Robust Deep Sensing Through Transfer Learning in Cognitive Radio

Qihang Peng, Andrew Gilman, Nuno Vasconcelos, Pamela C. Cosman, and Laurence B. Milstein

***Abstract*—We propose a robust spectrum sensing framework based on deep learning. The received signals at the secondary user's receiver are filtered, sampled and then directly fed into a convolutional neural network. Although this deep sensing is effective when operating in the same scenario as the collected training data, the sensing performance is degraded when it is applied in a different scenario with different wireless signals and propagation. We incorporate transfer learning into the framework to improve the robustness. Results validate the effectiveness as well as the robustness of the proposed deep spectrum sensing framework.**

***Index Terms*—Spectrum sensing, deep learning, robustness, transfer learning, cognitive radio.**

I. Introduction

Spectrum sensing enables cognitive radios (CRs) to discover unused spectrum of primary users (PUs), such that secondary users (SUs) can access unused bands to increase spectral utilization of the network [1]–[3]. Spectrum sensing is of critical importance for the realization of CR.

In recent years, deep learning (DL) techniques have achieved great success on many complex tasks, and the best performance is often obtained with end-to-end models [4], [5], where a DL system learns appropriate features in a data-driven fashion, instead of using hand-crafted features. Such models may also have potential in spectrum sensing.

A DL model was proposed in [6] for cooperative spectrum sensing, where the CR network combines the sensing results from each SU. Measured received signal strength (RSS) or binary sensing decisions were input to a deep neural network (DNN). A recent work on modulation recognition [7] using raw samples of the in-phase and quadrature-phase of the received temporal signals as input to a DNN shows significant gains compared to using conventional features. However, DL-based approaches require significant amounts of labeled training data which follows the same distribution as the test data. In [8] and [9], the authors propose adversarial generative networks to augment training examples, as well as domain adaptation to switch between signal types.

In this letter, we propose a DL-based spectrum sensing system, called deep sensing hereafter. Unlike existing DL-based spectrum sensing using expert features, the proposed method uses raw signals as inputs to a DNN. As we show that a DNN trained under one set of conditions may not perform well when wireless conditions change, we propose to incorporate transfer learning (TL) [10] to adapt the learned models to new communications settings. Results show that TL significantly improves the robustness.

The main contributions of this letter are: (1) It extends O'Shea's work on modulation recognition with DL to spectrum sensing in CR, and evaluates its performance against the optimal and energy detection. (2) This is the first exploration of transfer learning to address robustness in DL-based spectrum sensing. We consider both cases of domain adaptation (no labeled training examples) and fine-tuning (a small number of labeled training examples). (3) For the cases studied, we observed that very few labeled data were needed for a robust transfer to the new conditions. The rest of this letter is organized as follows. Section II presents the deep sensing algorithm and its performance. Robustness is analyzed, and two transfer learning frameworks are examined in Section III.

II. Deep Spectrum Sensing

Received radio signals pass through a rectangular bandlimited filter to limit noise, and then are sampled, producing a discrete-time sequence. A subsequence of N complex-valued samples, collected during a single sensing interval, is decomposed as a $2 \times N$ real-valued vector, with the first and second row being the in-phase and quadrature components, respectively, and forms a single input vector $\mathbf{x}$ to a DNN. The DNN outputs a binary class label y with value $y = 1$ when the SU makes a decision that the PU is present and $y = 0$ that the PU is absent.

We use a convolutional neural network (CNN) with two convolutional layers, followed by two dense layers (Table I). For the two convolutional layers, the stride is 1 and the zero padding equals 4. Rectified linear (ReLU) activation units are used as the non-linearity in each layer. Dropout with a rate of 0.50 is used to regularize fully connected and convolutional layers, to reduce over-fitting. The Adam optimizer is utilized, and the last layer uses the logistic function. Given a training set of n sensing interval examples $\mathbf{x}_i$ and their class labels y_i, denoted $D = \{\mathbf{x}_i, y_i\}_{i=1}^{n}$, the network parameters are learned by minimizing the empirical risk

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \frac{1}{n} \sum_i L[f(\mathbf{x}_i; \mathbf{w}), y_i] \tag{1}$$

Manuscript received June 28, 2019; revised August 20, 2019; accepted September 3, 2019. Date of publication September 10, 2019; date of current version January 9, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61301154, and in part by the China Scholarship Council under Grant 201506075100. The associate editor coordinating the review of this article and approving it for publication was C. Shen. *(Corresponding author: Qihang Peng.)*

Q. Peng is with the School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: anniepqh@uestc.edu.cn).

A. Gilman is with the School of Natural and Computational Sciences, Massey University, Auckland 0745, New Zealand (e-mail: a.gilman@massey.ac.nz).

