonly used in software-defined radio systems – indeed, it is also used by USRPs such as the E310 and B210. Moreover, the AD9361 provides basic FPGA reference designs and kernel-space drivers to ease prototyping and extensions. Perhaps more importantly, the AD9361 local oscillator (LO) frequency and RF bandwidth can be reconfigured at will through CPU registers.

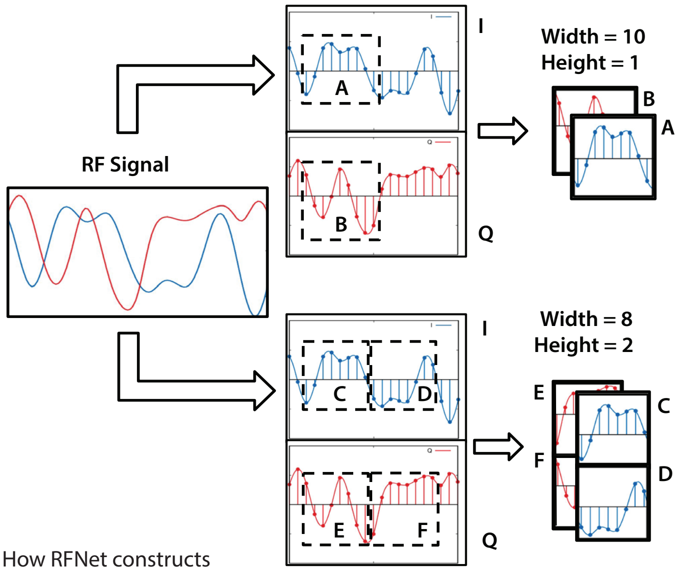

FIGURE 3. How RFNet constructs tensors from I/Q samples.

(2) Conversion from RF to FPGA domain. The AD9361 produces streams of I/Q samples of 200M samples/second – hence, it is clocked at 200 MHz. Since the AD9361 clock would be too fast for the other circuits in the FPGA, we implemented a FIFO to adapt the speed of samples from the AD9361 to the 100 MHz clock frequency used by the other circuits in the FPGA (step 2). We then use a direct memory access (DMA) core to store the stream of I/Q samples to a buffer in the DRAM (step 3). The use of DMA is crucial as the CPU cannot do the transfer itself, since it would be fully occupied for the entire duration of the read/write operation, and thus unavailable to perform other work. Therefore, we wrote a custom DMA driver to periodically fill a buffer of size B residing in the DRAM with a subset of I/Q samples coming from the FIFO.

(3) Learning and Receiver Polymorphism. After the buffer has been replenished, the first I/Q samples are sent to a BRAM (step 4) constituting the input to RFNet, a novel learning architecture based on ConvNets. The parameters of RFNet are read by an additional BRAM (step 5), which in effect allows the reconfiguration of RFNet to address multiple RF problems according to the current platform need. As explained in Section 4, RFNet produces a probability distribution over the transmitter's parameter set. After RFNet has inferred the transmitter's parameters, it writes on a block-RAM (BRAM) its probability distribution (step 6). Then, the baseband DSP logic (which may be implemented in both hardware and software) reads the distribution from the BRAM (step 7), selects the parameter set with highest probability, and "morphs" into a new configuration to demodulate the I/Q samples in the buffer (step 8).

DEEP WAVEFORM LEARNING

Deep learning relieves the burden of finding the right "features" characterizing a given wireless phenomenon. At the physical layer, this is a key advantage for the following reasons. First, deep learning offers high-dimensional feature spaces. In particular, O'Shea et al. [11] 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. Second, automatic feature extraction allows the reuse of the same hardware circuit to address different learning problems. Critically, this allows both latency and energy consumption to stay constant, which are particularly critical in wireless systems. Third, deep learning algorithms can be fine-tuned by performing batch gradient descent on fresh input data, avoiding manual retuning of the feature extraction algorithms. There are several primary advantages that make the usage of ConvNet–based models particularly desirable for the embedded RF domain. First, convolutional filters are designed to interact only with a very small portion of the input. This key property allows the achievement of significantly higher accuracy than traditional neural networks [15]. Perhaps even more importantly, ConvNets are scalable with the input size. For example, for a 200x200 input and a DL with 10 neurons, a traditional neural network will have $200^2 \cdot 10 = 400$k weights, which implies a memory occupation of $4 \cdot 400$k = 16 Mbytes to store the weights of a single layer (i.e., a float number for each weight). Clearly, this is unacceptable for the embedded domain, as the network memory consumption would become intractable as soon as several DLs are stacked on top of the other.

