Unsupervised Radio Scene Analysis Using Neural Expectation Maximization

Hao Chen and Seung-Jun Kim
Dept. of Computer Science & Electrical Engineering
University of Maryland, Baltimore County
Baltimore, MD 21250
E-mail: {chenhao1,sjkim}@umbc.edu

MILCOM 2022 - 2022 IEEE Military Communications Conference (MILCOM) | 978-1-6654-8534-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/MILCOM55135.2022.10017594

***Abstract*—An unsupervised learning-based blind RF scene analysis method is proposed. The method can analyze a complex radio scene containing a mixture of different transmission types and estimate the constituent signals with associated channel vectors from multi-antenna measurements. A deep neural network is trained to learn the unique time-frequency patterns of various signal types. The channels, noise powers, and encodings input to the neural network are estimated in a maximum likelihood framework via an expectation-maximization algorithm. Numerical tests using scenes constructed from real RF measurements verify the effectiveness of the proposed method.**

I. Introduction

Analyzing the activities in the radio frequency (RF) spectrum is critical for efficient spectrum sharing, interference mitigation, RF intelligence, and security assurance in wireless networks. In cognitive radio (CR) systems, estimation of spectrum occupancy in the time, frequency, and space domains is a key prerequisite for opportunistic dynamic spectrum utilization. In tactical scenarios, a wide bandwidth of spectrum over an operation area must be continuously monitored so that suspicious RF transmissions can be quickly identified and acted upon.

The spectrum sensing techniques developed for CRs often rely on the signal strength measurements, known pilot signals, or the 2nd-/higher-order statistics of the received signals to perform binary hypothesis testing and determine spectrum occupancy [1]. Therefore, the methods either require significant a priori information on the signal formats, or perform rather poorly in the presence of the detrimental effects of the wireless channel. Furthermore, they cannot effectively analyze the complex radio scenes containing multiple transmitters sharing the spectrum even in a non-orthogonal fashion.

Existing methods on analyzing the RF scenes with concurrent transmissions are rather limited. A robust feature extraction technique exploiting a priori knowledge of transmission protocols was studied [2]. Blind source separation (BSS) techniques based on subspace structures were employed in the CR setups [3], [4]. The power spectrum of the received signal was clustered over time and the 4th-order spectrum was analyzed using three-way tensor decomposition in [5]. Mixed signals were classified using dictionaries trained on the single-transmitter scenes in [6]. However, the unique time-frequency patterns of the constituent signals have not been fully exploited in a machine learning framework.

In recent years, deep learning techniques were shown to achieve remarkable performance in various applications including signal/image analysis, natural language processing, and dynamic decision making. Unfortunately, training deep neural networks often requires a large amount of data, which may be costly to collect, preprocess, and find labels for. In particular, a naive "black box" approach that does not carefully exploit the problem structure may require a very large training set to capture diverse underlying factors in play.

Our approach is to adopt unsupervised learning that does not require labels for the radio scenes. The mixture signals are processed in the time-frequency domain in a blind fashion without any prior knowledge on the signal characteristics or transmission patterns. Inspired by the related works in the audio signal processing [7], [8] and image classification applications [9], we achieve good performance with high data efficiency by judiciously combining a proven signal processing framework with a powerful deep learning technique. Specifically, a deep neural network is trained to model different signal types and their unique time-frequency transmission patterns, while a maximum likelihood (ML) parameter estimation is employed to estimate the multi-antenna channels, noise variance, and the encodings for the neural networks. The resulting algorithm performs the expectation-maximization (EM) iterations per RF signal snapshot, while training the neural network parameters over the entire data set.

The rest of the paper is organized as follows. The signal model and the problem statement are provided in Sec. II. The parameter estimation algorithm is derived in the EM framework in Sec. III. The overall training of the neural network is discussed in Sec. IV. The numerical test results are presented in Sec. V. Conclusions and future research directions are provided in Sec. VI.

II. Signal Model and Problem Statement

Let $\mathbf{x}(t) \in \mathbb{C}^M, t = 1, \ldots, T$, be the downconverted and sampled measurement vectors of the RF signal received using M antennas. The RF signal contains transmissions from J transmitters. For simplicity, it is assumed that the

Authorized licensed use limited to: UNIV OF CALIFORNIA. Downloaded on January 28,2023 at 20:50:59 UTC from IEEE Xplore. Restrictions apply.

transmitters have single antennas and the channel is frequency-nonselective. Upon denoting the signal transmitted at time t from the j-th transmitter as $s_j(t)$, the channel between the j-th transmitter and the receiver as $\mathbf{h}_j \in \mathbb{C}^M$, and the receiver noise vector at time t as $\mathbf{z}(t)$, $\mathbf{x}(t)$ can be expressed as

$$\mathbf{x}(t) = \sum_{j=1}^{J} \mathbf{h}_j s_j(t) + \mathbf{z}(t),\ t = 1, \ldots, T. \tag{1}$$

