A Photonics-Inspired Compact Network: Toward Real-Time AI Processing in Communication Systems

Hsuan-Tung Peng, Joshua C. Lederman, Lei Xu, Thomas Ferreira de Lima, Chaoran Huang, *Member, IEEE*, Bhavin J. Shastri, David Rosenbluth, and Paul R. Prucnal, *Life Fellow, IEEE*

(Invited Paper)

***Abstract*—Machine learning methods are ubiquitous in communication systems and have proven powerful for applications including radio-frequency (RF) fingerprinting, automatic modulation classification, and signal recovery in communication systems. However, the high throughput requirement of a communication link makes AI models difficult to implement in real-time on edge devices. In this work, we address this issue by improving both the algorithm and hardware to target real-time AI processing in communication systems. For algorithm development, we propose the first compact deep network consisting of a silicon photonic recurrent neural network model in combination with a simplified convolutional neural network classifier to identify RF emitters by their random transmissions. Our model achieves 96.32% classification accuracy over a set of 30 identical ZigBee devices when using 50 times fewer training parameters than an existing state-of-the-art CNN classifier (Merchant et al., 2018). Thanks to the large reduction in network size, we emulate the system using a small-scale FPGA board, the PYNQ-Z1, and demonstrate real-time RF fingerprinting with 0.219 ms latency. In addition, for hardware implementation, we further demonstrate a fully-integrated silicon photonic neural network for fiber nonlinearity compensation (Huang et al., 2021), which improves the received signal by 0.60 dB.**

***Index Terms*—Fiber nonlinear dispersion compensation, RF fingerprinting, silicon photonic neural network.**

I. Introduction

EVER-INCREASING demand for increased bandwidth and reduced latency in radio-frequency and fiber-optic communications systems has resulted in recent years in a variety of novel networks architectures and processing approaches, which often takes advantage of machine learning (ML) to improve communication system performance. Cognitive radio networks intelligently allocate bandwidth to individual transmitters, and may harness machine learning to ensure network security and prevent attacks [1], [3]. Undersea fiber communication systems experience nonlinear distortions which degrade performance but which may be corrected using an ML approach. These two key examples demonstrate the application of uniting ML technology with communications systems.

In both cases, minimizing latency is key to maximizing the user experience and managing the network effectively. However, most of the ML models for communication systems rely on large neural network size to fully capture features from the input data. This makes it difficult to implement in real-time. In addition, conventional digital electronic systems are limited in latency by the longest single-threaded operation required for the ML computation and the clock-frequency of the system (in turn limited by the fundamental RC time constant of transistor-based systems). Therefore, compact neural network models and new ML hardware platforms which can process the full bandwidth of incoming communication signals without compromising on latency are of high interest.

In this work, we contribute to both algorithm invention and hardware development to demonstrate the potential of silicon photonics neural network (PNN) for low-latency AI processing in communication systems. Our main contributions in this work are summarized as follows:

1) We propose a silicon photonic recurrent neural network (PRNN) model to learn more expressive features and enable a more compact NN classifier for RF fingerprinting [1]. This novel model largely reduces the parameters without sacrificing classification accuracy.
2) We implement the proposed compact neural network on a small-scale FPGA to demonstrate real-time processing for RF fingerprinting with 0.219 ms latency.
3) We provide a detailed review on our recent experimental demonstration of a silicon PNN chip to compensate fiber nonlinearity in undersea optical communication systems [2], [4] with comparable performance to the state-of-the-art technology.

Manuscript received 1 October 2021; revised 12 June 2022; accepted 28 July 2022. Date of publication 2 August 2022; date of current version 18 August 2022. This work was supported by DARPA PEACH under Grant AWD1006165. *(Corresponding author: Hsuan-Tung Peng.)*

Hsuan-Tung Peng, Joshua C. Lederman, Lei Xu, and Paul R. Prucnal are with the Department of Electrical and Computer Engineering, Princeton University, Princeton, NJ 08540 USA (e-mail: hpeng@princeton.edu; joshuacl@princeton.edu; leixu@princeton.edu; prucnal@princeton.edu).

Thomas Ferreira de Lima is with the Department of Electrical and Computer Engineering, Princeton University, Princeton, NJ 08540 USA, and also with the NEC Laboratories America Inc, Princeton University, Princeton, NJ 08540 USA (e-mail: tlima@princeton.edu).

Chaoran Huang is with the Department of Electronics Engineering, Chinese University of Hong Kong, Hong Kong, China (e-mail: crhuang@ee.cuhk.edu.hk).

Bhavin J. Shastri is with the Department of Physics, Engineering Physics and Astronomy Queen's University & Vector Institute, Kingston, ON K7L 3N6, Canada (e-mail: bhavin.shastri@queensu.ca).

David Rosenbluth is with the Lockheed AI Center, Shelton, CT 06484 USA (e-mail: rosenbluthd@gmail.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JSTQE.2022.3195824.

Digital Object Identifier 10.1109/JSTQE.2022.3195824

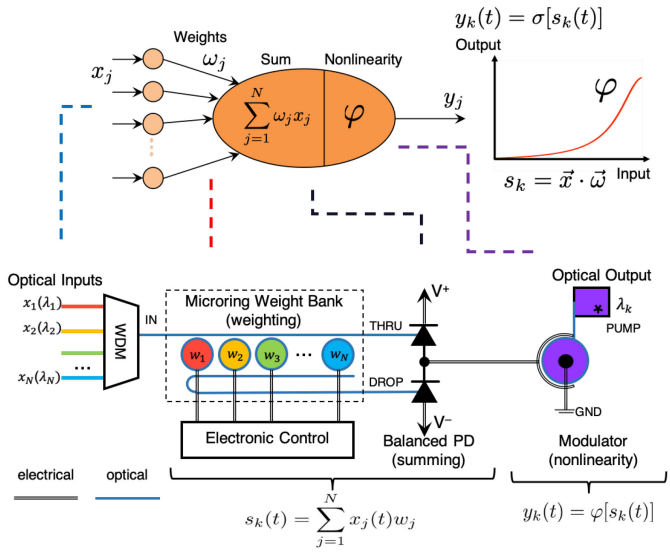


Fig. 1. Mathematical abstraction of neuronal function and silicon photonics implementation.

4) We analyze the latency performance of PNN-assisted hardware for both RF fingerprinting and fiber nonlinearity compensation.

The rest of this article is structured as follows: Section I-A discusses potential advantages of photonic AI hardware for communications. Section I-B presents related work. Background on silicon photonic neural networks is presented in Section II. In Section III, we show our novel photonics-inspired compact network for real-time RF fingerprinting. In Section IV, we review our work on experimental demonstration of an integrated silicon photonic neural network for fiber nonlinearity compensation. Section V contains performance analysis of silicon photonics-assisted AI hardware for communications. Section VI concludes the article.

A. Potential of AI Systems for Communications Using Photonics

In the recent decades, the world has experienced exponential growth in Internet traffic. To process the enormous amount of data, digital signal processing (DSP) techniques have been widely deployed. However, due to Moore's law, hardware scaling of application-specific integrated circuit (ASIC) based DSP chip will struggle to keep up with the exponentially increasing data traffic in the future. As a result, there has been effort on developing more compact processing blocks in communication systems using deep neural networks [5], [6], and some works, on the other hand, investigate novel hardware platforms to more efficiently process the data [7], [8].

For example. in high-speed digital coherent optical fiber communications systems, Zhang et al. [5] proposed using an artificial neural networks (ANN) with only two fully-connected layers to compensate nonlinear and linear distortion of received signals. On the other hand, in wireless communications, ANNs have found broad applications toward processing the received wireless signals to improve link performance [9], situation awareness [10], and security [1], [3]. Although deep learning algorithms have been shown to outperform traditional DSP algorithms in performing many functions in communication systems, the real-time implementation of them on ASIC chips remains a challenge due to the high throughput, low energy, and low latency requirement for communication systems.

The emerging field of neuromorphic photonics provides a promising platform to address these challenges [11]. The analog nature of neuromorphic photonic hardware allows it to emulate neural networks efficiently, reducing time and energy consumption and breaking the fundamental bandwidth/interconnectivity trade-offs within electronic devices by multiplexing wavelengths on the same waveguide. Providing advantages in throughput, latency, and power consumption, neuromorphic photonics can perform machine learning algorithms for communication systems that are unreachable by conventional digital electronic platforms [12]–[14].

B. Related Work

Machine learning implementations on photonic platforms have recently demonstrated their applications toward communications with system benefits such as performance, throughput, power consumption and latency. Based on the different photonic architectural designs and their training and applications, there have been several approaches, including neuromporphic photonics, optical reservoir computing and photonics-based adaptive online learning.

1) Neuromorphic Photonics: This approach aims to develop neural networks on photonic platforms to enable ultra-fast deep learning inference. The designed photonic neurons emulate biological neurons with all the essential functionalities, and the existing neural network algorithms can be leveraged to train the photonic neural network and map the training results to photonic hardware. We recently implemented a feed-forward neural network model over a silicon photonic chip and demonstrated optical fiber nonlinearity compensation in a digital coherent optical communications after transoceanic distance 10,080 km [2], [4]. We will further review more details of this work in the Section IV, with emphasis on the description of the silicon photonic chip hardware implementation in a digital coherent optical communication system.

2) Optical Reservoir Computing (RC): RC is a class of recurrent neural network (RNN) consisting of three parts, input layer, a randomly connected recurrent neural network, and an output layer. Reservoir computing system are designed to provide sufficiently complex dynamics to capture a large amount of features from the input data. Thanks to rich dynamics from recurrent neural networks, they only require to train the output of reservoir, and all the weights and biases of recurrent neurons remains untrained [15], [16]. This framework enables a more time-efficient training process. Recently, implementation of reservoir computing systems on optoelectronic hardware has been growing rapidly because of the advantages in rich dynamics [17], capability to perform high-speed data processing [18] and large hardware reduction [19]. These photonics-based RCs

provides a handful of applications in communication systems. For example, Antonik et al. [7] demonstrated real-time channel equalization with an opto-electrical reservoir and an FPGA for both generating the input symbols and training the readout weights. Argyris et al. [20] constructed a time-delayed feedback RC for signal recovery in optical communications. Most of these RCs are built from discrete optical elements which are not compact and limit the bandwidth of the systems. To build a compact and high-bandwidth RC for communication systems, in the recent years, more researches have focused on integrated photonic platforms [21], [22].

3) Photonic Adaptive Online Learning: In optical communication and wireless communication systems, there are scenarios in which multiple channels overlapping with each other and need to be separated adaptively. Silicon photonic circuits have been developed to separate channels which have strong mixing [23]. Employing photonic microring weight bank [24] with adaptive online learning, Ma et al. demonstrated blind source separation by de-mixing signals in optical domain with multiple wavelengths [25], [26]. In Section II, we will further explain the principle and design of the microring weight bank that can offer adequate control for the implementation of adaptive online learning algorithms for dynamic situations.

II. Background of Photonic Neural Network

Photonic neural networks have been demonstrated in different platforms, including free-space holographic neural network [27], [28], optical fiber based neural network [29], [30], diffractive optics [31]–[33], and integrated photonic circuits [12], [34]–[36]. In this work, we focus on the platform, WDM-compatible integrated silicon photonic neural networks.

1) Architecture of Integrated Silicon Photonic Neural Networks: The neural information processing of each neuron can be decomposed into four parts: (1) input signals reception, (2) independent weighting of inputs, (3) summation, and (4) nonlinear transformation. These operations can be mapped and implemented on a silicon photonic integrated circuit as shown in Fig. 1. The input signals are encoded in the power of the optical sources. Each optical input has a unique wavelength to represent the signal from a different pre-synaptic neuron. These signals are multiplexed through wavelength-division multiplexing (WDM) to an optical fiber, which is then coupled to an on-chip waveguide. The weighting is implemented using a set of silicon micro-ring resonators (MRR) [24], [37]. Each of the micro-ring resonators is designed to resonate with each input wavelength and control its weight independently by fine tuning the resonance to change the transmission. The weighted optical signals are summed by balanced photodetectors (BPDs), which linearly transform the sum of the optical signals to *photocurrent*. The photocurrent is then sent to a silicon micro-ring modulator (MRM) [34] as input to a neuron. It is worth noting that the weights are in the range of [-1,1], so usually an electrical amplifier such as a transimpedance amplifier (TIA) will be added to scale up the weighted value. Due to the carrier-induced nonlinear effect, a MRM will provide the nonlinear activation to the optical pump signal. This mechanism shows the working principle of a neuron node on a silicon photonic circuit. The optical output of the MRM can be further sent to other photonic neuron nodes to form a network system.