N. Vasconcelos, P. C. Cosman, and L. B. Milstein are with the Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, CA 92093 USA (e-mail: nuno@ucsd.edu; pcosman@ucsd.edu; lmilstein@ucsd.edu).

Digital Object Identifier 10.1109/LWC.2019.2940579

TABLE I
DEEP SENSING NEURAL NETWORK

Layer	Output dimensions	# of kernels	Kernel size
Input	$2 \times N$		
Conv1	$256 \times 2 \times N$	256	1×9
Conv2	$80 \times 2 \times N$	80	1×9
Dense1	256		
Dense2	2		
Output	1		

where $f(\mathbf{x}_i;\mathbf{w}) = p(y_i = 1|\mathbf{x} = \mathbf{x}_i;\mathbf{w})$ and the empirical risk uses the binary cross-entropy loss function

$$L[f(\mathbf{x}_i;\mathbf{w}), y_i] = -(y_i \log(f(\mathbf{x}_i;\mathbf{w})) + (1-y_i)\log(1-f(\mathbf{x}_i;\mathbf{w}))). \tag{2}$$

This is the set of network parameters that maximizes the likelihood $\prod_{i=1}^{n} f(\mathbf{x}_i;\mathbf{w})^{y_i}(1-f(\mathbf{x}_i;\mathbf{w}))^{1-y_i}$.

CNN was chosen as the neural network architecture in this letter for several reasons. First, it is a natural structure to consider because operation of a CNN kernel can be thought of as related to filtering operations that occur in communications receivers. Also, the modulation recognition work by O'Shea *et al.* [7] used a CNN. Further, in comparing a CNN with a fully connected (FC) network, it was found that the FC network has worse sensing performance than a CNN, and when we considered a CNN and a recurrent neural network (RNN) with identical sensing performance, the CNN had lower computational complexity than the RNN.

To compare the performance of spectrum sensing using DL, we adopt a setting where an analytical expression for the optimal sensing algorithm is available. We consider detecting a narrowband Gaussian-distributed signal in additive white Gaussian noise (AWGN), in which case the optimal sensing algorithm according to the log-likelihood ratio is[11]

$$LLR(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T(C_{\mathbf{z}}^{-1} - C_{\mathbf{x}}^{-1})\mathbf{x} \tag{3}$$

where $\mathbf{x}$ is a vector of received samples within one sensing duration, $C_{\mathbf{x}}$ is the covariance matrix of $\mathbf{x}$, and $C_{\mathbf{z}}$ is the covariance matrix of the additive noise after the filter.

We compare sensing performance using a narrowband Gaussian PU signal with zero mean, corrupted by AWGN. There are $N = 32$ samples in a sensing interval, and the signal-to-noise ratio (SNR) $10\log_{10}(\sigma_S^2/\sigma_n^2)$ is -4dB, where σ_S^2 is the PU signal variance and σ_n^2 is the noise variance after the filter. The PU signal bandwidth is 1/4 of the filter bandwidth. The network is trained with a training set D of $n = 2 \times 10^4$ and tested on an independent (but with the same transmitter, channel and receiver characteristics) test set of the same size. Fig. 1 shows the ROC curves for optimal and deep sensing as well as the performance of an energy detector [2]. The optimal sensing result was obtained with (3). The deep sensing result was obtained by computing probabilities of detection and false alarm on the test set, using different thresholds on the network output. The deep sensing, which does not require feature extraction of the received samples, outperforms energy detection (ED) and is close to the optimal.