Define $\mathbf{H} := [\mathbf{h}_1, \ldots, \mathbf{h}_J]$ and $\mathbf{s}(t) := [s_1(t), \ldots, s_J(t)]^\top$, where $^\top$ represents transposition. Then, (1) can be compactly written as

$$\mathbf{x}(t) = \mathbf{H}\mathbf{s}(t) + \mathbf{z}(t),\ t = 1, \ldots, T. \tag{2}$$

Our goal is to estimate $\{s_j(t)\}_j$ and $\{\mathbf{h}_j\}_j$ from $\mathbf{x}(t)$ using signal processing and machine learning methods. Specifically, it is postulated that for each j, transmission $s_j(t)$ belongs to a distinct type with unique features, such as Wi-Fi or Bluetooth transmissions. These patterns are to be learned using machine learning. However, instead of using a set of training waveforms for *each* signal type, it is desired that a training set of *mixture* waveforms is utilized. Thus, the algorithm can learn from samples collected in a densely interfered RF environment in an unsupervised fashion, without requiring much manual labeling and preprocessing of measurements. On the other hand, the channels can be estimated through signal processing techniques. This approach is advantageous since it obviates the need to prepare a data set that involves diverse channel realizations, which is critical if the channels were to be estimated using machine learning methods.

The patterns of different signal types can be extracted in the time-frequency domain. Let $\mathbf{x}(n,f) \in \mathbb{C}^M$, $\mathbf{s}(n,f) \in \mathbb{C}^J$, and $\mathbf{z}(n,f) \in \mathbb{C}^M$, with $n = 1, \ldots, N$ and $f = 1, \ldots, F$, be the short-time Fourier transforms (STFTs) of $\mathbf{x}(t)$, $\mathbf{s}(t)$, and $\mathbf{z}(t)$, respectively. To derive a ML estimator, probabilistic assumptions are made on the signals. Specifically, it is assumed that $\mathbf{s}(n,f)$ is complex Gaussian with mean $\mathbf{0}$ and covariance $\mathbf{R}_\mathbf{s}(n,f)$, which is a $J \times J$ diagonal matrix with $v_1(n,f), \ldots, v_J(n,f)$ on the diagonal. Also, it is assumed that $s_j(n,f)$ is statistically independent of $s_{j'}(n',f')$ unless $j = j'$, $n = n'$, and $f = f'$. Then, it is straightforward to verify that $\mathbf{x}(n,f)$ is independent across n and f, and has a complex Gaussian distribution with mean $\mathbf{0}$ and covariance

$$\mathbf{R}_\mathbf{x}(n,f) := \mathbf{H}\mathbf{R}_\mathbf{s}(n,f)\mathbf{H}^H + \mathbf{R}_\mathbf{z} \tag{3}$$

where $\cdot^H$ denotes Hermitian transpose and $\mathbf{R}_\mathbf{z}$ the diagonal covariance matrix of $\mathbf{z}(n,f)$.

To capture the time-frequency patterns in the j-th transmitted signal, matrix $\mathbf{V}_j \in \mathbb{R}_+^{N \times F}$, whose (n,f)-entry is $v_j(n,f)$, is modeled via a deep neural network $f_{\boldsymbol{\phi}}$ with parameter vector $\boldsymbol{\phi}$ as

$$\mathbf{V}_j = f_{\boldsymbol{\phi}}(\boldsymbol{\gamma}_j). \tag{4}$$

Here, $\boldsymbol{\gamma}_j$ is the input to the neural network, with a much lower dimension than that of $\mathbf{V}_j$, which is NF. $\boldsymbol{\gamma}_j$ not only encodes the differences in the signal types, but also any variabilities in the patterns within each signal type. $\{\gamma_j\}$ are estimated automatically in a ML framework as explained in Sec. III.

III. EM-Based Parameter Estimation

Suppose for now that the neural network parameter $\boldsymbol{\phi}$ is fixed. We wish to estimate the unknown parameters $\boldsymbol{\theta} := (\{\mathbf{h}_j\}, \{\boldsymbol{\gamma}_j\}, \mathbf{R}_\mathbf{z})$ based on the given time-frequency snapshot $\mathbf{X} := \{\mathbf{x}(n,f)\}_{n,f}$. This can be done in a ML framework by maximizing w.r.t. $\boldsymbol{\theta}$ the log-likelihood

$$\log p(\mathbf{X}; \boldsymbol{\theta}, \boldsymbol{\phi}) = -\sum_{n,f} \big[\log |\pi \mathbf{R}_\mathbf{x}(n,f)| + \mathbf{x}(n,f)^H \mathbf{R}_\mathbf{x}^{-1}(n,f)\mathbf{x}(n,f)\big] \tag{5}$$

where $|\cdot|$ denotes the determinant. However, this optimization is quite challenging. Instead, one can adopt the EM framework to solve the problem more efficiently [7], [9].