2) Microring Weight Banks: One important component of PNNs are MRR weight banks, which play the main role of configuring the weights of PNNs. The microring weight bank is implemented with in-ring N-doped photoconductive heaters [38] in the recent works [37], [39], [40]. Tait et al. [39] developed a feedback control mechanism to thermally tune the microring by adjusting the electrical current applied to the N-doped heaters. This technique has been demonstrated to perform continuous, multi-channel control with accuracy over 8 bits [40], [41], which is comparable to the resolution of matrix multipliers used in DSP ASICs. In the design of microring weight bank, there are two complementary outputs from the THRU and DROP ports of an MRR weight bank, which are detected by a balanced germanium-on-silicon photodetector [42] to achieve simultaneous summation of neural inputs.

3) Microring Modulators: The microring modulator, also called "photonic neuron," performs the nonlinear activation function. A photonic neuron will receive a combined photocurrents generated by the attached balanced photodetector, which results in the modulation of an MRM's transmission via free-carrier injection to the p-n junction. Thereby, if an extra optical source is provided and sent the the input port of a MRM, the optical power will be modulated nonlinearly due to the electrical-to-optical transfer functions. This nonlinear transfer function has been experimentally demonstrated to be programmable with different bias currents to the p-n junction of a MRM [34]. With the above design principle of fully-integrated silicon photonic neural networks, we will further introduce how its model and the actual hardware can be applied to AI-assisted communication systems in the following sections.

III. A Photonics-Inspired Compact Network for RF Fingerprinting

In this section, we take RF fingerprinting as an example to demonstrate the ability of a photonics-inspired neural network model to reduce the complexity of the conventional NN model [1], and further be emulated using a small scale FPGA for real-time AI processing in wireless communication systems. This model can supposedly be transferred to photonic hardware, but we leave this investigation as the future work. In the scope of this article, we propose the model and discuss its performance on FPGA emulation.

A. Background of RF Fingerprinting

Internet of things (IoT) devices without consistent human possession can be vulnerable to authentication-related threats, such as spoofing attacks in which digital IDs are cloned. *RF fingerprinting* is one method to prevent such attacks, involving the identification of specific devices by their unique emission characteristics originating from manufacturing imperfections. The variation in behavior among devices is statistical in nature and imparts a device-specific signature to the transmitted signals, allowing for RF identification. However, the device signature

is usually hidden by the much stronger signal carrying the information content of the transmission. Extracting the sparsely distributed and low signal-to-noise ratio (SNR) features and achieving high classification accuracy are the main challenges of RF fingerprinting.

Current state-of-the-art RF fingerprinting techniques require pre-existing knowledge of RF engineering [43] or large deep neural networks [1], [44], [45] and process radio signals offline. Although these approaches have achieved noteworthy results in demonstrating the effectiveness of ML for RF signal processing and classification, they face significant challenges during implementation in real-time at the network edge and are not scalable due to latency introduced by the digital electronic hardware and the large size of the network architectures.

B. Problem Formulation

We define our RF fingerprinting problem as the classification of demodulated and digitized I/Q signals from 30 identical Digi XBP24CZ7SITB003 ZigBee Pro devices with a carrier frequency of 2.405 GHz. Each transmission has a 32-byte random payload and a 2-byte checksum, and the data is provided by the Naval Research Laboratory (NRL) [1]. The transmissions contain features such as carrier frequency and phase offsets that can be used for trivial classification but are susceptible to an adversary's spoofing of the local oscillator of a transmitter. Therefore, to prevent this type of attack and further enhance the security of RF fingerprinting, we pre-process the raw demodulated I/Q samples from the transmissions to generate *residual data*, which is defined as follows:

$$Z = R_r - f_{\omega,\phi}(R_{gr}), \tag{1}$$

where Z is the residual data, R_r is the received raw data, R_{gr} is the ground truth data, ω and ϕ are the carrier frequency and phase offset of the transmission, respectively, $f_{\omega,\phi}$ is a function that applies the frequency and phase offset ω, ϕ to the ground truth data, and the ground truth data is the signal that we regenerate after decoding the transmission. To derive ω, ϕ and residual data, we follow the ZigBee data recovery steps proposed by [1]. The details of residual processing are provided in Appendix A. After residual data processing, the trivial signatures (carrier frequency and phase offset) are removed from the received raw data. Furthermore, by subtracting the ground truth signal, the model can detect the device-specific features that are usually hidden within the received raw signal. The final dataset is the residual data consisting of 34 bytes per transmission and about 1,120 transmissions for each device. We split 80% of the transmissions for the training dataset, 10% for the development dataset, and 10% for the test dataset.

The original residual data has 2 channels (I/Q) and a total of 34 bytes (17,408 samples). In the same manner as Ref. [1], we define a two-byte segment of the transmission as the data unit for the classifier. As such, each transmission has 17 data units, where each unit is a two-channel time series with 1,024 steps. We denote the n-th data unit of a transmission as $Z_n = [Z_n^0, \ldots, Z_n^i, \ldots, Z_n^{1023}], \forall n \in 0, 1, \ldots, 16$. Here, at each time step, $Z_n^i = [I_n^i, Q_n^i]^T$ is the i-th sample in the data unit. We treat the consecutive 32 samples of each channel as features and reshape the input from (2,1024) to (64,32). The reshaped data unit can be represented as

$$\begin{aligned} X_n &= \left[X_n^0, \ldots, X_n^i, \ldots, X_n^{31}\right], \\ X_n^i &= \left[I_n^{32i}, I_n^{32i+1}, \ldots I_n^{32i+31}, Q_n^{32i}, Q_n^{32i+1}, Q_n^{32i+31}\right]^T, \\ &\quad \forall i \in 0, 1, \ldots, 31 \end{aligned} \tag{2}$$

The model will take each segment of data Y_n at once and output a log probability $\log \vec{P}_n = [\log P_n^1, \ldots, \log P_n^k, \ldots, \log P_n^{30}], \forall k \in 1, 2, \ldots, 30,$ where $\log P_n^k = \log P(C_k|X_n)$, and C_k is the k-th transmitter. The overall classification result is determined by the accumulated log probability of N data units of each transmission, i.e. $\log \vec{P} = \sum_{n=1}^{N} \log \vec{P}_n$. Here, N can be chosen from 1,2...,17.

C. AI-Enhanced Algorithm

In this work, we propose a compact neural network model to classify on the steady state of the transmission and perform a benchmark comparison to the multi-layered CNN model proposed by Ref. [1], with which we share the same source data. For the rest of this manuscript, we use *"NRL CNN"* to refer to the baseline model proposed by the Naval Research Laboratory [1], and call our proposed model the *"PRNN-CNN"*.

The proposed PRNN model is inspired by silicon photonic neural networks [34], [46]. Recently silicon photonics-based integrated circuits have found important applications in machine learning due to their high bandwidth, low latency, low power consumption, and parallel processing capability [14]. Photonic RNNs, implemented using a broadcast-and-weight system configured by microring weight banks, have been shown to be isomorphic to continuous-time RNN (CTRNN) models with application to ultra-fast information processing for control and scientific computing [46].

Here, we propose using the photonic hardware compatible RNN model [14], [46] to extract the temporal features in a transmission and emulating this model on an FPGA to achieve real-time processing. In the future, this RNN model can be further realized on a fully-integrated photonic neural network to improve the latency of the RF fingerprinting system.

The dynamics of a photonic recurrent neural network can be described by the following equation [46]:

$$\frac{d\vec{s}}{dt} = \frac{-\vec{s}}{\tau} + \mathbf{W}\vec{y}(t) \tag{3}$$

$$\vec{y}(t) = \sigma(\vec{s}(t)) \tag{4}$$

where $\vec{s}$ is the neuron's state, $\vec{y}$ is output signal, τ is the time constant of the photonic circuit, $\mathbf{W}$ is the photonic weight, and $\sigma(\cdot)$ is the transfer function of the silicon photonic modulator neurons [4], [34]. It is worth noting that the nonlinear transfer function can be expressed as Lorentzian function, $\sigma(x) = x^2/(x^2 + (ax + b)^2)$, where a, b are constants. More generally, we can add an external input signal $\vec{x}(t)$ to the photonic RNN, and the dynamical equation become:

$$\tau\frac{d\vec{s}}{dt} = -\vec{s} + f(\vec{s}, \vec{x}, \vec{\theta})$$

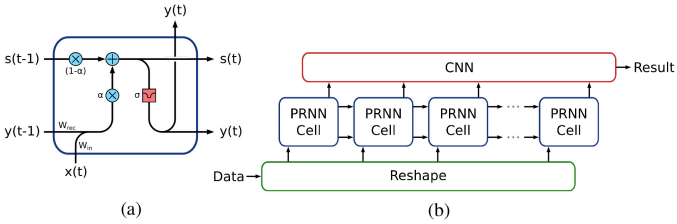


Fig. 2. Framework of the discrete version of the photonic RNN model. (a) Schematic diagram of the discrete photonic RNN cell. Here $\alpha = \Delta t/\tau$ (b) An example of connecting a photonic RNN to a CNN classifier for a time series classification task.

$$f(\vec{s}, \vec{x}, \vec{\theta}) = \mathbf{W}_{in}\vec{x}(t) + \mathbf{W}_{rec}\sigma(\vec{s}(t)) + \vec{b}, \tag{5}$$

where θ represents the trainable parameters for the network such as the bias $\vec{b}$, the recurrent photonic weight $\mathbf{W}_{rec}$, and the input weight $\mathbf{W}_{in}$. In this work, we approximate the dynamical equation in (5) using the *forward-Euler method* and use this discrete version of dynamics to construct the photonic RNN (PRNN) cell shown in Fig. 2(a) and implement it on an FPGA. The PRNN dynamics can be expressed by the following formula:

$$\begin{aligned}
\tau\frac{d\vec{s}}{dt} &= -\vec{s} + f(\vec{s}, \vec{x}, \vec{\theta}) \Rightarrow \vec{s}(t + \Delta t) \\
&= \vec{s}(t) + \Delta\vec{s} = \vec{s}(t) + \frac{\Delta t}{\tau}(-\vec{s}(t) + f(\vec{s}, \vec{x}, \vec{\theta})), \\
&= \left(1 - \frac{\Delta t}{\tau}\right)\vec{s}(t) + \frac{\Delta t}{\tau}f(\vec{s}(t), \vec{x}(t), \vec{\theta}) \\
&= (1 - \alpha)\vec{s}(t) + \alpha f(\vec{s}(t), \vec{x}(t), \vec{\theta})
\end{aligned} \tag{6}$$

where $\alpha = \Delta t/\tau$. The state variable $\vec{s}$ is updated based on (6) and can be constructed using Pytorch nn.module [47], which is available on GitHub.[1] For the rest of the paper, we set $\Delta t/\tau = 0.5$, and use an experimentally measured transfer function of a photonic neuron, $\sigma(x) = x^2/(x^2 + (0.3 + 0.25\,x)^2)$. We propose an RF fingerprinting system which consists of a PRNN layer with PRNN unit cells and a CNN with two Conv1D layers and one fully connected layer. It is shown in Fig. 2(b). The input data has 64 channels with length 32 as described in Section III-B, and the PRNN has 16 neurons to generate 16 channels of output sequence. The output $\vec{y}(t)$ from the PRNN is then sent to the convolutional layers. We flatten the output of the convolutional layers and connect it to the fully-connected layer with 30 neurons and a log softmax activation function. The details of the parameters of the model are given in Appendix B, and the overall procedure to demonstrate RF fingerprinting using the PRNN-CNN model is shown in Fig. 3. This architecture has 6,302 parameters in total, which is 50 times fewer than the number of parameters used in the model proposed in Ref. [1]. With the update equation provided in (6), this PRNN can be trained end-to-end with the CNN model and with a back propagation through time (BPTT) algorithm [48]. In this work, the training objective is the negative log likelihood loss given by

[1][Online]. Availabe: https://github.com/Hsuan-Tung/PCICN_RFFingerprinting

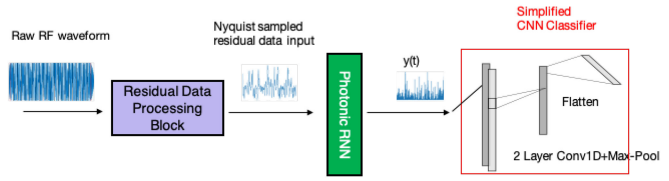


Fig. 3. Procedure of the proposed RF fingerprinting system.