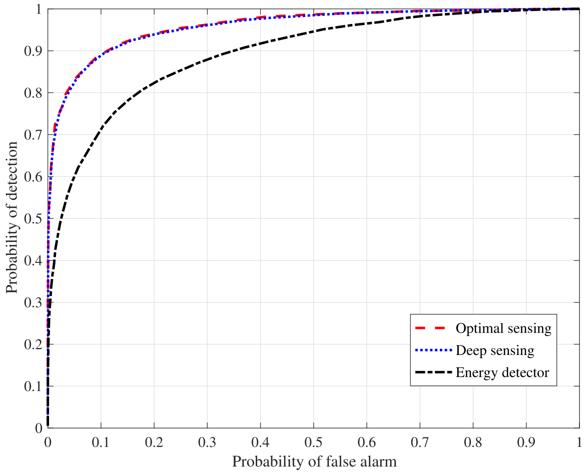

Fig. 1. Deep spectrum sensing compared with optimal sensing.

The optimal scheme for a particular sensing scenario is only optimal if it has perfect information on the required parameters. For example, the optimal scheme in Fig. 1 requires the covariance matrices of the received samples and of the additive noise after the receive filter. With estimation error in the required information, the performance degrades. Also, for different sensing scenarios, the optimal sensing scheme differs, so a dedicated sensing receiver is required for every scenario, which is costly. Furthermore, the optimal sensing for modulated signals typically requires averaging over a very large number of realizations of transmitted symbols, and thus, it is computationally infeasible.

III. ROBUST DEEP SENSING WITH TRANSFER LEARNING

Robustness was shown to be a problem when applying DL for automatic modulation recognition [12]. We examine deep sensing robustness by considering different PU signals: narrowband Gaussian signals with zero mean in AWGN with an SNR of -4dB, and QPSK signals that use a square root raised cosine filter with a roll-off factor of 0.5 for pulse shaping. The QPSK signals experience path loss with average SNR between -2dB and -4dB and frequency-selective Rayleigh fading with 3 discrete paths. The data is obtained from simulations in MATLAB. Datasets collected under these different characteristics will belong to different, but related, distributions, i.e., domains. The *source* domain is used to train the network, and the *target* domain is used for testing. Both training and test sets have size $n = 2 \times 10^4$. Results are in Fig. 2, where the probability of detection (p_d) versus the probability of false alarm (p_{fa}) is plotted. In Fig. 2(left), we use QPSK as source domain and Gaussian as target domain. The resulting sensing performance, marked "QPSK→Gaussian", is significantly worse than the case where we use 2×10^4 examples of Gaussian signals to train and test the network (curve labeled "Gaussian→Gaussian").

Similar observations can be made from Fig. 2(right), where the curve "Gaussian→QPSK" is obtained using Gaussian signals in the source domain and QPSK signals in the target domain, and the curve "QPSK→QPSK" is plotted for

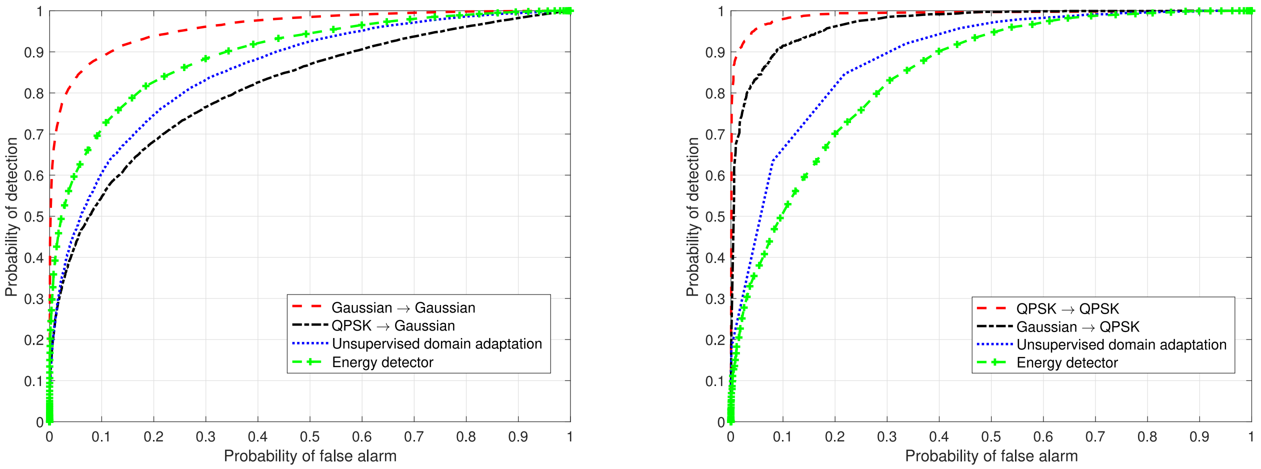

Fig. 2. Deep sensing using transfer learning with no labeled data: (left) from QPSK to zero-mean narrowband Gaussian signals; (right) from zero-mean narrowband Gaussian to QPSK signals.