Let $\mathbf{S}_j := \{s_j(n,f)\}_{n,f}$ for $j = 1, \ldots, J$, where $s_j(n,f)$ is the j-th entry of $\mathbf{s}(n,f)$. Define also $\mathbf{S} := \{\mathbf{S}_j\}_j$. With $\mathbf{S}$ not observed, the EM algorithm aims at maximizing a lower bound of (5) based on the current k-th iterate of $\boldsymbol{\theta}$, denoted as $\boldsymbol{\theta}^{(k)}$. This lower bound is given by

$$Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(k)}) := \mathbb{E}_{\mathbf{S}|\mathbf{X};\boldsymbol{\theta}^{(k)}} \{\log p(\mathbf{X}, \mathbf{S}; \boldsymbol{\theta})\} \tag{6}$$

where the dependence on $\boldsymbol{\phi}$ is suppressed for brevity. Now define

$$\mathbf{W}(n,f) := \mathbf{R}_\mathbf{s}(n,f)\mathbf{H}^H\mathbf{R}_\mathbf{x}(n,f)^{-1} \tag{7}$$

$$\hat{\mathbf{s}}(n,f) := \mathbf{W}(n,f)\mathbf{x}(n,f) \tag{8}$$

$$\mathbf{C}_{\mathbf{s}|\mathbf{x}}(n,f) := (\mathbf{I} - \mathbf{W}(n,f)\mathbf{H})\mathbf{R}_\mathbf{s}(n,f) \tag{9}$$

$$\mathbf{R}_{\mathbf{ss}|\mathbf{x}}(n,f) := \mathbf{C}_{\mathbf{s}|\mathbf{x}}(n,f) + \hat{\mathbf{s}}(n,f)\hat{\mathbf{s}}(n,f)^H. \tag{10}$$

Then $Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(k)})$ can be expressed as

$$\sum_{n,f} \Big[-\log|\mathbf{R}_\mathbf{s}(n,f)| - \mathrm{tr}\Big\{\mathbf{R}_{\mathbf{ss}|\mathbf{x}}(n,f;\boldsymbol{\theta}^{(k)})\mathbf{R}_\mathbf{s}(n,f)^{-1}\Big\} - \log|\mathbf{R}_\mathbf{z}| - \mathbf{x}(n,f)\mathbf{R}_\mathbf{z}^{-1}\mathbf{x}(n,f)^H + 2\mathrm{Re}\Big\{\mathbf{x}(n,f)^H\mathbf{R}_\mathbf{z}^{-1}\mathbf{H}\hat{\mathbf{s}}(n,f;\boldsymbol{\theta}^{(k)})\Big\} - \mathrm{tr}\Big\{\mathbf{R}_{\mathbf{ss}|\mathbf{x}}(n,f;\boldsymbol{\theta}^{(k)})\mathbf{H}^H\mathbf{R}_\mathbf{z}^{-1}\mathbf{H}\Big\}\Big] \tag{11}$$

where it is emphasized in $\mathbf{R}_{\mathbf{ss}|\mathbf{x}}(n,f;\boldsymbol{\theta}^{(k)})$ and $\hat{\mathbf{s}}(n,f;\boldsymbol{\theta}^{(k)})$ that the quantities must be computed through (7)–(10) using parameters $\boldsymbol{\theta}^{(k)}$. The bound is maximized w.r.t. $\boldsymbol{\theta}$ to obtain the next iterate $\boldsymbol{\theta}^{(k+1)}$. In particular, to find $\mathbf{H}^{(k+1)}$ and $\mathbf{R}_\mathbf{z}^{(k+1)}$, the partial derivatives of $Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(k)})$ w.r.t. $\mathbf{H}$ and $\mathbf{R}_\mathbf{z}$ are set to zero. This results in the updates

$$\mathbf{H}^{(k+1)} = \hat{\mathbf{R}}_{\mathbf{x}\hat{\mathbf{s}}}\bar{\mathbf{R}}_{\mathbf{ss}|\mathbf{x}}^{-1} \tag{12}$$

$$\mathbf{R}_\mathbf{z}^{(k+1)} = \mathrm{diag}\Big\{\hat{\mathbf{R}}_{\mathbf{xx}} - \mathbf{H}^{(k)}\hat{\mathbf{R}}_{\mathbf{x}\hat{\mathbf{s}}}^H - \hat{\mathbf{R}}_{\mathbf{x}\hat{\mathbf{s}}}\mathbf{H}^{(k)H} + \mathbf{H}^{(k)}\bar{\mathbf{R}}_{\mathbf{ss}|\mathbf{x}}\mathbf{H}^{(k)H}\Big\} \tag{13}$$

Authorized licensed use limited to: Univ of Calif Davis. Downloaded on January 28,2023 at 20:50:59 UTC from IEEE Xplore. Restrictions apply.