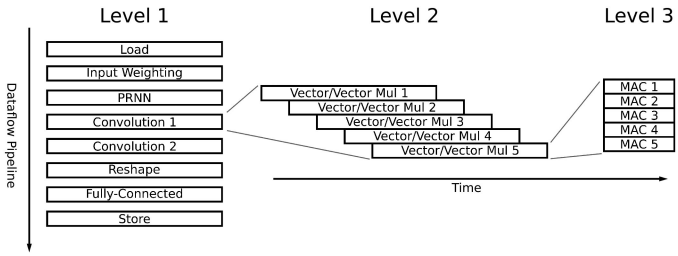


Fig. 4. Design hierarchy for the FPGA implementation of the PRNN-CNN classifier.

the output of the model and targeted label. Glorot uniform initialization [49] was used for initialization of all convolutional and dense layers, and Kaiming initialization [50] was used to initialize the parameters of PRNN layer. We use an ADAM optimizer [51] with an initial learning rate of 0.001. For each epoch, we train the model with a training dataset consisting of randomly shuffled data units and set the batch size to be 1,700. We validate the model by measuring its accuracy on the development dataset using $N = 17$ segments of each transmission. When the classification accuracy on the development dataset doesn't increase in 10 consecutive epochs, we decrease learning rate by 50% and keep training until 100 epochs.

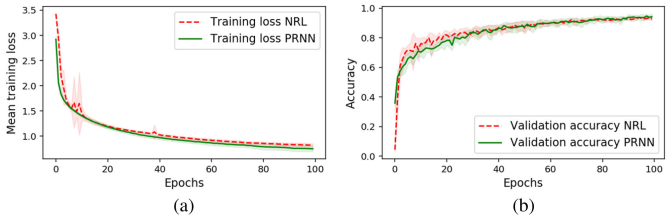
D. Experiments and Results

In this work, we performed two sets of experiments. Firstly, we implemented the baseline NRL model and our own using a NVIDIA GeForce GTX 1080 Ti GPU for training, validation, and testing. This experiment was implemented in Python using Pytorch 1.4.0 [47] as the back end and focused on comparing the classification performance of the two models. The results are shown in Section III-D.

In the second set of experiments, we sent the PRNN-CNN model with the trained parameters selected from Section III-D to an FPGA to demonstrate RF fingerprinting in real-time.

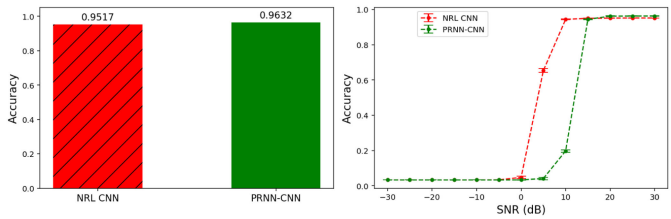
The PRNN-CNN classifier was implemented on the low-cost PYNQ-Z1 FPGA board, chosen to demonstrate the hardware and cost minimization opportunities offered by the compact model. The FPGA design was created using the High-Level Synthesis (HLS) feature of the Vivado Design Suit [52], which allows a simplified C++ model of the design to be compiled first into a hardware description language format and then into a bitstream file for the FPGA.

The FPGA design is organized in the three-level hierarchy shown in Fig. 4. The first level consists of a series of stages within an HLS Dataflow pipeline. Each stage corresponds to a layer of



(a) (b)

Fig. 5. (a) Training loss. (b) Accuracy on the development dataset. The training and validation process is repeated 10 times, and the standard deviation is shown as the filled region. Red: Baseline NRL CNN; Green: Proposed PRNN-CNN model.



(a) (b)

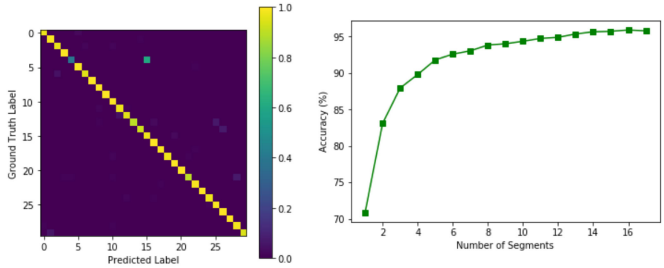
Fig. 6. Comparison of NRL's classifier [1] and our PRNN-CNN classifier based on (a) maximum accuracy and (b) noise performance.

the classifier, and a series of independent classifications pass through this pipeline, with an initiation interval corresponding to the latency of the slowest stage.

At the next level of abstraction, the operations within each stage are organized as a conventional HLS Pipeline. The execution path of each pipeline typically corresponds to a single vector-vector multiplication. A layer may be divided into a series of operationally identical such vector multiplications that progress through the pipelined execution path with a typical initiation interval of one cycle. (Read-after-write dependencies within the PRNN pipeline restrict the initiation interval to 16 cycles).

At the final level, the execution path of each layer may be divided at points into parallel paths executed simultaneously. These parallel paths, consisting of individual multiply-accumulate (MAC) operations, allow for better utilization of the available digital signal processing resources and better minimization of overall latency. The training and evaluation results can be visualized in Fig. 5. We trained the PRNN-CNN model and compared it with the baseline NRL CNN model. To ensure the consistency, both models were trained 10 times. The standard deviation of the training loss and validation accuracy are shown as the filled region in Fig. 5. When using the baseline NRL CNN architecture, we achieve 95.17% accuracy. The network has three convolutional layers and three fully connected layers with a total of 322,602 trainable parameters as detailed in Appendix B. We verified that our model converges to similar mean training loss and validation accuracy as the NRL CNN even with 50 times fewer parameters. For both the baseline and our proposed model, we selected the trained parameters with the best accuracy on the development dataset and evaluated the models' performance with these parameters by measuring their classification accuracy on the test dataset. Both models achieve over 95% classification accuracy on the test dataset as shown in Fig. 6(a).

To further analyze the performance of the trained fingerprint classifier under less than ideal conditions, we added noise to both the I and Q channels of the test dataset to create variations in the input data. For simplicity, a standard artificial white Gaussian noise (AWGN) channel was implemented and simulated. In each run of the experiment, we added AWGN with a specific SNR in the set of $\{-30, -25, -20, \ldots, 25, 30\}$(dB), and repeated for 20 times to check consistency of classification results. As shown in Fig. 6(b), for the PRNN-CNN model, the classification accuracy stays above 95% when the SNR is at least 15 dB while



(a) (b)

Fig. 7. Performance of classification on 30 devices. (a) Confusion matrix of classification results. In the confusion matrix, component M_{ij} represents the ratio between the output prediction and ground truth label, i.e. $M_{ij} = N_i^{pre/} N_j^{tar}$, where N_i^{pre} is the number of predictions for the i-th device, and N_j^{tar} is the number of transmissions for the j-th device. The color yellow represents the perfect ratio, and purple represents the zero ratio. (b) Impact of using different numbers of segments for classification.

degrading when the SNR is below 15 dB. On the other hand, the NRL CNN model is more robust to the noise, as the accuracy starts decreasing when the SNR is below 10 dB. To improve the robustness to noise, Ref. [53] suggests using a liquid time constant RNN model to adjust the decay time of each neuron based on input perturbation. In this work, we mainly focus on constructing a compact network that can be implemented on an FPGA to provide low-latency and high-bandwidth processing, and we will leave improvement in noise performance to our future work.

After training and constructing the PRNN-CNN with trained parameters on the PYNQ-Z1, we implemented RF fingerprinting on the test dataset. The results when using the PRNN-CNN model are shown as a confusion matrix in Fig. 7(a). Our model reaches 95.90% accuracy. There was only a 0.42% decrease for the FPGA implementation due to slight additional sources of imprecision.

During testing there is a trade-off between accuracy and throughput. Maximum accuracy requires that all 17 segments of a transmission be used, but using only 8 segments still allows for a high level of accuracy, nearly 94%, while more than doubling the throughput due to the reduction in calculation per classification required. The precise relationship between number of segments and accuracy is shown in Fig. 7(b).

With the above achievement, we demonstrate that a PRNN model can be applied to RF fingerprinting and be emulated

by a small-scale FPGA processor to real-time classification.[2] Although the model is currently emulated on FPGA, it can be potentially implemented on integrated photonic platforms to further reduce the latency. As a result, the hybrid photonic/FPGA edge AI processor could be a promising platform to investigate in the future.

IV. Fiber Nonlinearity Compensation With PNN

In this section, we would take a step further to show the experimental implementation of a fully-integrated silicon photonic neural network for communication systems. We target to fiber nonlinear dispersion compensation as an example since a model with compatible neural network size for current scale of photonic chip has been proposed [5]. Here, we adapt the model based on physical characteristics of PNN, and implement it on our photonic chip to demonstrate its validity in terms of signal quality improvement and latency performance.

A. Background of Fiber Nonlinearity Compensation

In fiber communication systems, the optical signal is impaired mainly due to the linear dispersion and the nonlinear Kerr effect, which can be described by the following Nonlinear Schrödinger Equation:

$$\frac{\partial u_{x/y}(t,z)}{\partial z} + i\frac{\beta_2}{2}\frac{\partial^2 u_{x/y}(t,z)}{\partial^2 t} = i\frac{8}{9}\gamma[\mid u_{x/y}(t,z)\mid^2 + \mid u_{y/x}(t,z)\mid^2]\mid u_{y/x}(t,z), \quad (7)$$

where $u_{x/y}(t,z)$ is the optical field at the x and y polarizations, respectively, β_2 is the group velocity dispersion governing the linear impairment, and γ is the Kerr nonlinear coefficient governing the nonlinear impairment in the fiber transmission systems. The state-of-the-art DSP can recover the linear impairments, but the nonlinear effects are difficult to overcome because they requires the implementation of computationally intensive nonlinear compensation algorithms using application specific integrated circuits (ASICs). As a result, even though recent demonstrations have shown transmission capacities approaching the Shannon limit in the linear regime of optical fibers, the nonlinear impairment remains the major bottleneck in long-distance fiber transmission systems [54].

Zhang et al. [5] proposed using a fully-connected ANN to compensate fiber nonlinearity impairment. This approach achieves comparable signal quality improvement while reducing computational complexity. However, such an ANN is implemented on digital electronic hardware and cannot be processed in real-time for optical communication systems, for which the signal bandwidth is tens of GHz and requires real-time processing. Photonic neural networks (PNNs) supporting high bandwidth data processing pave the way to realizing real-time fiber nonlinearity compensation. In the following subsections, we will introduce the problem setting, algorithm, experimental method and implementation results of fiber NLC with an integrated silicon PNN.

[2]Our real-time demo can be found at https://www.youtube.com/watch?v=CIfddiscE3k.

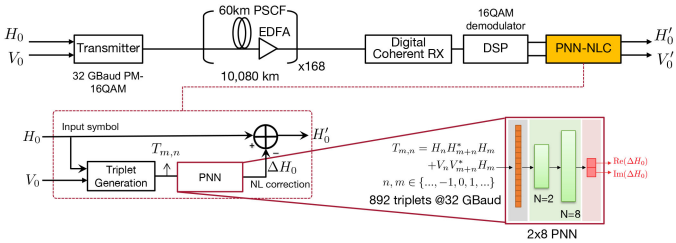


Fig. 8. The PNN-assisted fiber-optic transmission system of a 10,080 km trans-pacific uncompensated transmission link carrying a single-channel 32 Gbaud polarization-multiplexed (PM)-16QAM signal. PSCF: pure silica core fiber; EDFA: erbium-doped fiber amplifier; Rx: receiver. PNN-NLC: photonic neural network-based nonlinearity compensation.

B. Problem Formulation

In this work, we explore a long distance fiber communication link with a transmission distance of 10,080 km and carrying a single channel 32 Gbaud/s dual-polarization (DP) 16 quadrature amplitude modulation (QAM) signal.

The nonlinearity impairment in the system can be considered the perturbation due to the nonlinear effects, *i.e.*, $u_{x/y}(t,z) = u_{0,x/y}(t,z) + \Delta u_{x/y}(t,z)$, where $u_{0,x/y}(t,z)$ is the solution of linear propagation and $\Delta u_{x/y}(t,z)$ is the nonlinear perturbation.

The nonlinear perturbation can be approximated by a combination of triplets T [55]:

$$\Delta u_{x/y}(0,z) = \sum P_0^{3/2}(H_n H_{m+n}^* H_m + V_n V_{m+n}^* H_m)C_{m,n} \quad (8)$$

where P_0 is the launch power, $C_{m,n}$ are the nonlinear perturbation coefficients, H and V are the received symbol sequences at the x and y polarizations respectively, and m and n are symbol indices with respect to the symbols of interest H_0 and V_0. The triplet is defined as $T = H_n H_{m+n}^* H_m + V_n V_{m+n}^* H_m$. Triplets represent the intra-polarization and inter-polarization nonlinear interactions corresponding to the nonlinear terms on the right hand side of (7).