There are a number of key challenges in RF learning that are substantially absent in the CV domain. Among others, we know that RF signals are continuously subject to dynamic (and usually unpredictable) noise/interference coming from various sources. This may decrease the accuracy of the learning model. For example, portions of a QPSK transmission could be mistaken for 8PSK transmissions since they share part of their constellations. We address the above core design issues with the following intuitions. First, although RF signals are affected by fading/noise, in most practical cases their effect can be considered as constant over small intervals. Second, though some constellations are similar to each other, the transitions between the symbols of the constellations are distinguishable when the waveform is sampled at a higher sampling rate than the one used by the transmitter. Third, convolution operations are equivariant to translation, so they can recognize I/Q patterns regardless of where they occur. By leveraging these key concepts, we can design a learning system that distinguishes waveforms by recognizing transitions in the I/Q complex plane regardless of where they happen, by leveraging the shift-invariance property of convolutional layers and feeding discrete-time complex-valued I/Q sequence to the ConvNet. Figure 3 depicts an example of a 2x4 and 1x3 filters operating on a waveform, while Figure 4 shows the complete architecture of RFNet. Similar to existing work [11] and computer vision-based models, the network is composed by M convolutional (Conv) layers with *C* filters each, followed by rectified linear units (ReLU) as activation functions. These are then followed by a series of dense layers, each having *D* neurons. The final layer is a softmax output, which gives the probability distribution over the set of all possible classes.

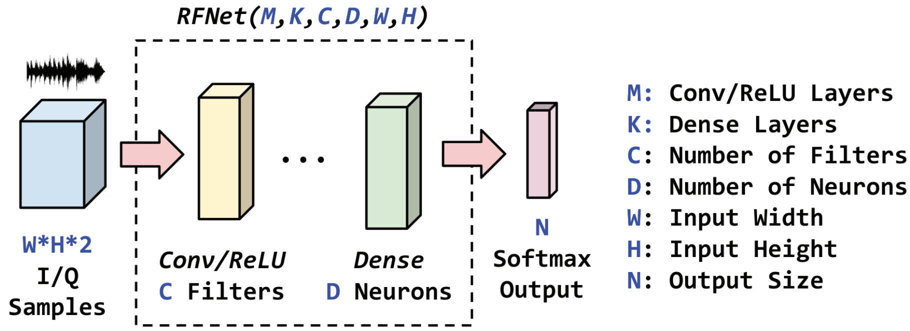

FIGURE 4. RFNet Architecture.

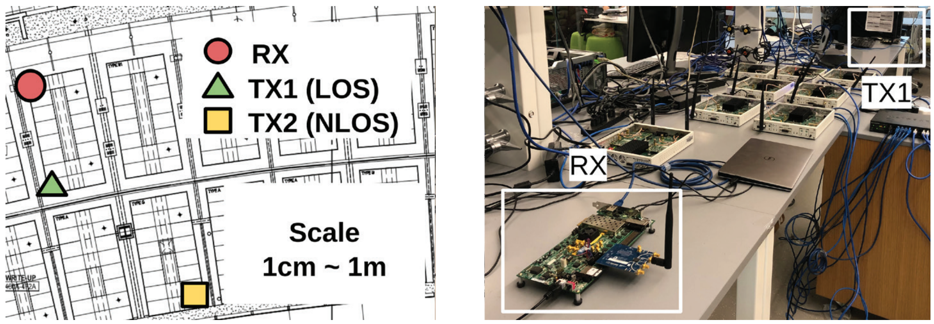

FIGURE 5. Left: Placement of the radios for experimental evaluation. Right: Experimental setting.