reference. Figs. 1 and 2 show that when source and target domains are the same, deep sensing performance can be close to optimal, whereas when they are mismatched, deep sensing performance can degrade significantly. As transmitted signals can vary in several ways (e.g., alphabet sizes, coding schemes) and signal propagation depends on many factors (e.g., frequency, terrain profile), getting enough ground-truth labeled training data across all possible scenarios is difficult. Thus TL procedures are important.

A. Transfer Learning With no Labeled Data

The transfer approaches in this category are referred to as unsupervised domain adaptation. Let $X_{src} = \{\mathbf{x}_{src_i}\}$ and $X_{tar} = \{\mathbf{x}_{tar_i}\}$ denote the data in the source and target domains, respectively. As shown above, directly applying the neural network (NN) trained with X_{src} may not work well for X_{tar}. To leverage the knowledge learned by the NN from X_{src}, we use the TL method of [13]. This aims to discover a latent space described by a kernel-induced feature transformation function ϕ such that the marginal distributions of $\phi(X_{src})$ and $\phi(X_{tar})$ are close. A nonparametric distance estimate, referred to as the Maximum Mean Discrepancy (MMD)[13], is defined by embedding distributions in a reproducing kernel Hilbert space (RKHS) and is calculated by $\|\frac{1}{n_1}\sum_{i=1}^{n_1}\phi(\mathbf{x}_{src_i}) - \frac{1}{n_2}\sum_{i=1}^{n_2}\phi(\mathbf{x}_{tar_i})\|^2_{\mathcal{H}}$, where $\|\cdot\|_{\mathcal{H}}$ is the RKHS norm. Making the distributions of the source and target data close is equivalent to minimizing the MMD distance [13]. Let $\mathbf{K} = [\phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)]$, and $\mathbf{L}_{i,j} = 1/n_1^2$ if $\mathbf{x}_i, \mathbf{x}_j \in X_{src}$, else $\mathbf{L}_{i,j} = 1/n_2^2$ if $\mathbf{x}_i, \mathbf{x}_j \in X_{tar}$, otherwise, $\mathbf{L}_{i,j} = -1/n_1 n_2$. The MMD distance can then be written as $\mathrm{tr}(\mathbf{KL})$, and the learning problem formulated as [13]

$$\begin{aligned} \min_{\mathbf{W}} \ & \mathrm{tr}(\mathbf{W}^T\mathbf{KLKW}) + \mu \cdot \mathrm{tr}(\mathbf{W}^T\mathbf{W}) \\ s.t. \ & \mathbf{W}^T\mathbf{KHKW} = \mathbf{I} \end{aligned} \tag{4}$$

where $\mathrm{tr}(\cdot)$ stands for the trace operation, $\mathbf{H} = \mathbf{I} - (1/(n_1 + n_2))\mathbf{1}\mathbf{1}^T$ is the centering matrix, $\mathbf{1}$ is a $(n_1 + n_2) \times 1$ column vector with all 1's, a regularization term $\mathrm{tr}(\mathbf{W}^T\mathbf{W})$ controls the complexity of $\mathbf{W}$, $\mu > 0$ is a tradeoff factor between the MMD distance between distributions and complexity, and $\mathbf{I}$ is the identity matrix. The data in the latent space is W^TK, and the solution of W corresponds to the m ($m \leq N$) leading eigenvectors of $(\mathbf{KLK} + \mu\mathbf{I})^{-1}\mathbf{KHK}$.

We use p_{fa} and p_d as the sensing performance metrics. Fig. 2(left) shows that when QPSK data is used as source data and Gaussian data as target data, the TL algorithm improves the sensing, compared to when we directly use the NN trained on QPSK data for sensing Gaussian PU signals. However, the improved deep sensing is still worse than ED. Further, interchanging source and target data, Fig. 2(right) shows that unsupervised domain adaptation does not improve performance, although in this case both deep sensing outperform ED. These results indicate that this transfer with no labeled target domain data is not robust.