Our goal is to improve the received signal quality by building a small neural network model to predict the nonlinear impairment compensation term, which will be subtracted from the received symbols. This NN model can be further implemented on an integrated silicon photonic circuit for high throughput data processing. The proposed PNN-assisted NLC system is shown in Fig. 8. Here, we only consider compensating nonlinear impairments in x-polarization as an example, but the same technique can be applied and transferred to y-polarization.

C. AI-Enhanced Algorithm

Instead of using conventional digital backpropagation method [56], [57] to solve nonlinear Schrödinger equation, we aim to construct an ANN to learn the nonlinear perturbation $\Delta u_{x/y}(t,z)$ with the abundance of transmission data, and compensate the nonlinear impairment by subtracting the predicted perturbation from the received optical field $u_{x/y}(t,z)$. Here, the

Algorithm 1: Training ANN for NLC.

Setting:
Initial data: transmitted symbols $A_{x,k}$; received symbols H_k
Input: $\hat{X}$ = triplets data set $H_n H^*_{m+n} H_m + V_n V^*_{m+n} H_m$
Neural Network Model: $\mathcal{F}_\theta(\cdot)$
Optimizer: $optim$, Adam optimizer with learning rate lr
Output: $\hat{H}_{k,NL} = [\hat{H}^{re}_{k,NL}, \hat{H}^{im}_{k,NL}]$
function Train($\mathcal{F}_\theta(\cdot)$, $\hat{X}_{train}$, $optim$)
  for x_k in $\hat{X}$ **do**:
    Compute neural network output: $\hat{H}_{k,NL} = \mathcal{F}_\theta(x_k)$
    Compute recovered symbol: $\hat{H}_k = H_k - \hat{H}_{k,NL}$
    Compute error: $L = \text{MSE}(A_{x,k}, \hat{H}_k)$
    Backpropagation for GD: L.backward()
    Update Weights and Biases: $optim$.step()
  end for
  return $\mathcal{F}_{\hat{\theta}}(\cdot)$
end Function

proposed ANN model is a fully-connected feed-forward neural network consisting of an input layer taking 892 input triplets, two hidden layers with two and eight neurons respectively, and an output layer producing two outputs representing the nonlinear compensation term in both real and imaginary parts. The inputs of the ANN are the triplets T at x-polarization [5]. The neural network model can be expressed by a function:

$$\mathcal{F}_\theta(\cdot) : \mathbb{R}^{892} \rightarrow \mathbb{R}^2, \tag{9}$$

which takes 892 triplets as input and output a two dimensional tensor to predict the real and imaginary nonlinear compensation terms respectively, and θ is the set of parameters of the neural network model including weights and biases.

The ANN is trained with 32,106 training symbols in a single channel 32 Gbaud PM-16-QAM signal. One thing worth noting is that the weights are confined in the range of -1 to 1 to match the physical constraints of MRR weight bank. The activation function of the neurons in two hidden layers is a Lorentzian function, $\sigma(\cdot)$, characterized by experimental measurement of a silicon MRM neuron, which is shown in Fig. 11(b) and can be numerically modelled as $\sigma(x) = -0.28/[(x-0.5)^2 + 0.49^2] + 1.35$. The outputs of the ANN are the real and imaginary components of the estimated nonlinearity $\hat{H}_{k,NL}$ at the k-th symbol. The recovered symbol $\hat{H}_k$ is obtained by subtracting $\hat{H}_{k,NL}$ from the received symbol H_k, i.e. $\hat{H}_k = H_k - \hat{H}_{k,NL}$. The ANN is trained to minimize the mean squared error (MSE) between the $\hat{H}_k$ and the transmitted symbol $A_{x,k}$. The details of training procedure is given in Algorithm 1, and we validate our model by evaluating the Q-factor of the signal from a completely independent data set (cross-validation data set), which contains 32,106 symbols. Once the convergence of the Q-factor of the cross-validation data (CV data) is achieved, we can load the parameters of ANN to PNN and send a test set to the trained PNN with the on-chip Lorentzian activation function. The strength of nonlinear distortion calculated by the ANN is weighted by the difference between the input power of the training set and the test set. The architecture and procedure of PNN-NLC can be visualized in Fig. 9(a), and we will show the photonic implementation in the next section.

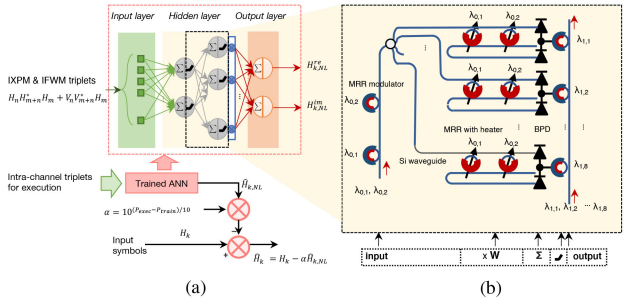


Fig. 9. (a) The architecture of the neural network for fiber nonlinearity compensation (adapted from Reference [5]). (b) PNN implementation of the second hidden layer of the ANN shown in (a) based on weight-and-broadcast architecture [46], [58]. Figure is adapted from Huang et al. [40].

D. Photonic Hardware Implementation

In this section, we show the implementation of NN model with a fully integrated PNN and demonstrate the capability of PNN in executing fast and accurate inference. The designed PNN comprises of two arrays of MRR weight banks connected to two photonic neurons, and each array of MRR consists of four ring resonators, which form a 4×2 PNN. Because of the scale of our PNN, in this experiment, we aim to emulate the computation of the second hidden layer in the ANN model. Despite only a part of the ANN model has been experimentally demonstrated, we have shown the scalability of this PNN platform [34], and provided a thorough discussion fo model in [2]. Also, note that due to the non-optimized gap between MRR modulator and the bus waveguide, the extinction ratio of the transmission is only 2 dB. Thereby, the photonic neurons in this work have to be operated in forward biased mode. With the injected current, the extinction ratio can be increased to 15 dB, however, the signal bandwidth is limited in the range of tens to hundreds of MHz. Although the current experiment cannot process the same data rate of original transmitted signals. The same NN model can be implemented on the same silicon photonics platform except using carrier-depletion-basd MRR modulator for photonic neuron, i.e. MRR modulator is reversely biased, to achieve tens of GHz bandwidth. The details of discussion is in Section V.

In this experiment, since the second hidden layer has 8 neurons, each of which is connected to two neurons in the first hidden layer with a 2×1 weight matrix. Our PNN can only support a 2×2 network at once, thereby, we emulated neural information processing in the second layer by sending the same input four times and reconfigure the 2×2 weights and biases of two neurons to complete the full emulation.

The detailed implementation is shown in Fig. 9(b), where the two neuron outputs in the first layer are encoded to two wavelengths $\lambda_{0,1}$ and $\lambda_{0,2}$. They are multiplexed on a waveguide, and are split equally to the two neurons in the second layer. The MRR weight bank weights the two neuron outputs from the first hidden layer, with in-ring N-doped photoconductive

heaters. The weights are controlled by the tuning transmission of MRR, which can be programmed by sourcing the electrical current to the N-doped heaters [39]. The N-doped heaters enable continuous, multi-channel control of the MRR weight bank with accuracy and precision over 8 bits [40], [41]. The MRR weight bank has two complementary optical output, each of which is detected by a germanium-on-silicon photodetector. The two photodetectors form a balanced photodetector [42], in which the output photocurrent represents the subtraction operation between the two outputs of the MRR weight bank, resulting in -1 to $+1$ continuous weight range. The combined photocurrent modulates the transmission of the MRR modulator (*i.e.* photonic neuron) via free-carrier injection to a p-n junction, and ultimately modulates the optical power of a continuous-wave laser signal (labeled as "neuron pump"). The on-chip inductor and capacitor provide a network matching circuit for efficient optical-electrical-optical (OEO) conversion. The MRR modulator exhibits nonlinear electrical-to-optical transfer functions that yield the activation function in the neural network. Besides, the neuron biases can be configured by changing the biasing current I_b to a modulator. This feed-forward ANN model is implemented using a broadcast-and-weight architecture, which is based on the concept of WDM to support the parallelism. The fabricated photonic NN and the schematic diagram of the photonic circuits are shown in Fig. 10.

For the silicon photonic neural network, the fabrication error and temperature fluctuation could lead to undesired performance. Therefore, when we configure the photonic neural networks, we will first run a calibration procedure, which is discussed in details in Ref. [41], to lock the micro-rings to the desired resonance wavelength. Experimentally, our chip is connected to temperature controller to make sure the temperature is constant to minimize the thermal fluctuation. Once we lock the micro-rings, we can set the target weights with over 8-bit precision and run the neural network properly. In the foreseeable future, one can improve the fabrication and temperature tolerance by using trimming to reduce resonator variation [59], or applying trench isolation for heaters [60], [61]. To improve the tuning efficiency, non-thermal tuning techniques such as using microelectronic mechanical systems (MEMS) [62] and using barium titanate (BTO) on silicon [63], [64] to tune weights by Pockels effect are also potential candidates for robust and power-efficient photonic neural networks in the future.

The E/O response of the MRR modulators is a Lorentzian function. However, due to the power range of the input signal, the measured transfer function is usually only a fraction of the Lorentzian function. To characterize the full activation function, we sweep the neuron input around the MRR modulator resonance, and reconstruct the full activation function (as shown in Fig. 11(b)).

The fitted Lorenztian activation function is applied in the NN model for training (*i.e.* optimizing weights and neuron biases) with the Adam optimizer. The weights are constrained to a range of -1 to 1, according to the operation range of MRR weight bank.

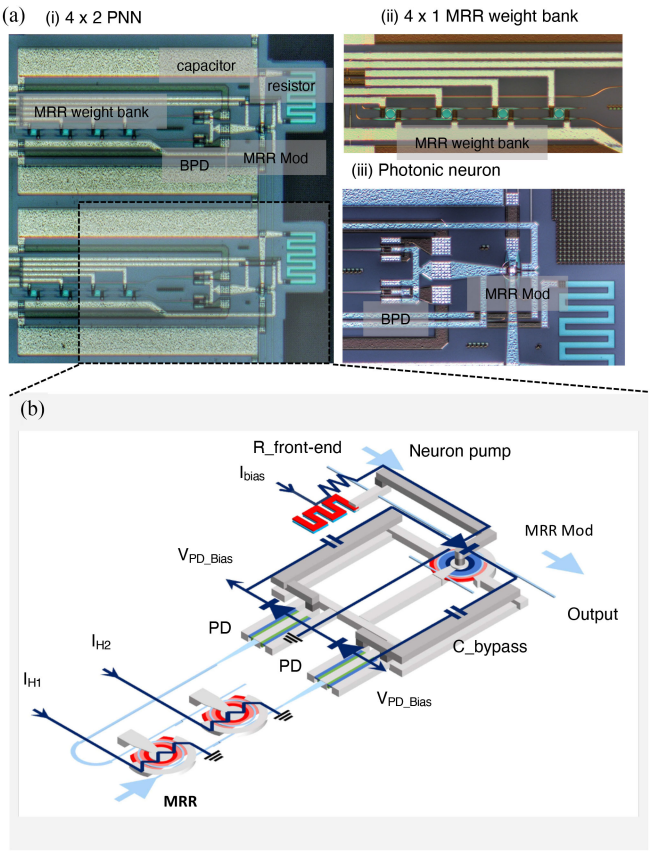


Fig. 10. (a) A micrography of the fabricated photonic neuron. (b) Illustration of a photonic neuron on silicon photonic platform and its equivalent circuit; the capacitor is formed by two layers of metal and has an capacitance of 2 pF; and the planar spiral inductor has a inductance of 4120 pH [65]. This figure is adapted from Huang et al. [2].

E. Results

The procedure to implement NLC is shown as Fig. 11(a). We first use a digital processor, i.e. CPU or GPU, to numerically train the NN model with on-chip activation function provided in 11(b). During the training phase, we validate the NN model to make sure it is successfully trained to recognize the fiber nonlinear distortion by monitoring the Q-factor of a cross-validation (CV) data set as shown in Fig. 11(c). During training, the Q-factor of the CV data increases gradually to 8.1 dB and then converges after 21,600 steps. After the convergence of Q-factor in CV data, we simulate the trained neural network with a 32-bit GPU-assisted workstation and use it as a benchmark to evaluate the implementation accuracy of PNN. The Q-factor obtained from the simulation shows 0.65 dB Q-factor improvement. At this point, we have demonstrated the successful training of the neural network with a data set obtained from a fiber communication system and with a realistic photonic activation function.

