IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 29, 2021, accepted October 2, 2021. Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2021.3120635

Robust RF Mixture Signal Recognition Using Discriminative Dictionary Learning

HAO CHEN, (Graduate Student Member, IEEE),
AND SEUNG-JUN KIM, (Senior Member, IEEE)
Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, Baltimore, MD 21250, USA

Corresponding author: Seung-Jun Kim (sjkim@umbc.edu)

This work was supported in part by NSF under Grant 1547347 and Grant 1631838, and in part by Lockheed Martin Corporation.

ABSTRACT RF signal recognition is an important element toward RF situational awareness and dynamic spectrum management. In this work, machine learning-based signal recognition algorithms are proposed. Our key contribution is to engineer feature learning such that the classifiers can perform robustly even when a mixture of heterogeneous signal classes is observed, although the training is still done using non-mixture single-label samples. To achieve this, discriminative dictionary learning algorithms are developed with various feature-shaping constraints. The signal detection can then be done in a way reminiscent of the multi-user detection in wireless communication, employing linear equalizers. The algorithms are tested using real wideband RF measurement data. It is verified that the proposed algorithms can robustly classify the component signals even when their powers are widely different and their number is not known a priori.

INDEX TERMS Dictionary learning, multiple signal classification, RF signal recognition, supervised learning.

I. INTRODUCTION

RF signal recognition, such as modulation recognition, wireless technology identification, and wireless device fingerprinting, is an important building block for RF situational awareness in military applications and for dynamic spectrum sensing and interference management in commercial networks. With the advent of the 5G and Internet-of-Things (IoT) networks, more and more wireless devices with diverse access technologies cohabit in a shared spectrum, pressing the need for accurate and flexible RF interference recognition techniques.

Traditionally the RF signal recognition problems have been tackled by extracting handcrafted features from the signals, including the carrier frequency, the cyclic features such as the spectral correlation function (SCF), and the features related to modulation types and orders such as the higher-order moments and cumulants [1, Ch. 11]. More recently, various machine learning techniques are explored for this problem, where useful features are learned directly from the signal sets, often without much domain-specific adaptation. A convolutional neural network (CNN) was adopted to classify the modulation types of communication signals from their in-phase/quadrature (I/Q) samples in [2], [3]. A multi-modality fusion approach was proposed in [4], where handcrafted features and CNN-extracted features were combined for modulation classification. RF signals generated by various communication protocols were classified using CNNs [5]. By exploiting intrinsic variabilities in the radio hardware components, individual RF emitters were identified using deep neural network architectures [6], [7].

All these works dealt with the problem of classifying each input signal to a single signal type among multiple possible types. However, as more transmitters with various wireless technologies coexist in a crowded spectrum, it is desired to recognize the RF signals of concurrent transmissions belonging to multiple signal types. For example, in the ISM band, different wireless technologies, such as Wi-Fi, Bluetooth, and Zigbee, may coexist, and the identification of the interference mixture is important for an efficient use of the spectrum [8]. In the cognitive radio systems, secondary radios must sense the spectrum to prevent interfering with on-going transmissions of the primary or other secondary users sharing the band [9]. Recognition of mixture signals may also be instrumental for RF-based device/tag identification, where the signatures of multiple devices may be observed simultaneously [10]. A signal recognition algorithm designed for detecting a

The associate editor coordinating the review of this manuscript and approving it for publication was Mingjun Dai.

single type of transmission will not fare well in such scenarios.

Works that address RF mixture signal recognition are rather scarce. Features that can be robustly extracted from mixtures are manually selected based on the prior knowledge of the signals such as the unique protocol characteristics [8]. Under a linear mixing model, by exploiting a subspace structure or statistical independence, blind source separation (BSS) can estimate the individual components first, which can then be classified [11]–[13]. However, the number of sensors often needs to be no smaller than the number of components.