B. Transfer Learning With a Small Amount of Labeled Data

When we have a small amount of labeled data, we can use fine-tuning, the dominant TL procedure in computer vision [10]. The deep sensing system, trained on a large source dataset, is a starting point for further training using data from the target dataset. For training the baseline network, it is assumed that simulation data is used. For the TL, we use simulation data also, but in practice the SU would need to acquire some real labeled data in its actual environment. One way to accomplish this is through cooperation between PUs and SUs. With a small loss of throughput, the PUs could use occasional sensing intervals for providing ON and OFF periods so that each SU can acquire labeled data. Alternatively, by listening and comparing across consecutive sensing and data transmission intervals, an SU could develop estimates of the labels. Note that the surrounding environment can change during the collection of training examples for fine tuning, and the estimates of the labels can be inaccurate.

For each sensing interval, the PU and the SU are assumed to be randomly located uniformly within a 1000m × 1000m square area, and the path loss between them is calculated using the Frii transmission formula [14]. We start with a NN

TABLE II
DEEP SENSING PERFORMANCE (AREA UNDER CURVE) FOR VARIOUS SIGNALS AND CHANNEL MODELS. IN THIS TABLE, PL AND R DENOTE PATH LOSS AND RAYLEIGH FADING

Source domain → target domain	Fine-tune	Train from scratch
BPSK +PL → QPSK +PL,R	845.64	673.98
QPSK +PL,R → BPSK +PL	938.72	849.61
QPSK +PL → 16QAM +PL,R	816.55	655.63
16QAM +PL → BPSK +PL,R	870.26	760.05

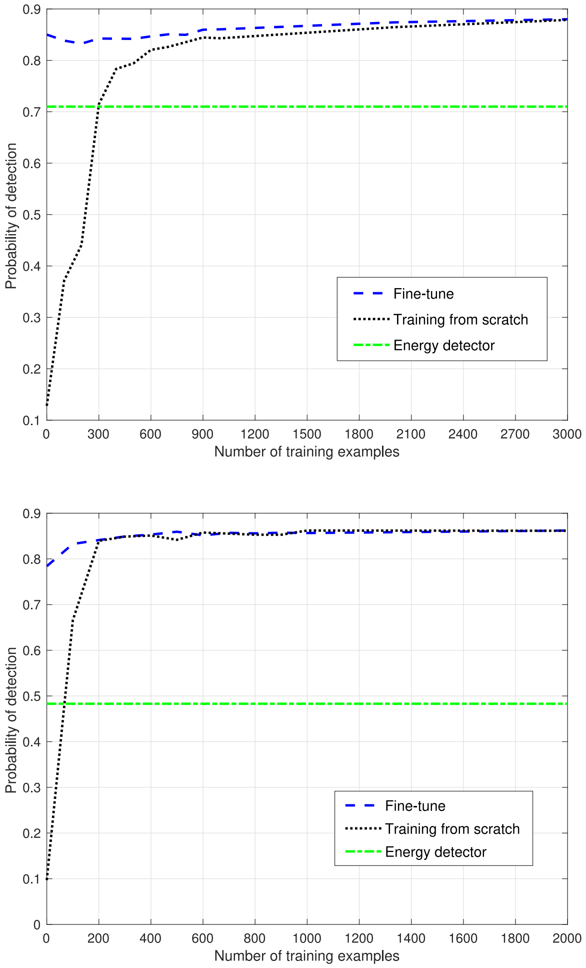

Fig. 3. Deep sensing performance with fine tuning: (top) from QPSK to zero-mean narrowband Gaussian signals; (bottom) from zero-mean narrowband Gaussian to QPSK signals.

pre-trained using 2×10^4 examples of QPSK data, and fine tune it using a variable number of examples of Gaussian signals. The tuned network is applied for sensing zero-mean Gaussian signals. We also plot the performance of ED and of DL-based sensing by training from scratch, which initializes the NN randomly and trains it using a variable number of Gaussian examples. As the stochastic gradient descent optimization uses random weight initialization, the network is trained 10 times and the results are averaged. Fig. 3(top) shows p_d vs. the number of examples of Gaussian signals, with $p_{fa} = 0.1$. With no labeled Gaussian data, $p_d > 0.80$ for the network trained by QPSK data, and p_d is around 0.20 for the randomly initialized network, showing that QPSK-trained initialization is beneficial, and the DL-based sensing outperforms ED. Note also that fine tuning outperforms training from scratch. Given enough training data, the performance of random initialization approaches that of the pre-trained network.