Our last step to apply the same model to PNN. As mentioned before, here we only emulate the neural information processing in the second hidden layer. To do so, we encode the first hidden layer outputs, which are calculated from the trained NN model with a GPU, with two external CW lasers that are modulated

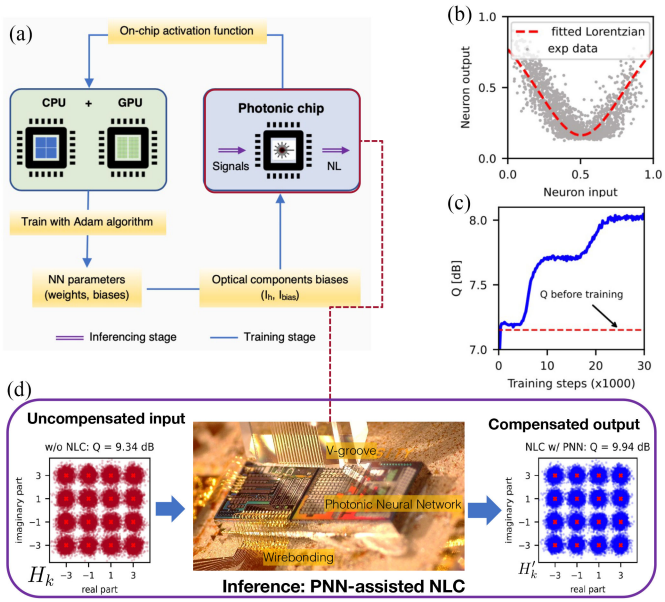


Fig. 11. (a) Execution procedure for the PNN-assisted NLC system (b) The activation function of the photonic neuron modulator (grey dots) and Lorentzian fitting (red dash line) (c) Validation of training process: The Q factor converges as ANN-NLC is trained, CV: cross-validation, *i.e.*, 32,106 symbols generated from the transmission system to monitor the training process; red dash line: Q factor of received signals before nonlinearity compensation. (d) At inference stage, PNN has been applied to uncompensated input signal, and achieved 0.60 dB gain. Figures are adapted from Huang et al. [2], [4].

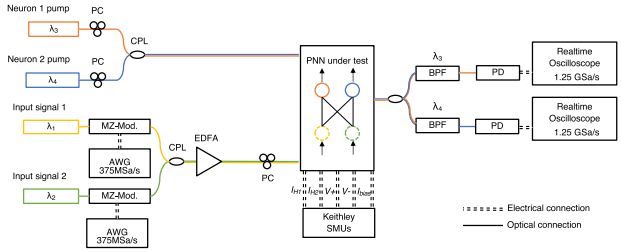


Fig. 12. PNN-NLC experimental setup. AWG: arbitrary waveform generator; CPL: coupler; EDFA: erbium-doped fiber amplifier; MZ-Mod: Mach-Zehnder modulator; PC: polarization controller; PNN: photonic neural network; PD: photodetector; SMU: source meter units. Figure is adapted from [2].

by two channels of signal from a arbitrary waveform generator (AWG). The signal of each channel operates at a rate of 375 MSample/s and generates a waveform of 32,106 test symbols at symbol rate of 46.875 Mbaud/s. This data rate is smaller than the orignal transmitted signal's data rate in order to match the bandwidth limit of our carrier-injection-based MRR modulators. The experimental setup is given in Fig. 12. Based on the weights and biases of the trained NLC model, we program the 2×2 feed-forward PNN four times to implement the second hidden layer of the model, and test the PNN with test symbols obtained from the same transmission system. The highest Q-factor of the signal without nonlinearity compensation is 9.34 dB, which is obtained at the optical power of -1 dBm. The signal's Q factor with PNN is reconstructed from the eight photonic neuron outputs. The constellation of the test symbols measured from

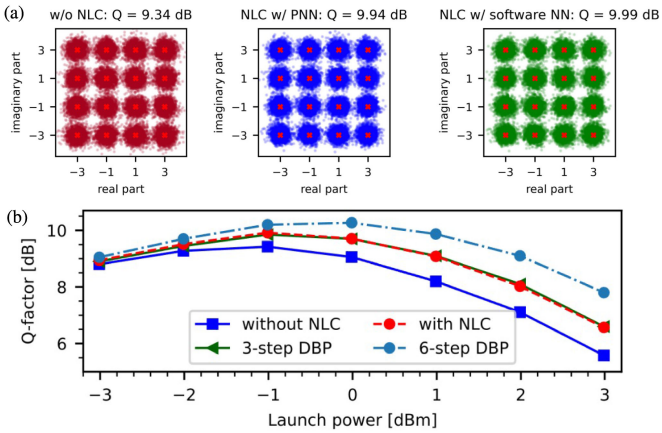


Fig. 13. (a) Constellations of the 16-QAM signal on the X-polarization, with the PNN-NLC gain of 0.60 dB in Q-factor (blue dot) and with the software NN-NLC gain of 0.65 dB in Q-factor (green dot). Red cross: constellation diagram of 16QAM without NLC (b) Performance comparisons of different fiber nonlinear compensation approaches. Blue rectangle: signal Q-factor without NLC; Red circle: PNN-NLC corresponds to the fiber nonlinearity compensation based on the NN model with on-chip activation functions; Green triangle: DBP 3 steps NLC; Cyan circle: DBP 6 steps NLC. DBP 3 steps and 6 steps indicate that fiber nonlinearity is compensated based on the digital backpropagation (DBP) algorithm by equally dividing 10,080 km transmission link into 3 and 6 sections respectively. Figures are adapted from Huang et al. [2].

TABLE I
ESTIMATED NRL AND PRNN-CNN CLASSIFIER PERFORMANCE RESULTS ON PER-CLASSIFICATION BASIS. TIMING RESULTS FOR THE PRNN-CNN CONFIRMED EXPERIMENTALLY

Metric	NRL Model	PRNN-CNN Model	Improvement Factor
Latency (μs)	26,190	219	119
Throughput (1/s)	50	12,192	244

the PNN output is plotted in Fig. 13(a), showing 0.60 dB Q-factor improvement due to fiber nonlinearity compensation. The Q-factor obtained from the software NN implemented on a 32-bits GPU workstation is 0.65 dB Q-factor improvement. The penalty of loading the trained neural network to the PNN is only 0.05 dB in this NLC application, accounting for all the intrinsic noise from the physical components and equipment. We also compare the performance of PNN with conventional computational intensive NLC approaches in Fig. 13(b). As a result, we come to a conclusion that the proposed PNN is comparable to 3-step digital backpropagation (DBP) in terms of Q-factor improvement. In addition, we will show the latency advantages provided by the PNN in section V.

V. Performance Analysis

In this section, we analyze the performance of silicon photonics-assisted AI systems for two communication systems, and compare it to the state-of-the-art digital electronic platforms.

A. RF Fingerprinting

In this subsection, we seek to evaluate the performance of our proposed system using two metrics: throughput per inference and latency per inference. Therefore, we make a comparison between our compact NN model in the hybrid hardware system and

TABLE II
ESTIMATED LATENCY PERFORMANCE ON PER-SYMBOL BASIS. THE LATENCY FOR SiPNN APPROACHES IS BASED ON BANDWIDTH CALCULATION OF PHOTONIC NEURONS. THE IMPROVEMENT FACTOR USES DSP'S LATENCY AS BASE

Metric	DSP	SiPNN-CI	SiPNN-CD
Latency	> 1 µs	13.3 ns	200 ps
Improvement Factor	1	> 75.19	> 5000

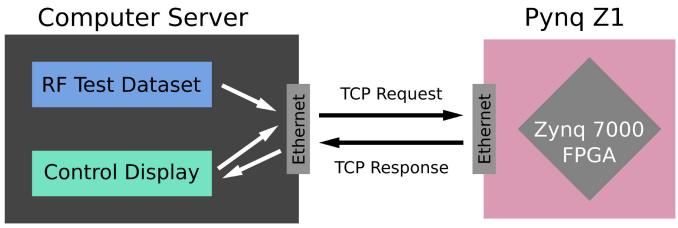


Fig. 14. Diagram of the real-time test setup.

the large CNN model proposed in Ref. [1] on an FPGA.Table II shows the performance estimations for FPGA implementations of the NRL and PRNN-CNN classifiers. Both classifiers are built with the same hierarchical design principles, but the NRL network could not be implemented on the PYNQ-Z1 board due to insufficient available memory to hold the parameters and intermediate data representations. Latency and throughput estimates are derived directly from Vivado HLS compilation reports. In order to confirm the timing behavior of the system, the PRNN-CNN model was implemented on the PYNQ-Z1 and evaluated experimentally in a real-time test. As shown in Fig. 14, randomly selected radio frequency data segments were streamed from a remote computer to the FPGA board, which processed them in real time. The measured timing showed negligible deviation from the estimated values.

The reduction in size and simplification of the PRNN-CNN network with respect to the NRL network allows for a 119-times reduction in latency, and a 244-times increase in throughput based on the estimated parameters. Latency reduction and throughput maximization are key to enabling real-time processing, as both must fall within context-dependent thresholds to prevent data queuing and to enable fast response. Improvement in these metrics allows real-time RF fingerprinting to be expanded to environments necessitating very fast response to large-bandwidth RF data.

B. Fiber Nonlinearity Compensation

Here, we compare different technologies for performing NN model for NLC at the edge, and particularly, we focus on the analysis of latency of the following three technologies: (1) DSP technology, (2) silicon PNN technology with carrier-injection mode MRR modulator neuron (SiPNN-CI), and (3) silicon PNN technology with carrier-depletion mode MRR modulator neuron (SiPNN-CD).

The latency of a NN model is dominated by the time it takes for data routing and executing logic operations, and here we discuss about the latency to process each symbol processed in the NN model. Using DSP technology, the latency is fundamentally bottlenecked by the clock. The data movement and clock distribution along the metal wires also contribute the latency of the DSP. We assume that the DSP circuit has enough number of computing blocks to perform all the MAC operations in one layer in parallel. In this case, the latency is given by the delay of performing 8-bit MAC operation which has a typical number of a few microseconds [66].

For silicon PNNs, the matrix multiplication can be done at a single time step with a few picoseonds. The latency is mainly limited by the bandwidth of the MRR modulator. In this work, as mentioned in Section IV, our experiment used carrier-injection-based MRR modulators to increase the sensitivity of photonic neurons, but the bandwidth of photonic neurons is only ≈ 150 MHz based on the experimental measurement. Therefore, for two layers of NN model, the estimated latency is about 13.3 ns. If the MRR modulator is operated in carrier-depletion mode, then the bandwidth can be increased up to tens GHz [67], [68],. In our current chip layout, Huang et al. have simulated the overall circuits including balanced PDs, the E/O link circuit components such as bypass capacitor and load resistor, and the reversely biased MRR modulator, and shown the bandwidth of the photonic neuron reaches 10 GHz bandwidth, which leads to roughly 200 ps latency for PNN-NLC model.The performance of comparison among these three platforms is summarized in Table, which shows that the silicon PNN chip from this work has at least 75 x improvement in terms of latency compared to DSP approach. With the carrier-depletion mode MRR modulator, the latency can be further improved by two orders of magnitude. Thus, in the above analysis, we show that compared to digital electronic systems, photonic neural networks can provide lower latency and higher throughput to enable real-time processing for tasks in communication systems. On the other hand, power consumption for this system is not the focus of real-time communication systems and beyond the scope of this work. But for those who are interested in this topic, the details of power consumption analysis can be found in the supplementary materials in Ref. [2].

VI. CONCLUSION

In this manuscript, we have discussed the potential of using a microring-based integrated silicon photonic neural network and its model for real-world communication systems. Our model for RF fingeprinrting has achieved the following criteria for a real-time RF fingerprinting system:

- Accuracy improvement: We propose a novel PRNN-CNN model for RF fingerprinting, which achieves over a 96% accuracy when the SNR is at least 15 dB.
- Compact model: The proposed model has been shown to reduce the number of parameters by 50 times compared to the large CNN model proposed in Ref. [1].
- Implementability: The PRNN-CNN model can be fully implemented on a small-scale FPGA, and has the potential to be implemented on a hybrid photonic/FPGA edge processor that consists of a photonic RNN to process sequential data with low latency and a CNN on the FPGA to enable massive parallel machine learning inference.

- Hardware efficiency: We estimate the power and latency for the proposed RF fingerprinting system using PYNQ, and show the throughput may be improved by over 200 times and the latency reduced to more than 100 times over previous approaches.

With the above achievement, we show the potential of using the small-scale FPGA processor to perform RF fingerprinting in real time. A number of issues are still left open in this investigation such as the improvement for noisy data, and real-time implementation of RF fingerprinting on a hybrid photonic/FPGA processor. With photonic model, the latency will theoretically be further improved, however, this will require more demonstration and we will leave this topic as our future work.