PROTOTYPE AND EVALUATION

Our prototype is entirely based on off-the-shelf equipment. Specifically, we use a Xilinx Zynq-7000 XC7Z045-2FFG900C system-on-chip (SoC), which is a circuit integrating CPU, FPGA and I/O all on a single substrate [25]. We chose an SoC since it provides significant flexibility in the FPGA portion of the platform, thus allowing us to fully evaluate the trade-offs during system design. Moreover, the Zynq-7000 fully supports embedded Linux, which in effect makes the ZC706 a good prototype for a wireless platform. Our Zynq-7000 contains two ARM Cortex-A9 MPCore CPUs and a Kintex-7 FPGA [26], running on top of a Xilinx ZC706 evaluation board [27]. To study our system under realistic channel environments, we have used the experimental setup shown in Figure 5. These scenarios investigate a line-of-sight (LOS) configuration where the transmitter is placed approximately 3m from the receiver, and a challenging non-line-of-sight (NLOS) channel condition where the transmitter is placed at 7m from the receiver and in the presence of several obstacles between them. Thus, the experiments were performed in a contested wireless environment with severe interference from nearby Wi-Fi devices as well as multipath effect.

As far as the data collection and testing process is concerned, we first constructed a ~10GB dataset by collecting waveform data in the line-of-sight (LOS) configuration, then used this data to train RFNet through Keras. Then, we tested our models on live-collected data in both LOS and NLOS conditions. The transmitter radio used was a Zedboard equipped with an AD9361 as RF front-end and using Gnuradio for baseband processing. Waveforms were transmitted at center frequency of 2.432 GHz (i.e., Wi-Fi's channel 5). To train RFNet, we use an ℓ_2 regularization parameter $\lambda = 0.0001$. We also use an Adam optimizer with a learning rate of $l = 10^{-4}$ and categorical cross-entropy as a loss function. All architectures are implemented in Python, on top of the Keras framework and with Tensorflow as the backend engine.

We evaluated our approach on an OFDM system (in short, Poly-OFDM) which supports 3 different FFT sizes (64, 128, 256) and 3 different symbol modulations in the FFT bins (BPSK, QPSK, 8PSK), creating in total a combination of 9 different parameter sets, which are switched pseudo-randomly by the transmitter. A demo video where the transmitter switches FFT size every 0.5s is available at https://youtu.be/5vf_pb0nvKk. In the following, we use the C=25,25, 20x20, pipelined RFNet architecture, which presents latency of about 17ms [28]. In these

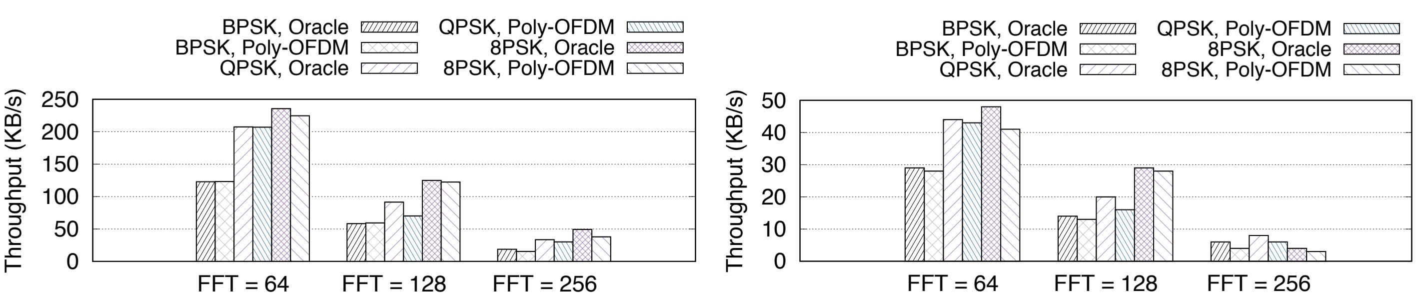

FIGURE 6. Comparison between Oracle and Poly-OFDM (left) LOS and (right) NLOS scenarios.

experiments, we set (i) the transmitter's sampling rate to 5M samples/sec; the buffer size is set to 250k I/Q samples; (iii) the switching of the transmitter to 250ms. Thus, RFNet is run approximately five times during each switching time.