where

$$\hat{\mathbf{R}}_{\mathbf{x}\hat{\mathbf{s}}} := \frac{1}{NF}\sum_{n,f}\mathbf{x}(n,f)\hat{\mathbf{s}}(n,f;\boldsymbol{\theta}^{(k)})^H \tag{14}$$

$$\bar{\mathbf{R}}_{\mathbf{ss}|\mathbf{x}} := \frac{1}{NF}\sum_{n,f}\mathbf{R}_{\mathbf{ss}|\mathbf{x}}(n,f;\boldsymbol{\theta}^{(k)}) \tag{15}$$

$$\hat{\mathbf{R}}_{\mathbf{xx}} := \frac{1}{NF}\sum_{n,f}\mathbf{x}(n,f)\mathbf{x}(n,f)^H \tag{16}$$

and $\mathrm{diag}\{\mathbf{M}\}$ is a diagonal matrix of the same size as $\mathbf{M}$ with the diagonal entries equal to those in $\mathbf{M}$.

For updating $\{\boldsymbol{\gamma}_j\}$, a closed-form update rule cannot be derived. Instead, a simple gradient descent method can be employed. That is, the following update is repeated L times starting with $\boldsymbol{\gamma}_j^{[0]} = \boldsymbol{\gamma}_j^{(k)}$.

$$\boldsymbol{\gamma}_j^{[\ell+1]} = \boldsymbol{\gamma}_j^{[\ell]} + \eta_1 \frac{\partial Q(\boldsymbol{\theta};\boldsymbol{\theta}^{(k)})}{\partial \boldsymbol{\gamma}_j}\bigg|_{\boldsymbol{\gamma}_j=\boldsymbol{\gamma}_j^{[\ell]}}, \ \ \ell = 0,\ldots,L-1 \tag{17}$$

where $\eta_1 > 0$ is a setp size, and the derivative is obtained from (11) together with the back-propagation (chain rule) applied to (4). Then, $\boldsymbol{\gamma}_j^{(k+1)}$ is set to $\boldsymbol{\gamma}_j^{[L]}$, for $j = 1,\ldots,J$.

IV. Overall Algorithm

While $\boldsymbol{\theta}$ is estimated for a given snapshot $\mathbf{X}$ using the EM algorithm described in Sec. III, the neural network parameter $\boldsymbol{\phi}$ is learned from the entire data set $\{\mathbf{X}[i]\}_{i=1}^{I}$. Note that the same neural network $f_{\boldsymbol{\phi}}$ with a shared parameter vector $\boldsymbol{\phi}$ is used for all J transmissions, which can improve the sample efficiency for training. The log-likelihood in (5) is used for the objective function for training. Thus, $\boldsymbol{\phi}$ is updated using the b-th mini-batch of samples $\{\mathbf{X}[i]\}_{i\in\mathcal{I}_b}$ as

$$\boldsymbol{\phi}^{[b+1]} = \boldsymbol{\phi}^{[b]} + \frac{\eta_2}{|\mathcal{I}_b|}\sum_{i\in\mathcal{I}_b}\frac{\partial \log p(\mathbf{X}[i];\boldsymbol{\theta}[i],\boldsymbol{\phi})}{\partial\boldsymbol{\phi}}\bigg|_{\boldsymbol{\phi}=\boldsymbol{\phi}^{[b]}} \tag{18}$$

where $\eta_2 > 0$ is a step size, and $\boldsymbol{\theta}[i]$ is the $\boldsymbol{\theta}$ estimated from the EM algorithm for snapshot $\mathbf{X}[i]$. The overall training algorithm is described in pseudocode in Table I.

Since the optimization problem is highly nonconvex, it is important to initialize the algorithm properly. In particular, in the early stage of training, the EM algorithm must produce good estimates to guide the neural network training. For this, similarly to [7], the channel matrix $\mathbf{H}$ was initialized to a rough estimate obtained from a clustering algorithm applied to $\mathbf{X}[i]$. The underlying assumption is that for each time-frequency bin, there is a dominant transmission. Parameters $\{\boldsymbol{\gamma}_j\}$ and $\boldsymbol{\phi}$ are initialized randomly.

Once the training is complete, for a new snapshot $\mathbf{X}[I+1]$, the EM algorithm in lines 4–14 in Table I can be executed using the trained parameter $\boldsymbol{\phi}^*$. Upon convergence of the EM loop, $\{\hat{\mathbf{s}}(n,f) = [\hat{s}_1(n,f),\ldots,\hat{s}_J(n,f)]^\top\}_{n,f}$ contains the estimated individual transmission signals and $\mathbf{H}^{(\infty)} = [\hat{\mathbf{h}}_1,\ldots,\hat{\mathbf{h}}_J]$ the corresponding channel estimates.

TABLE I: Overall training algorithm.