On the other hand, in the review of our work on PNN for fiber nonlinearity compensation, we have shown a programmable opto-electronic integrated neural network on a silicon photonic platform that has achieved the following milestones:

- Fully on-chip neural information processing, including all the essential functionalities of a neural network, i.e. weighting, summing, and biased nonlinear activation.
- Success of experimental of neural network model for NLC with comparable Q-factor improvement (0.60 dB).
- Low latency processing for NLC (13.3 ns for carrier-injection MRR modulator, and 200 ps for carrier-depletion MRR modulator).

In summary, high-accuracy, real-time signal processing on a hardware platform with small size, weight and power (SWaP) will enable the deployment of ML technique at the communication edge. This work makes important contributions to this endeavor with both innovative compact neural network models and experimental validation with both PNN and a small scale FPGA emulation for applications of fiber NLC and RF fingerprinting respectively. These provide a solid foundation to show the potential of using microring-based silicon PNN for AI-assisted communication at the edge.

Appendix A
Residual Data Processing

In the appendix, we will provide details on residual data processing, neural networks on silicon photonic circuits, and the NRL and PRNN models for RF fingerprinting. Our code to implement experiments on GPUs/CPUs is avaiable at https://github.com/Hsuan-Tung/PCICN_RFFingerprinting, and we also provide a live demonstration for real-time RF fingerprinting using PYNQ-Z1 FPGA board, which can be watched at the https://www.youtube.com/watch?v=CIfddiscE3k.

The RF transmitters of the same type follow the same steps and communication protocols to process the transmitting bits by encoding, modulation, pulse shaping and up-conversion to generate the signals for transmission. Due to the electronic component's performance variations, the transmitted signals carry small and hidden information which is intrinsic to each individual transmitter. We employ a residual data processing step, which extracts an error signal by subtracting the recovered data waveform from the received raw data waveform. The error signal, instead of the received raw data, is fed to the learning algorithm and neural networks for classification. Ideally, the error signal contains only residual signal elements directly associated with the intrinsic makeup of the transmitting device, rather than the transmitted bit information which the transmitters can change almost randomly.

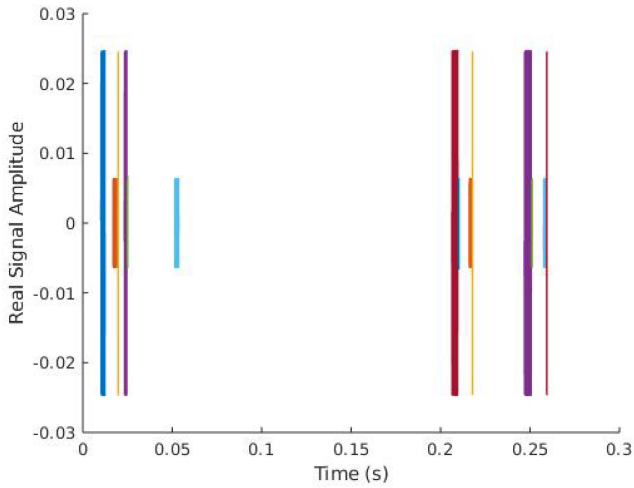


Fig. 15. The real portion of the first 300 ms of broadband radio frequency data for a single device. Each color corresponds to a separate transmission.

The residual data processing, which serves as a pre-processing step before the neural network classification, removes the information data from the signal by recovering the measured data, generating an ideal signal matching this data, then subtracting the ideal signal from the measured signal, leaving only non-idealities associated intrinsically with each individual device. In order to generate the ideal signal, the measured signal is decoded then re-encoded. This section describes how this was done.

A. Raw RF Data Settings

The Naval Research Laboratory collected radio frequency data measured from thirty ZigBee Pro devices operating at a central carrier frequency of 2.405 GHz, and a power level of 18 dBm. The data provided has been downconverted to a complex baseband signal at 16 MHz in a 32-bit floating point form (4 bytes per value, or 8 bytes per complex sample). No further corrections had been applied to the data. In the following subsections, we will describe the details of residual data processing step by step.

B. Separating Out Transmissions

Under the ZigBee protocol, a device broadcasts data separated into individual transmissions, each of which may be considered separately. These are sliced out by identifying large duration during which the signal is below a threshold, which separate transmissions. A segment of data with multiple transmissions is shown in Fig. 15.

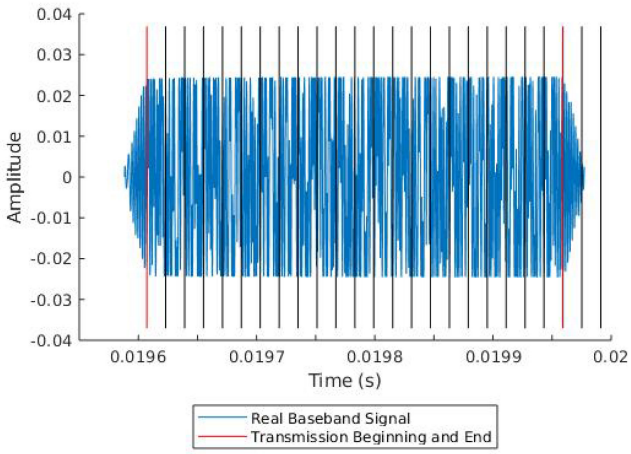


Fig. 16. The real portion of a single transmission, with beginning and end determined and the entire transmission segmented into separate samples.

C. Trimming and Synchronizing the Transmission

To be decoded correctly, transmissions must be aligned precisely, trimming off the start and end of the transmission to begin and end at the right sample. Each transmission begins with a fixed five bytes preamble, corresponding to four bytes of zeros and one byte equal to 0xA7. The beginning of the transmission may be identified by matching the signal corresponding to an ideal preamble to the measured signal. This is complicated by the fact that the initial phase of the signal is unknown. To avoid this issue, the discrete derivative of the phase of the complex ideal and measured signals were compared (unwrapped to avoid issues of phase jumping between -2π and 2π). This quantity may still be accurately matched to the preamble while being independent of the initial phase.

Mathematically t_0 was determined such that the following quantity was minimized:

$$\sqrt{\sum_t \left\{ \frac{d}{dt}\left[\text{unwrap}(\angle x(t+t_0))\right] - \frac{d}{dt}\left[\text{unwrap}(\angle y(t))\right] \right\}}$$

Where $\angle x$ represents the phase of the measured signal and $\angle y$ the phase of the reference preamble, with the sum over the length of y.

Fig. 16 shows a single transmission, equivalent to the third transmission (in yellow) in Fig. 15. The plot is segmented into signals (four bytes sequences) separated by the vertical black and red lines. The first ten signals correspond to the preamble. (This is a very short transmission, and the preamble makes up the bulk of it). The position of the first red line, the start of the transmission, was determined through the above process.

Another red line indicates the end of the transmission. Every transmission has a length equivalent to an even number of signals, and the end is determined by finding the latest sample obeying this requirement. As the signal decoder is reasonably robust to additional length in the case that the chosen end sample is too late, more precision is not required.

D. Correcting Phase and Frequency Offsets

Data transmitted according to the ZigBee protocol is encoded using offset quadrature phase shift keying (O-QPSK). The signal has in-phase (I) and quadrature (Q) components which must be accurately separated.

The following model on the separation of the I and Q components is drawn from a paper published by [1], with some adjustments and additions.

The data is encoded by the transmitter as a complex baseband signal $\tilde{a}(t)$, with time t. The transmitted signal $b(t)$, with carrier frequency ω_c would then be:

$$b(t) = \text{Re}\left[\tilde{a}(t)e^{i\omega_c t}\right] \tag{10}$$

However, suppose the transmitter frequency and phase differs from the receiver frequency and phase. Modeling the difference as a phase offset ϕ_o and frequency offset ω_o at the transmitter side, the transmitted signal is then:

$$b(t) = \text{Re}\left[\tilde{a}(t)e^{i[(\omega_c+\omega_o)t+\phi_0]}\right] \tag{11}$$

The receiver measures the signal and downconverts it to a baseband signal $\tilde{c}(t)$, "removing" the carrier frequency ω_c. In addition, there is a substantial attenuation in signal amplitude between the transmitter and receiver, represented by α, and other phase distortion factors, small compared to the effect of ω_o and ϕ_o, represented by $D(t)$. $D(t)$ includes elements associated with a device's unique wireless signature.

$$\tilde{c}(t) = \alpha\tilde{a}(t)e^{i(\omega_o t+\phi_0+D(t))} \tag{12}$$

The challenge is to extract $\tilde{a}(t)$ (or, more accurately, its phase for QPSK) from the measured $\tilde{c}(t)$ so that its effect may be subtracted out to produce the error signal, incorporating effects associated with $D(t)$. Let ω and ϕ represent the estimated phase and frequency correction, designed to compensate for ω_o and ϕ_o. These are applied to the measured signal $\tilde{c}(t)$ to produce a corrected signal $\tilde{d}(t)$:

$$\tilde{d}(t) = \tilde{c}(t)e^{-i(\omega t+\phi)} \tag{13}$$

We will find ω and ϕ such that $\omega = \omega_o$ and $\phi = \phi_0$:

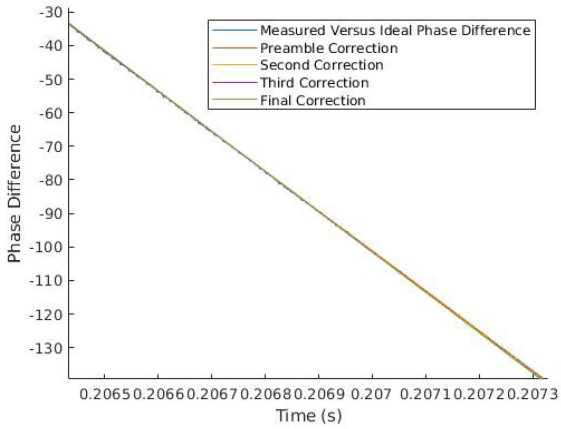
$$\tilde{d}(t) = \tilde{c}(t)e^{-i(\omega t+\phi)} = \alpha\tilde{a}(t)e^{i[(\omega_o-\omega)t+\phi_o-\phi+D(t)]}$$
$$= \alpha\tilde{a}(t)e^{iD(t)} \tag{14}$$

Representing $a(t)$ and $c(t)$ as quantities with amplitudes and phases, the latter represented $\phi_a(t)$ and $\phi_c(t)$, we have a phase equation:

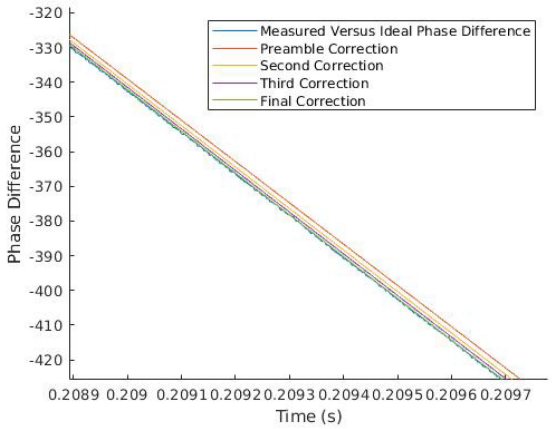
$$\phi_c(t) - \omega t - \phi = \phi_a(t) + D(t) \rightarrow \text{unwrap}\left(\phi_c(t)\right.$$
$$\left.-\phi_a(t)\right) = \omega t + \phi + D(t) \tag{15}$$

The function unwrap() above means that the phase difference is unrolled, allowing it to increase past π to 2π, 3π, 4π and so forth rather than circling back to $-\pi$.

Equation (15), in words: the phase difference between the measured signal $\tilde{c}(t)$ and the ideal signal $\tilde{a}(t)$ increases roughly linearly, with an initial phase offset ϕ and a slope corresponding to the frequency offset ω. There is also a small deviation from this linear model, $D(t)$. If one has both the measured signal and



(a)



(b)

Fig. 17. The phase difference between a typical measured transmission and each iterative linear correction. (a) Early in the transmission. Note that all corrections are accurate. (b) Near the end of the transmission. The earlier corrections have drifted substantially.

an idea signal, linear regression may be used to estimate ω and ϕ.

The fixed first five bytes of a transmission (the preamble) are used to generate an ideal reference signal matching the first five bytes of the measured transmission. Linear regression produces estimates of ω and ϕ. In theory, these corrections may be applied to the measured signal $\tilde{c}(t)$ to produce $\tilde{d}(t)$, which may be decoded, as its I and Q components are appropriately separated.

Unfortunately, the estimates of the frequency offset ω based on the first five bytes of the signal are too imprecise. As time progresses, any slight inaccuracy in ω caused by the semi-random $D(t)$ builds up. Once the phase discrepancy at a certain point in time gets too high, the I and Q components of the signal will not be accurately separated and the decoder fails.