The most critical aspect to be evaluated is how Poly-OFDM, an inference-based system, compares with an ideal system that has perfect knowledge of the modulation and FFT size being used by the transmitter at each time, which we call *Oracle*, for simplicity. Although Oracle cannot be implemented in practice, we believe this experiment is crucial to understand what the throughput loss with respect to a system is where the physical-layer configuration is known a priori. Figure 6 shows the comparison between Oracle and Poly-OFDM as a function of the FFT size and the symbol modulation. We notice that the overall throughput results decrease in the NLOS scenario, which is expected, given the impairments imposed by the challenging channel conditions. On the other hand, the results in Figure 6 confirm that Poly-OFDM is able to obtain similar throughput performance with that of a traditional OFDM system, obtaining on the average 90% and 87% throughput of that of the traditional system. ■

Acknowledgements

This work is supported in part by the Office of Naval Research (ONR) under contracts N00014-18-9-0001. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements of the ONR or the U.S. Government.

Francesco Restuccia is an Assistant Professor of Electrical and Computer Engineering at Northeastern University, and a member of the Institute for the Wireless Internet of Things. His research interests are at the intersection of wireless networks, embedded systems, and artificial intelligence.

Tommaso Melodia is the William L. Smith Professor of Electrical and Computer Engineering at Northeastern University, and the Director of the Institute for the Wireless Internet of Things. His research interests are in modeling, optimization, and experimental evaluation of wireless networked systems. He is an IEEE Fellow and a recipient of the NSF CAREER award.

REFERENCES

[1] Cisco Systems, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016-2021 White Paper." 2017. http://tinyurl.com/zzo6766.

[2] M.D.V. Peña, J.J.Rodriguez-Andina, and M. Manic. 2017. The Internet of Things: The role of reconfigurable platforms. *IEEE Industrial Electronics Magazine*, vol. 11, no. 3, 6–19.

[3] J. Mitola and G.Q. Maguire. 1999. Cognitive radio: Making software radios more personal. *IEEE Personal Communications*, vol. 6, no. 4, 13–18.

[4] T.C. Clancy, 2011. Efficient OFDM denial: Pilot jamming and pilot nulling. *Proceedings of IEEE International Conference on Communications (ICC).*

[5] T.D. Vo-Huu, T.D. Vo-Huu, and G. Noubir. 2016. Interleaving jamming in Wi-Fi Networks. *Proceedings of ACM Conference on Security & Privacy in Wireless and Mobile Networks (WiSec).*

[6] S. Subramaniam, H. Reyes, and N. Kaabouch, 2015. Spectrum occupancy measurement: An autocorrelation based scanning technique using USRP. *Proceedings of IEEE Annual Wireless and Microwave Technology Conference (WAMICON).*

[7] Y. Chen and H.-S. Oh. 2016. A survey of measurement-based spectrum occupancy modeling for cognitive radios. *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, 848–859.

[8] X. Jin, J. Sun, R. Zhang, Y. Zhang, and C. Zhang. Dec. 2018. SpecGuard: Spectrum misuse detection in dynamic spectrum access systems. *IEEE Transactions on Mobile Computing*, vol. 17, no. 12, 2925–2938.

[9] W. Xiong, P. Bogdanov, and M. Zheleva. 2019. Robust and efficient modulation recognition based on local sequential IQ features. *Proceedings of IEEE Conference on Computer Communications (INFOCOM).*

[10] Q. Mao, F. Hu, and Q. Hao. 2018. Deep learning for intelligent wireless networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, 2595–2621.