Input: $\{\mathbf{X}[i]\}$, J, η_1, η_2, and $\{\mathcal{I}_b\}$
Output: $\boldsymbol{\phi}^*$

1: Initialize $\boldsymbol{\phi}^{[0]}$ randomly
2: For $b = 0, 1, \ldots$ /* b-th mini-batch */
3: For $\mathbf{X}[i],\ i \in \mathcal{I}_b$
/* EM iterations */
4: Initialize $\mathbf{H}^{(0)}$ and $\{\boldsymbol{\gamma}_j^{(0)}\}$
5: For $k = 0, 1, \ldots$
/* E-step */
6: Compute $\hat{\mathbf{s}}(n,f)$, $\mathbf{C}_{\mathbf{s}|\mathbf{x}}(n,f)$, and $\mathbf{R}_{\mathbf{ss}|\mathbf{x}}(n,f)$ for all n, f via (8)–(10)
/* M-step */
7: Compute $\hat{\mathbf{R}}_{\mathbf{x}\hat{\mathbf{s}}}$, $\bar{\mathbf{R}}_{\mathbf{ss}|\mathbf{x}}$, and $\hat{\mathbf{R}}_{\mathbf{xx}}$ via (14)–(16)
8: Compute $\mathbf{H}^{(k+1)}$ and $\mathbf{R}_{\mathbf{z}}^{(k+1)}$ via (12) and (13)
9: Set $\boldsymbol{\gamma}_j^{[0]} = \boldsymbol{\gamma}_j^{(k)}$ for all $j = 1,\ldots,J$
10: For $\ell = 0, 1, \ldots, L-1$
11: Update $\{\boldsymbol{\gamma}_j^{[\ell]}\}$ per (17)
12: Next ℓ
13: Set $\boldsymbol{\gamma}_j^{(k+1)} = \boldsymbol{\gamma}_j^{[L]}$ for all j
14: Next k
15: Set $\boldsymbol{\theta}[i] = \left(\mathbf{H}^{(\infty)}, \{\boldsymbol{\gamma}_j^{(\infty)}\}, \mathbf{R}_z^{(\infty)}\right)$
16: Next i
/* Update neural network parameters $\boldsymbol{\phi}$ */
17: Update $\boldsymbol{\phi}^{[b]}$ per (18)
18: Next b
19: Set $\boldsymbol{\phi}^* = \boldsymbol{\phi}^{[\infty]}$

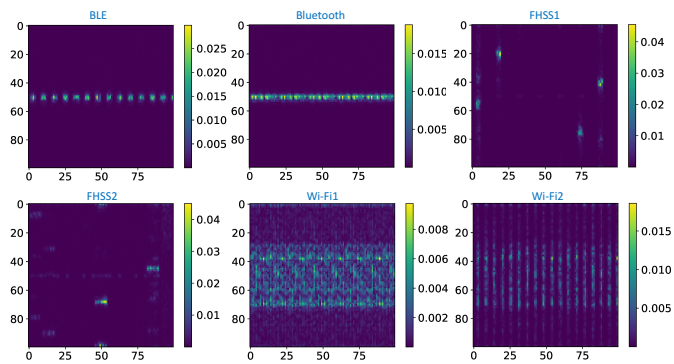


Fig. 1: Spectrograms for different types of transmissions.

V. Numerical Tests

A. Experiment Setup

We tested the proposed algorithm to the RF data set collected in the 2.4 GHz band using a software-defined radio from Ettus Research. The baseband complex signals of 40 MHz bandwidth were acquired inside a RF shield box. In total, 6 signal types were used, namely 2 types of Wi-Fi signals, Bluetooth (BT), Bluetooth Low Energy (BLE), and 2 types of frequency hopping spread spectrum (FHSS) transmissions [6]. The 2 types of Wi-Fi signals correspond to the Wi-Fi transmission in a high occupancy scenario (denoted as "Wi-Fi1" in the sequel), capturing intensive Wi-Fi usage such as downloading a large file, and in a low occupancy scenario (Wi-Fi2), representing a more sporadic use case. Two types of drone controllers generated unique proprietary FHSS waveforms, which we denote as "FHSS1" and "FHSS2". The

Authorized licensed use limited to: Univ of Calif Davis. Downloaded on January 28,2023 at 20:50:59 UTC from IEEE Xplore. Restrictions apply.

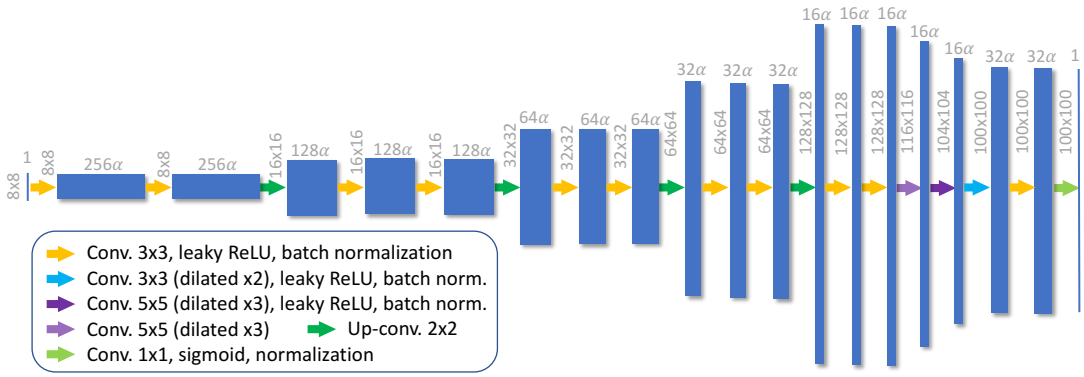


Fig. 2: Neural network architecture.