Next we interchange the training and test data, pre-training with Gaussian signals and fine tuning with QPSK signals. We test on QPSK signals, and average over 10 runs. Fig. 3(bottom) shows a similar pattern as before: when only a small amount of QPSK training data is available, better performance is achieved by fine tuning than by random initialization. Further, fine tuning outperforms ED for the whole curve, and the DL-based sensing by training from scratch outperforms ED as well when the number of training examples exceeds roughly 100.

In addition to the narrowband Gaussian and QPSK signals, we tested several other signals and channel models. For curves of the type shown in Fig. 3, the area under the curves over the x-axis range [0, 1000] for both fine-tuning and training from scratch are in Table II. All results were consistent with Fig. 3, in that fine-tuning outperformed training from scratch.

IV. CONCLUSION

We demonstrate the application of DL to spectrum sensing. The approach does not require feature extraction from the received signals at the SU. As deep spectrum sensing is not robust when applied in a different communications scenario from the training data, we incorporate TL to ensure robustness. With no labeled target data, the transfer is unreliable and depends on whether QPSK or Gaussian signals are the source or target. When there is a small amount of labeled target data, fine tuning is robust for transferring into a variety of domains.

REFERENCES

[1] Y. Zeng, Y.-C. Liang, A. T. Hoang, and R. Zhang, "A review on spectrum sensing for cognitive radio: Challenges and solutions," *EURASIP J. Adv. Signal Process.*, vol. 2010, Dec. 2010, Art. no. 381465. doi: 10.1155/2010/381465.

[2] D. Cabric, S. M. Mishra, and R. G. Brodersen, "Implementation issues in spectrum sensing for cognitive radios," in *Proc. 38th Asilomar Conf. Signals Syst. Comput.*, Nov. 2004, pp. 772–776.

[3] Y. Li and Q. Peng, "Achieving secure spectrum sensing in presence of malicious attacks utilizing unsupervised machine learning," in *Proc. IEEE MILCOM*, Baltimore, MD, USA, Nov. 2016, pp. 174–179.

[4] M. Bojarski *et al.*, "End to end learning for self-driving cars," *arXiv preprint, arXiv: 1604.07316*, 2016.

[5] Y. Wu *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint, arXiv: 1609.08144*, 2016.

[6] W. Lee, M. Kim, and D. Cho, "Deep cooperative sensing: Cooperative spectrum sensing based on convolutional neural networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 3005–3009, Mar. 2019.

[7] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-air deep learning based radio signal classification," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 168–179, Feb. 2018.

[8] K. Davaslioglu and Y. E. Sagduyu, "Generative adversarial learning for spectrum sensing," in *Proc. IEEE Int. Conf. Commun.*, Kansas City, MO, USA, May 2018, pp. 1–6.

[9] T. Erpek, Y. E. Sagduyu, and Y. Shi, "Deep learning for launching and mitigating wireless jamming attacks," *IEEE Trans. Cogn. Commun. Netw.*, vol. 5, no. 1, pp. 2–14, Mar. 2019.

[10] J. Donahue *et al.*, "DeCAF: A deep convolutional activation feature for generic visual recognition," in *Proc. 31st Int. Conf. Mach. Learn.*, 2014, pp. 647–655.

[11] T. A. Schonhoff and A. A. Giordano, *Detection and Estimation Theory and Its Applications*. Upper Saddle River, NJ, USA: Prentice-Hall, 2006.

[12] B. Luo, Q. Peng, P. C. Cosman, and L. B. Milstein, "Robustness of deep modulation recognition under AWGN and Rician fading," in *Proc. Asilomar Conf. Signals Syst. Comput.*, Oct. 2018, pp. 447–450.

[13] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Netw.*, vol. 22, no. 2, pp. 199–210, Feb. 2011.

[14] T. S. Rappaport, *Wireless Communications: Principles & Practice*. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.