Fig. 17 illustrates this problem. Early in the transmission, shown in Fig. 17(a), the linear fit associated with the preamble correction is quite accurate. By the end of the transmission, in Fig. 17(b), the fit has drifted several radians from the phase difference.

The solution was to implement an iterative correction algorithm. The preamble correction is used to decode the signal up to the point that the decoder fails. The correctly decoded

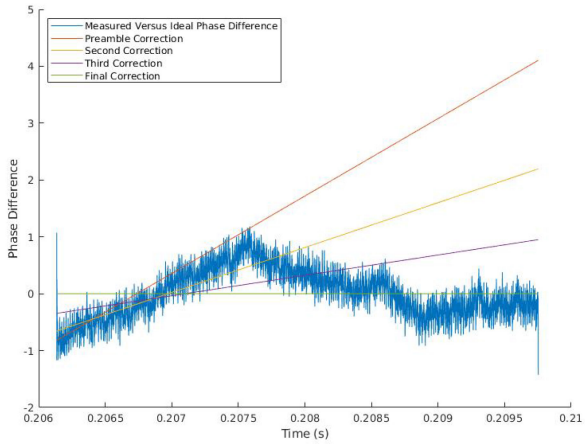


Fig. 18. The phase difference between a typical measured transmission and each iterative correction, with the final correction subtracted out.

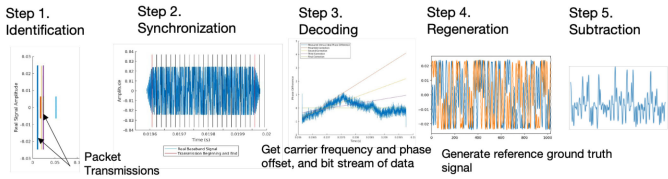


Fig. 19. Procedure of residual data processing.

portion may then be used to generate a longer ideal signal and therefore a longer phase difference plot. Linear regression on this extended dataset produces more precise estimates of the phase and frequency offset, which in turn allow the measured signal to be more accurately corrected. This loop continues until the entire signal had been decoded, typically after 1–3 iterations following the preamble correction.

Fig. 17 shows each of these further corrections. Notably, in Fig. 17(b) the final correction has not drifted from the phase difference signal. Fig. 18 illustrates this in a more readable fashion. The final correction has been subtracted from the phase difference and the other corrections, so that the final correction itself appears flat. Each correction represents the best fit for a segment of the data beginning at the start of the transmission, with this segment lengthening with each correction.

With the entire signal decoded, we have a corrected signal $\tilde{d}(t)$ and an idea signal corresponding to $\alpha\tilde{a}(t)$, the original ideal signal scaled to match the measured signal. The error signal $\tilde{e}(t)$ is then:

$$\tilde{e}(t) = \tilde{d}(t) - \alpha\tilde{a}(t) \tag{16}$$

The overall residual processing is summarized in Fig. 19.

Appendix B
Neural Network Classifiers for RF Fingerprinting

In Ref. [1], the authors proposed a multi-layer convolutional neural network to classify the transmission of 7 identical ZigBee devices. We have built the same structure CNN as proposed in [1] but for classification of 30 devices. The network has 322,602 trainable parameters, and Table III shows the structure

TABLE III
MULTI-LAYER CNN ARCHITECTURE FOR 30 DEVICES CLASSIFICATION

Layer	Dimension (channel, length)	Parameters	Activation
Input	2×1024	–	–
Conv1D	128×19	4992	ELU
Max Pooling	–	–	–
Conv1D	32×15	61472	ELU
Max Pooling	–	–	
Conv1D	16×11	5648	ELU
Max Pooling	–	–	
Flatten	–	–	
Fully-Connected	128	239744	ELU
Dropout	–	–	
Fully-Connected	64	8256	ELU
Dropout	–	–	
Fully-Connected	30	1950	Log Softmax

TABLE IV
PRNN ASSISTED RF FINGERPRINTING SYSTEM

Layer	Dimension (channel, length)	Parameters	Activation
I/Q Input	2×1024	–	–
Reshape	64×32	–	–
PRNN	16×32	1312	Lorenztian
Conv1D	16×5	1296	ELU
Max Pooling	–	–	–
Conv1D	16×3	784	ELU
Max Pooling	–	–	
Flatten	–	–	
Fully-Connected	30	2910	Log Softmax

of multi-layer CNN used for classification of 30 Zigbee devices. In this CNN, aside from the last layer which has a log softmax nonlinearity, all the layers use the exponential linear unit as their activation function, which has the following nonlinearity:

$$ELU(x) = \begin{cases} x, \text{ if } x > 0 \\ \alpha(exp(x) - 1), \text{ if } x \leq 0 \end{cases}$$

In our experiment, we set the hyperparameter $\alpha = 1$.The details of our proposed PRNN-CNN architecture are shown in Table IV. It has a total of 6,302 parameters. In the PRNN layer, the nonlinear activation function is a Lorentzian function, which is designed to match the nonlinear transfer function of a silicon photonic modulator [34].

ACKNOWLEDGMENT

The authors would like to thank Thomas Carroll and Bryan Nousain from Naval Research Lab for providing the data source and discussions on RF fingerprinting techniques, and Dr. Young-Kai Chen and Dr. Gregory Jones from DARPA for the support and insightful discussions.

REFERENCES

[1] K. Merchant, S. Revay, G. Stantchev, and B. D. Nousain, "Deep learning for RF device fingerprinting in cognitive communication networks," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 160–167, Feb. 2018.

[2] C. Huang et al., "A silicon photonic-electronic neural network for fibre nonlinearity compensation," *Nature Electron.*, vol. 4, pp. 837–844, 2021.

[3] K. Merchant and B. D. Nousain, "Securing IoT RF fingerprinting systems with generative adversarial networks," in *Proc. IEEE Mil. Commun. Conf.*, 2019, pp. 584–589.

[4] C. Huang et al., "Demonstration of photonic neural network for fiber nonlinearity compensation in long-haul transmission systems," in *Proc. Opt. Fiber Commun. Conf. Exhib.*, 2020, pp. 1–3.

[5] S. Zhang et al., "Field and lab experimental demonstration of nonlinear impairment compensation using neural networks," *Nature Commun.*, vol. 10, no. 1, pp. 1–8, 2019.

[6] 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: Springer, 2016, pp. 213–226.

[7] P. Antonik et al., "Online training of an opto-electronic reservoir computer applied to real-time channel equalization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 11, pp. 2686–2698, Nov. 2017.

[8] E. G. Giacoumidis, Y. Lin, M. Blott, and L. P. Barry, "Real-time machine learning based fiber-induced nonlinearity compensation in energy-efficient coherent optical networks," *Appl. Phys.*, vol. 5, no. 4, 2020, Art. no. 041301.

[9] T. J. 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.

[10] T. Erpek, T. J. O'Shea, Y. E. Sagduyu, Y. Shi, and T. C. Clancy, "Deep learning for wireless communications," in *Proc. Develop. Anal. Deep Learn. Architectures*, 2020, pp. 223–266.

[11] B. J. Shastri et al., "Photonics for artificial intelligence and neuromorphic computing," *Nature Photon.*, vol. 15, pp. 102–114, 2020.

[12] H.-T. Peng, M. A. Nahmias, T. F. De Lima, A. N. Tait, and B. J. Shastri, "Neuromorphic photonic integrated circuits," *IEEE J. Sel. Topics Quantum Electron.*, vol. 24, no. 6, Nov./Dec. 2018, Art. no. 6101715.

[13] M. A. Nahmias et al., "Photonic multiply-accumulate operations for neural networks," *IEEE J. Sel. Topics Quantum Electron.*, vol. 26, no. 1, Jan./Feb. 2020, Art. no. 7701518.

[14] T. F. De Lima et al., "Machine learning with neuromorphic photonics," *J. Lightw. Technol.*, vol. 37, no. 5, pp. 1515–1534, Mar. 2019.

[15] H. Jaeger, "The 'Echo State' approach to analysing and training recurrent neural networks-with an erratum note," Bonn, Germany: German Nat. Res. Center Inf. Technol. GMD Tech. Rep. 148, 2001.

[16] A. Rodan and P. Tino, "Minimum complexity echo state network," *IEEE Trans. Neural Netw.*, vol. 22, no. 1, pp. 131–144, Jan. 2011.

[17] S. Takeda et al., "Photonic reservoir computing based on laser dynamics with external feedback," in *Proc. Int. Conf. Neural Inf. Process.*, 2016, pp. 222–230.

[18] L. Larger et al., "High-speed photonic reservoir computing using a time-delay-based architecture: Million words per second classification," *Phys. Rev. X*, vol. 7, Feb. 2017, Art. no. 011015. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevX.7.011015

[19] L. Appeltant et al., "Information processing using a single dynamical node as complex system," *Nature Commun.*, vol. 2, 2011, Art. no. 468.

[20] A. Argyris, J. Bueno, and I. Fischer, "Photonic machine learning implementation for signal recovery in optical communications," *Sci. Rep.*, vol. 8, 2018, Art. no. 8487.

[21] K. Vandoorne et al., "Experimental demonstration of reservoir computing on a silicon photonics chip," *Nature Commun.*, vol. 5, no. 1, pp. 1–6, 2014.

[22] F. D.-L. Coarer et al., "All-optical reservoir computing on a photonic chip using silicon-based ring resonators," *IEEE J. Sel. Topics Quantum Electron.*, vol. 24, no. 6, Nov.–Dec. 2018, Art. no. 7600108.

[23] A. D. Annoni et al., "Unscrambling light-automatically undoing strong mixing between modes," *Light, Sci., Appl.*, vol. 6, 2017, Art. no. e17110 .

[24] A. N. Tait et al., "Microring weight banks," *IEEE J. Sel. Topics Quantum Electron.*, vol. 22, no. 6, pp. 312–325, Nov./Dec. 2016.

[25] P. Y. Ma et al. Prucnal, "Blind source separation with integrated photonics and reduced dimensional statistics," *Opt. Lett.*, vol. 45, pp. 6494–6497, 2020.

[26] P. Y. Ma et al., "Photonic independent component analysis using an on-chip microring weight bank," *Opt. Exp.*, vol. 28, pp. 1827–1844, 2020.

[27] D. Psaltis, D. Brady, and K. Wagner, "Adaptive optical networks using photorefractive crystals," *Appl. Opt.*, vol. 27, no. 9, pp. 1752–1759, May 1988.

[28] D. Psaltis et al., "Optoelectronic implementations of neural networks," *IEEE Commun. Mag.*, vol. 27, pp. 37–40, Nov. 1989.