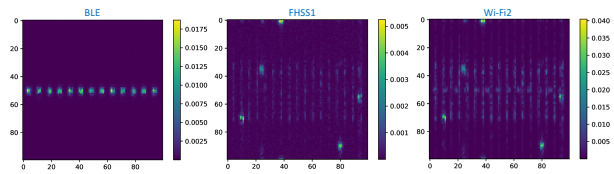
sample spectrograms of the six types of transmissions are shown in Fig. 1. The data set was collected using a single antenna. The multi-antenna mixture signals were synthetically generated by multiplying the steering vectors corresponding to suitable incident angles and summing multiple transmissions. $M = 3$ or 6 antennas were postulated. Each snapshot is 100 ms long, which was processed to obtain a STFT with $N = 100$ and $F = 100$. All transmissions are assumed to be of equal power. We used $I = 3,000$ snapshots for training, 100 for validation, and 100 for testing.

B. Neural Network Architecture

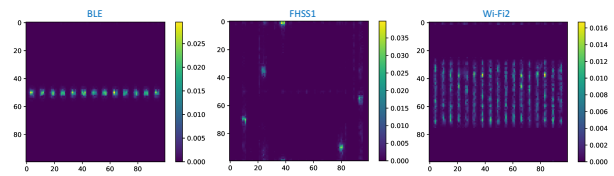
The architecture of the neural network used is depicted in Fig. 2, which is similar to the decoding part of U-Net [10]. An 8×8 matrix initialized randomly is used for the input $\boldsymbol{\gamma}_j$ to the network. The size was of $\boldsymbol{\gamma}_j$ was chosen from trial and error. The network eventually produces output $\mathbf{V}_j$, which is a 100×100 matrix. Each arrow in Fig. 2 represents a combination of a 2-D convolution operation, a nonlinear transformation, and normalization, as indicated in the legend. Each rectangular block represents the resulting tensor, which is a collection of matrices of the size indicated at the left side of each block, with the number of the matrices (channels) shown on top of each block. The parameter α was set to either 1 or 1.5 depending on J as explained in Sec. V-C, with larger α imparting more capacity to the network. All the normalizations are the batch normalization [11], except the normalization in the last layer, which simply scales so that the maximum entry in the output $\mathbf{V}_j$ is equal to 1. Such normalization is useful since there is scaling ambiguity in the estimate of $\mathbf{H}$ and $\mathbf{V}_j$; see (3).

C. Test Results

1) Case with $J = 3$: First, consider the case with $J = 3$ signal types (BLE, FHSS1 and Wi-Fi2) present in the mixture. The number of antennas is $M = 3$ as well. The incident angles of the signals are $\{15, 60, -45\}$ degrees. The neural network architecture in Fig. 2 with $\alpha = 1$ was used. To benchmark the performance of our neural EM (NEM) algorithm, an EM algorithm without employing the neural network is also tested. In the latter algorithm, only lines 4–14 in Table I are run,



(a) EM estimates



(b) NEM estimates

Fig. 3: Estimated spectrograms when $J = 3$.

with lines 9–13 replaced by a routine that estimates $v_j(n, f)$ simply from the j-th diagonal entry of $\mathbf{R}_{\mathbf{ss}|\mathbf{x}}(n, f)$ [7]. Thus, it neglects the patterns in the time-frequency domain. Fig. 3 shows the estimated spectrograms $|\hat{s}_j(n, f)|$ from the EM and the NEM methods. It can be seen that NEM method obtains the components that are closer to the ground truth spectrograms in Fig. 1. In particular, the FHSS1 and Wi-Fi2 spectrograms are seen to be not well separated in the EM results, compared to those from the NEM algorithm.

In Fig. 4, the average correlation coefficients between the true channels and the estimated channels are shown at various signal-to-noise power ratio (SNR) levels as the red curves with triangle markers. Additive white Gaussian noise was added to simulate the SNRs. The SNR equal to ∞ represents the case where no additional noise was added to the measurements. The solid curve corresponds to the NEM results and the dashed curve the EM ones. The averaging was done over 100 test snapshots as well as the three signal types. It can be seen that NEM obtains the channels that are more highly correlated to the ground truths than the EM algorithm at all SNR levels. In fact, the performance gap is seen to widen at lower SNR values. Similarly, the red curves with triangle markers in Fig. 5

Authorized licensed use limited to: Univ of Calif Davis. Downloaded on January 28,2023 at 20:50:59 UTC from IEEE Xplore. Restrictions apply.