[11] T.J. O'Shea, T. Roy, and T.C. Clancy. 2018. Over-the-air deep learning based radio signal classification. *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, 168–179.

[12] T.J. O'Shea and J. Hoydis. 2017. An introduction to deep learning for the physical layer. *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, 563–575.

[13] M. Kulin, T. Kazaz, I. Moerman, and E.D. Poorter, 2018. End-to-end learning from spectrum data: A deep learning approach for wireless signal identification in spectrum monitoring applications. *IEEE Access*, vol. 6, 18 484–18 501.

[14] K. Karra, S. Kuzdeba, and J. Petersen. 2017. Modulation recognition using hierarchical deep neural networks. In *Proceedings of IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN).*

[15] F. Restuccia and T. Melodia, 2019. Big data goes small: Real-time spectrum-driven embedded wireless networking through deep learning in the RF loop. *Proceedings of IEEE Conference on Computer Communications (INFOCOM).*

[16] F. Restuccia, S. D'Oro, and T. Melodia. Dec. 2018. Securing the Internet of Things in the age of machine learning and software-defined networking. *IEEE Internet of Things Journal*, vol. 5, no. 6, 4829–4842.

[17] F. Restuccia and T. Melodia. 2020. Deep learning at the physical layer: System challenges and applications to 5G and beyond. *IEEE Communications Magazine*, vol. 58, no. 10, 58–64.

[18] T.J. O'Shea, J. Corgan, and T.C. Clancy. 2016. Convolutional radio modulation recognition networks. *Proceedings of the International Conference on Engineering Applications of Neural Networks*. Springer, 213–226.

[19] F. Restuccia and T. Melodia, 2020. DeepWiERL: Bringing deep reinforcement learning to the internet of self-adaptive things. *Proceedings of IEEE Conference on Computer Communications (INFOCOM).*

[20] J.L. Xu, W. Su, and M. Zho. 2010. Software-defined radio equipped with rapid modulation recognition. *IEEE Transactions on Vehicular Technology*, vol. 59, no. 4, 1659–1667.

[21] S.U. Pawar and J.F. Doherty. 2011. Modulation recognition in continuous phase modulation using approximate entropy. *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 3, 843–852.

[22] Q. Shi and Y. Karasawa. April 2012. Automatic modulation identification based on the probability density function of signal phase. *IEEE Transactions on Communications*, vol. 60, no. 4, 1033–1044.

[23] S. Ghodeswar and P.G. Poonacha. 2015. An SNR estimation based adaptive hierarchical modulation classification method to recognize M-ary QAM and M-ary PSK signals. *Proceedings of International Conference on Signal Processing, Communication and Networking (ICSCN).*

[24] Analog Devices Incorporated. 2018. AD9361 RF agile transceiver data sheet, http://www.analog.com/media/en/technical-documentation/data-sheets/AD9361.pdf.

[25] R.F. Molanes, J.J. Rodríguez-Andina, and J. Fariña. 2018. Performance characterization and design guidelines for efficient processor – FPGA communication in Cyclone V FPSoCs. *IEEE Transactions on Industrial Electronics*, vol. 65, no. 5, 4368–4377.

[26] Xilinx Inc., "Zynq-7000 SoC Data Sheet: Overview." 2018. https://www.xilinx.com/support/documentation/data_sheets/ds190-Zynq-7000-Overview.pdf.

[27] ——, "ZC706 Evaluation Board for the Zynq-7000 XC7Z045 All Programmable SoC User Guide," 2018. https://www.xilinx.com/support/documentation/boards_and_ kits/zc706/ug954-zc706- eval- board- xc7z045- ap- soc.pdf.

[28] F. Restuccia and T. Melodia. 2020. PolymoRF: Polymorphic wireless receivers through physical- layer deep learning. *Proceedings of the Twenty-First International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, 271–280.