[29] B. J. Shastri et al., "Spike processing with a graphene excitable laser," *Sci. Rep.*, vol. 6, 2016, Art. no. 19126126.
[30] D. Brunner, M. C. Soriano, C. R. Mirasso, and I. Fischer, "Parallel photonic information processing at gigabyte per second data rates using transient states," *Nature Commun.*, vol. 4, 2013, Art. no. 1364. [Online]. Available: http://dx.doi.org/10.1038/ncomms2368
[31] X. Lin et al., "All-optical machine learning using diffractive deep neural networks," *Science*, vol. 361, no. 6406, pp. 1004–1008, 2018.
[32] J. Chang, V. Sitzmann, X. Dun, W. Heidrich, and G. Wetzstein, "Hybrid optical-electronic convolutional neural networks with optimized diffractive optics for image classification," *Sci. Rep.*, vol. 8, 2018, Art. no. 12324.
[33] T. Zhou et al., "In situ optical backpropagation training of diffractive optical neural networks," *Photon. Res.*, vol. 8, pp. 940–953, 2020.
[34] A. N. Tait et al., "Silicon photonic modulator neuron," *Phys. Rev. Appl.*, vol. 11, no. 6, 2019, Art. no. 064043.
[35] Y. Shen et al., "Deep learning with coherent nanophotonic circuits," *Nature Photon.*, vol. 11, no. 7, 2017, Art. no. 441.
[36] J. Feldmann, N. Youngblood, C. D. Wright, H. Bhaskaran, and W. Pernice, "All-optical spiking neurosynaptic networks with self-learning capabilities," *Nature*, vol. 569, no. 7755, pp. 208–214, 2019.
[37] A. N. Tait, T. Ferreira de Lima, M. A. Nahmias, B. J. Shastri, and P. R. Prucnal, "Multi-channel control for microring weight banks," *Opt. Exp.*, vol. 24, no. 8, pp. 8895–8906, Apr. 2016.
[38] H. Jayatilleka et al., "Crosstalk in SOI microring resonator-based filters," *J. Lightw. Technol.*, vol. 34, no. 12, pp. 2886–2896, Jun. 2016. [Online]. Available: http://dx.doi.org/10.1109/JLT.2015.2480101
[39] A. N. Tait et al., "Feedback control for microring weight banks," *Opt. Exp.*, vol. 26, no. 20, pp. 26422–26443, 2018.
[40] C. Huang et al., "Demonstration of scalable microring weight bank control for large-scale photonic integrated circuits," *APL Photon.*, vol. 5, no. 4, 2020, Art. no. 040803.
[41] W. Zhang et al., "Silicon microring synapses enable photonic deep learning beyond 9-bit precision," *Optica*, vol. 9, no. 5, pp. 579–584, 2022.
[42] M. S. Hai, M. N. Sakib, and O. Liboiron-Ladouceur, "A 16 Ghz silicon-based monolithic balanced photodetector with on-chip capacitors for 25 Gbaud front-end receivers," *Opt. Exp.*, vol. 21, pp. 32680–32689, 2013.
[43] S. Chen, F. Xie, Y. Chen, H. Song, and H. Wen, "Identification of wireless transceiver devices using radio frequency (RF) fingerprinting based on STFT analysis to enhance authentication security," in *Proc. IEEE 5th Int. Symp. Electromagn. Compat.*, 2017, pp. 1–5.
[44] J. Yu, A. Hu, G. Li, and L. Peng, "A robust RF fingerprinting approach using multisampling convolutional neural network," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6786–6799, Aug. 2019.
[45] T. Jian et al., "Deep learning for RF fingerprinting: A massive experimental study," *IEEE Internet Things Mag.*, vol. 3, no. 1, pp. 50–57, Mar. 2020.
[46] A. N. Tait et al., "Neuromorphic photonic networks using silicon photonic weight banks," *Sci. Rep.*, vol. 7, no. 1, 2017, Art. no. 7430.
[47] A. Paszke et al., "Pytorch: An Imperative style, high-performance deep learning library," in *Proc. 33rd Adv. Neural Inf. Process. Syst.*, 2019, pp. 8026–8037.
[48] P. Werbos, "Backpropagation through time: What it does and how to do it," *Proc. IEEE*, vol. 78, no. 10, pp. 1550–1560, Oct. 1990.
[49] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2010, pp. 249–256.
[50] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1026–1034.
[51] D. P. Kingma and J. Ba, "Adam: A. method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015.
[52] "Xilinx, vivado design suite user guide: High-level synthesis," 2015. [Online]. Available: http://www.xilinx.com/support/documentation/sw_manuals/xilinx2015_2/ug902-vivado-high-level-synthesis.pdf
[53] M. Lechner, R. M. Hasani, and R. Grosu, "Neuronal circuit policies," 2018, *arXiv:1803.08554*.
[54] R.-J. Essiambre, G. Kramer, P. J. Winzer, G. J. Foschini, and B. Goebel, "Capacity limits of optical fiber networks," *J. Lightw. Technol.*, vol. 28, no. 4, pp. 662–701, 2010.
[55] Z. Tao et al., "Multiplier-free intrachannel nonlinearity compensating algorithm operating at symbol rate," *J. Lightw. Technol.*, vol. 29, no. 17, pp. 2570–2576, Sep. 2011.
[56] E. Ip and J. M. Kahn, "Compensation of dispersion and nonlinear impairments using digital backpropagation," *J. Lightw. Technol.*, vol. 26, no. 20, pp. 3416–3425, 2008.
[57] L. B. Du et al., "Digital fiber nonlinearity compensation: Toward 1-tb/s transport," *IEEE Signal Process. Mag.*, vol. 31, no. 2, pp. 46–56, Mar. 2014.
[58] A. N. Tait, M. A. Nahmias, B. J. Shastri, and P. R. Prucnal, "Broadcast and weight: An integrated network for scalable photonic spike processing," *J. Lightw. Technol.*, vol. 32, no. 21, pp. 4029–4041, Nov. 2014.
[59] P. Alipour, A. Atabaki, M. Askari, A. Adibi, and A. A. Eftekhar, "Robust postfabrication trimming of ultracompact resonators on silicon on insulator with relaxed requirements on resolution and alignment," *Opt. Lett.*, vol. 40, pp. 4476–4479, 2015.
[60] P. Dong et al., "Thermally tunable silicon racetrack resonators with ultralow tuning power," *Opt. Exp.*, vol. 18, no. 19, pp. 20298–20304, 2010.
[61] J. E. Cunningham et al., "Highly-efficient thermally-tuned resonant optical filters," *Opt. Exp.*, vol. 18, no. 18, pp. 19055–19063, 2010.
[62] C. Errando-Herranz et al., "MEMS for photonic integrated circuits," *IEEE J. Sel. Topics Quantum Electron.*, vol. 56, no. 1, Feb. 2020, Art. no. 8400210.
[63] F. Eltes et al., "A batio3-based electro-optic pockels modulator monolithically integrated on an advanced silicon photonics platform," *J. Lightw. Technol.*, vol. 37, no. 5, pp. 1456–1462, Mar. 2019.
[64] S. Abel et al., "Large pockels effect in micro- and nanostructured barium titanate integrated on silicon," *Nature Mater.*, vol. 18, pp. 42–47, 2018.
[65] T. F. de Lima et al., "Real-time operation of silicon photonic neurons," in *Proc. Opt. Fiber Commun. Conf.*, 2020, Paper. no. M2K–4.
[66] M. J. Marinella et al., "Multiscale co-design analysis of energy, latency, area, and accuracy of a reram analog neural training accelerator," *IEEE Trans. Emerg. Sel. Topics Circuits Syst.*, vol. 8, no. 1, pp. 86–101, Mar. 2018.
[67] P. Dong et al., "High speed silicon microring modulator based on carrier depletion," in *Proc. Conf. Opt. Fiber Commun.*, 2010, pp. 1–3.
[68] Y. Zhang et al., "200 Gbit/s optical pam4 modulation based on silicon microring modulator," in *Proc. Eur. Conf. Opt. Commun.*, 2020, pp. 1–4.

Hsuan-Tung Peng received the B.S. degree in physics from National Taiwan University, Taipei, Taiwan, in 2015, and the Ph.D. degree in electrical and computer engineering from Princeton University, Princeton, NJ, USA, in February 2022. He is currently a photonic integrated circuits (PIC) R&D Engineer with PsiQuantum, Taipei, Taiwan. His research interests include neuromorphic photonics, photonic-integrated circuits, optical signal processing, and photonic quantum computing.

Joshua C. Lederman received the graduation degree, in 2019, from Cornell University, Ithaca, NY, USA, where he studied Engineering Physics while researching in areas, including high-energy plasma physics and deep-ultraviolet LEDs. He is currently working toward the Ph.D. degree with Lightwave Laboratory, Princeton University, Princeton, NJ, USA, where he develops neuromorphic photonic systems, including FPGA-photonic cointegrated processors, with application to radio-frequency fingerprinting and convolutional image processing.

Lei Xu received the B.S. degree in geophysics from Peking University, Beijing, China, in 1997, the M.Eng. degree in electronic engineering from Tsinghua University, Shanghai, China, in 2000, and the Ph.D. degree from Princeton University, Princeton, NJ, USA, in 2005. He is currently a Research Scholar with Lightwave Lab, Electrical and Computer Engineering Department, Princeton University. He was a Senior Research Staff Member with NEC Labs America from 2005 to 2012, and a Technology Cofounder of Sodero Networks, Torray Networks, and Eagle Nebula Inc from 2012 to 2019. He has authored or coauthored more than 100 research papers, and has 59 U.S. patents. His research interests include neuromorphic photonic computing, silicon photonics, software-defined optical networking, and high-speed optical communications.

Thomas Ferreira de Lima received the bachelor's degree and the Ingénieur Polytechnicien master's degree from Ecole Polytechnique, Palaiseau, France, with a focus on physics for optics and nanosciences, and the Ph.D. degree in electrical engineering from the Lightwave Communications Group, Department of Electrical Engineering, Princeton University, Princeton, NJ, USA. His research interests include integrated photonic systems, nonlinear signal processing with photonic devices, spike-timing-based processing, ultrafast cognitive computing, and dynamical light-matter neuro-inspired learning and computing. He is also a contributing author to the textbook, *Neuroomorphic Photonics*.

Chaoran Huang (Member, IEEE) received the B.Eng. degree in optoelectronic engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2012, and the Ph.D. degrees in electronic engineering (photonics) from The Chinese University of Hong Kong (CUHK), Hong Kong, in 2016. During 2017–2021, She was a Postdoctoral Research Associate with Princeton University, Princeton, NJ, USA. She is currently an Assistant Professor with CUHK. She has authored or coauthored more than 50 peer-reviewed research papers. Her research interests include photonic integrated circuits, optical information processing, and nonlinear optics with emphasis on applications, such as neuromorphic computing, photonic neural networks, and optical communications. Her Ph.D. thesis was selected as finalist for CUHK Engineering Faulty Outstanding Thesis Award, for her work in optical signal processing for advanced fiber communications. She was the recipient of 2019 the Rising Stars Women in Engineering Asia. She is an Active Reviewer for high quality international journals, and the TPC Member of International Conferences.

Bhavin J. Shastri received the Honours B.Eng. (with distinction), M.Eng., and Ph.D. degrees in electrical engineering (photonics) from McGill University, Montreal, QC, Canada, in 2005, 2007, and 2012, respectively. He is currently an Assistant Professor of engineering physics with Queen's University, Kingston, ON, Canada, and a Faculty Affiliate with the Vector Institute for Artificial Intelligence, Canada. He was an NSERC and Banting Postdoctoral Fellow (during 2012–2016) and an Associate Research Scholar (during 2016–2018) with Princeton University, Princeton, NJ, USA. He has authored or coauthored more than 70 journal articles and 90 conference proceedings, seven book chapters, and given more than 60 invited talks and lectures, including five keynotes and three tutorials. His research interests include silicon photonics, photonic integrated circuits, neuromorphic computing, and machine learning. He is a coauthor of the book, *Neuromorphic Photonics* (Taylor & Francis, CRC Press, 2017).

from the ICO. He is a Senior Member of the OSA. Dr. Shastri was the recipient of the 2020 IUPAP Young Scientist Prize in Optics for his pioneering contributions to neuromorphic photonics, 2014 Banting Postdoctoral Fellowship from the Government of Canada, 2012 D. W. Ambridge Prize for the top graduating Ph.D. student at McGill, IEEE Photonics Society 2011 Graduate Student Fellowship, 2011 NSERC Postdoctoral Fellowship, 2011 SPIE Scholarship in Optics and Photonics, 2008 NSERC Alexander Graham Bell Canada Graduate Scholarship, including the best student paper awards at the 2014 IEEE Photonics Conference, 2010 IEEE Midwest Symposium on Circuits and Systems, 2004 IEEE Computer Society Lance Stafford Larson Outstanding Student Award, and 2003 IEEE Canada Life Member Award.

David Rosenbluth biography is not available at the time of publication.

Paul R. Prucnal (Life Fellow, IEEE) received the A.B. (graduating *summa cum laude*) in mathematics and physics from Bowdoin College, Brunswick, ME, USA, and the M.S., M.Phil. and Ph. D. degrees in electrical engineering from Columbia University, New York, NY USA. After the doctorate, Prucnal joined the faculty with Columbia University, where he was a Member of the Columbia Radiation Laboratory, he performed groundbreaking work in OCDMA and self-routed photonic switching. In 1988, he joined the Faculty with Princeton University, Princeton, NJ, USA. He has authored or coauthored more than 350 journal articles and book chapters and holds 28 U.S. patents. His research on optical CDMA initiated a new research field in which more than 1000 papers have since been published, exploring applications ranging from information security to communication speed and bandwidth. In 1993, he invented the Terahertz Optical Asymmetric Demultiplexer, the first optical switch capable of processing terabit per second (Tb/s) pulse trains. He is the Author of the book, *Neuromorphic Photonics*, and the Editor of the book, *Optical Code Division Multiple Access: Fundamentals and Applications*. He was the Area Editor of IEEE TRANSACTIONS ON COMMUNICATIONS. He is a Fellow of the Optical Society of America (OSA) and the National Academy of Inventors (NAI), and a Member of honor societies, including Phi Beta Kappa and Sigma Xi. He was the recipient of the 1990 Rudolf Kingslake Medal for his paper entitled Self-routing photonic switching with optically-processed control, Gold Medal from the Faculty of Mathematics, Physics and Informatics at the Comenius University, for leadership in the field of Optics 2006 and has won multiple teaching awards at Princeton, including the E-Council Lifetime Achievement Award for Excellence in Teaching, the School of Engineering and Applied Science Distinguished Teacher Award, The President's Award for Distinguished Teaching. He has been instrumental in founding the field of Neuromorphic Photonics and developing the photonic neuron, a high speed optical computing device modeled on neural networks, and integrated optical circuits to improve wireless signal quality by cancelling radio interference.