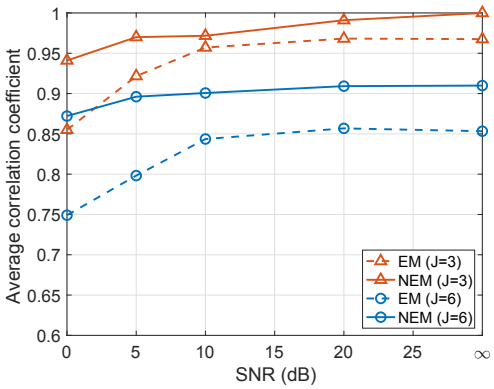


Fig. 4: Channel correlations between true and estimated values.

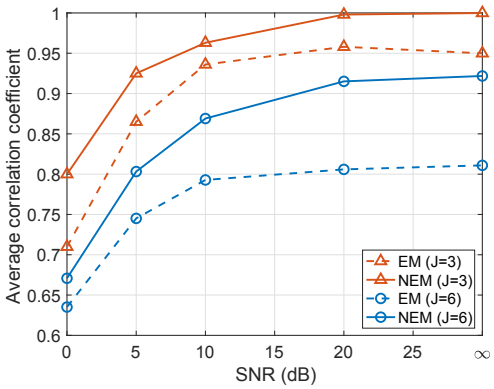
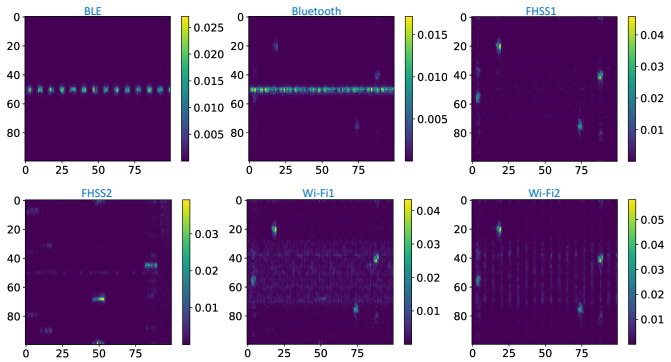
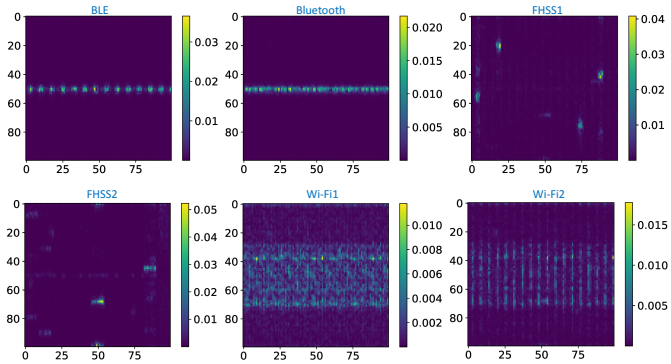


Fig. 5: STFT correlations between true and estimated values.



(a) EM estimates



(b) NEM estimates

Fig. 6: Estimated spectrograms when $J = 6$.

depict the average correlation coefficients between the true and the estimated STFTs. The NEM estimates are again observed to achieve higher correlations to the ground truth STFTs.

2) Case with $J = 6$: We also tested the case where all six signal types were mixed together. The directions of arrival for the BLE, BT, FHSS1, FHSS2, Wi-Fi1 and Wi-Fi2 transmissions were at $\{15, 45, 75, -15, -45, -75\}$ degrees, respectively. This case is more challenging than the $J = 3$ case discussed in Sec. V-C1, as some signal types possess strong similarities among them and the incident angles are also smaller. To increase the capacity of the neural network to cope with more diverse signal types, $\alpha = 1.5$ was used in the neural network in Fig. 2. Also, $M = 6$ antennas are employed.

An exemplar set of estimated spectrograms of the individual transmissions are shown in Fig. 6. In Fig. 6(a), the spectrograms from the EM algorithm without a neural network are depicted. By comparing them with those in Fig. 1, one can observe that the two Wi-Fi transmissions (Wi-Fi1 and Wi-Fi2) were not clearly estimated and strong interference from FHSS1 is present. The results from the NEM algorithm in Fig. 6(b) are seen to be much clearer, although some small interference from FHSS2 and Wi-Fi transmissions is visible in the FHSS1 spectrogram.

To see the robustness against the noise, the average correlation coefficients of the estimated channels and STFTs against the ground truths were again calculated at various SNR levels. The results from both the EM and the NEM algorithms are depicted in Fig. 4 and Fig. 5 in the blue curves with circle markers. As before, the NEM results are significantly better than the EM performances. Both algorithms experience degradation in the correlation due to the aforementioned challenges associated with mixtures with larger number of components.

3) Case with Mismatched J: To see how the proposed algorithm fares when the number J of the component signals is unknown, the actual number J of signals in the mixture is varied from 2 to 6. This gives rise to a total of $\sum_{J=2}^{6} \binom{6}{J} = 57$ combinations of possible mixtures. The value of J can be estimated in practice from, e.g., the eigen-analysis of $\hat{\mathbf{R}}_{\mathbf{xx}}$ or using more sophisticated methods [12]. Here, $J = 6$ is assumed by the algorithm for simplicity.

The experiments were conducted with 10 testing samples per combination. The resulting correlation coefficients for the STFTs and the channels are shown in Table II. It can be seen that when the true J matches the assumed $J = 6$, the performance of the EM and the NEM algorithms are the best or close to the best. On the other hand, the NEM performance degrades significantly when J is mismatched, although as J gets smaller, the performance is seen to rebound, as the estimation is done with fewer interferers. Still, the NEM is seen to be superior to the EM counterpart across all values of true J.

Authorized licensed use limited to: Univ of Calif Davis. Downloaded on January 28,2023 at 20:50:59 UTC from IEEE Xplore. Restrictions apply.

True J	2	3	4	5	6
EM	0.9216	0.8718	0.8729	0.8616	0.8344
NEM	0.9419	0.8919	0.8891	0.8772	0.9436

(a) Average correlation coefficients for STFTs

True J	2	3	4	5	6
EM	0.9209	0.8696	0.8703	0.8367	0.8316
NEM	0.9496	0.8898	0.8769	0.8558	0.9346

(b) Average correlation coefficients for channels

TABLE II: Correlation performances.

VI. Conclusions and Future Works

An unsupervised learning algorithm has been proposed that can automatically analyze a RF scene containing a mixture of various types of transmissions into constituent signals and associated channel vectors. A deep neural network was trained to capture the time-frequency patterns unique to different signal types. The channel vectors, the measurement noise covariance, and the embeddings input to the neural network were estimated in a ML estimation framework using the EM algorithm. Only unlabeled mixture signals are used for training, and the data set does not need to contain diverse realizations of channel vectors. The proposed algorithm was tested using mixture STFTs constructed from six different types of real RF signal measurements, and the performance was seen superior to that of an EM algorithm that does not employ neural networks. Joint estimation of the number of component signals, semi-supervised training using weak labels, and developing fully neural network-based algorithms for faster training and operation, are left for future work.

References

[1] E. Axell, G. Leus, E. G. Larsson, and H. V. Poor, "Spectrum sensing for cognitive radio: State-of-the-art and recent advances," *IEEE Sig. Process. Mag.*, vol. 29, no. 3, pp. 101–116, May 2012.

[2] S. Grimaldi, A. Mahmood, and M. Gidlund, "Real-time interference identification via supervised learning: Embedding coexistence awareness in IoT devices," *IEEE Access*, vol. 7, pp. 835–850, Jan. 2019.

[3] O. Duval, A. Punchihewa, F. Gagnon, C. Despins, and V. K. Bhargava, "Blind multi-sources detection and localization for cognitive radio," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, New Orleans, LA, Dec. 2008, pp. 1–5.

[4] W. Guibène and D. Slock, "Signal separation and classification algorithm for cognitive radio networks," in *Proc. Int. Symp. Wireless Commun. Syst.*, Paris, France, Aug. 2012, pp. 301–304.

[5] G. Ivković, P. Spasojević, and I. Šeškar, "Single sensor radio scene analysis for packet based radio signals," in *Proc. Info. Theory Appl. Workshop (ITA)*, La Jolla, CA, Jan.-Feb. 2010, pp. 1–10.

[6] H. Chen and S.-J. Kim, "Robust RF mixture signal recognition using discriminative dictionary learning," *IEEE Access*, vol. 9, pp. 141107–141120, Oct. 2021.

[7] N. Q. K. Duong, E. Vincent, and R. Gribonval, "Under-determined reverberant audio source separation using a full-rank spatial covariance model," *IEEE Trans. Audio, Speech, and Lang. Process.*, vol. 18, no. 7, pp. 1830–1840, Sep. 2010.

[8] A. Ozerov and C. Févotte, "Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation," *IEEE Trans. Audio, Speech, and Lang. Process.*, vol. 18, no. 3, pp. 550–563, 2009.

[9] K. Greff, S. V. Steenkiste, and J. Schmidhuber, "Neural expectation maximization," in *Advances in Neural Info. Process. Syst.*, 2017, vol. 30, pp. 6691–6701.

[10] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015, pp. 234–241.

[11] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. of the 32nd Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[12] J. Azcarreta, N. Ito, S. Araki, and T. Nakatani, "Permutation-free CGMM: Complex Gaussian mixture model with inverse Wishart mixture model based spatial prior for permutation-free source separation and source counting," in *Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2018, pp. 51–55.

Authorized licensed use limited to: Univ of Calif Davis. Downloaded on January 28,2023 at 20:50:59 UTC from IEEE Xplore. Restrictions apply.