From the machine learning perspective, the problem can be viewed as an instance of multi-label classification, where one input is associated with potentially more than one class labels [14], [15]. Then, a simple approach is the binary relevance method, where a binary classifier is trained for each class to detect the presence of the class in the mixture. Another common approach is to build a power set of labels, and train a multi-class single-label classifier based on the expanded set of labels. In all these methods, it is critical that the multi-label (mixture) data sets are prepared and used for training.

However, collecting and training with multi-label data sets may become quite cumbersome. If there are C classes, the number of different mixtures increases exponentially as 2^C. The complexity can be significantly aggravated if one desires to build robustness against the dynamic ranges of the component signals, as then the combinations of different signal powers need to be considered as well. It is quite impractical to prepare data sets anticipating all possible power ratios of the components. It is noted that such disparate power ratios are pretty common in the RF applications.

In this paper, we tackle the mixture classification problem from a feature learning perspective. Different from the traditional multi-label classification methods that require mixture training samples, we propose to use *non-mixture* (single-label) samples in the training stage. Our key idea is to engineer the feature learning such that robust classification performance is achieved even for the *mixture* signals. For this purpose, novel dictionary learning (DL) formulations are proposed.

The DL framework postulates that the data possess a union-of-subspace structure, allowing the data samples to be well represented by a linear combination of a small number of atoms in a dictionary. The DL problem has often been formulated as unsupervised learning, minimizing a reconstruction error, and shown impressive performance in denoising, imputation, and dimensionality reduction [16], [17]. DL can be extended to supervised learning, where a dictionary that captures discriminative patterns in the data is learned by employing a label prediction cost, in addition to the reconstruction error [18], [19]. For instance, Fisher's discriminant cost can be incorporated, where the sparse coefficients corresponding to the same class data are encouraged to cluster together, while the distances between the cluster centroids are maximized [20].

In formulating the DL problems for mixture classification, our basic assumption is that a mixture signal sample is close to the linear combination of the samples of the individual component signals. For example, such a property holds for the cyclostationarity features such as the SCF when the component signals are uncorrelated. Recently, it was observed that deep neural network architectures can yield features that essentially linearize the data manifold, allowing the linear arithmetic in the feature space to have corresponding effects in the semantics [21], [22].

Even with approximate linearity, when a component signal is significantly stronger than others, the strong component can swamp the overall measurement and prevent the classifier from properly recognizing the weaker components. In wireless communication, such a phenomenon is called the *near-far problem* [23], [24]. The problem can be mitigated by employing appropriate equalization strategies, among others.

Inspired by this, and adopting the Fisher discriminant approach for supervised DL, our novel idea is to constrain the centroids corresponding to different classes to be approximately orthogonal to one another. The multi-label classifier can then be designed in a way similar to multi-user detection [25]. This approach significantly improves the classification performance of the weak signals mixed with stronger signals. To further improve the performance for severe near-far scenarios, we also consider a formulation that approximately spheres the distribution of the features around each centroid. Efficient optimization algorithms are derived for the formulated DL problems, where each step involves solving a convex optimization problem. The efficacy of the proposed methods is tested using RF data sets collected using software defined radios.

Preliminary results of this work were reported in a conference precursor [26]. In the present paper, more detailed exposition and derivation of the algorithms are provided, in addition to the results of extensive numerical tests with rigorous parameter tuning. Furthermore, a novel algorithm is developed (Algorithm 3), and its merit compared to other algorithms is verified. A Bayesian sparse coding method is also incorporated for the case where the number of mixture components is unknown.

The rest of the paper is organized as follows. In Sec. II, a supervised DL algorithm is derived using Fisher's criterion, which learns a dictionary as well as a linear transformation for the sparse coefficients. This algorithm serves as the baseline for performance comparison and subsequent algorithm derivation. In Sec. III, an algorithm tailored for mixture classification is developed employing feature orthogonality constraints. In Sec. IV, the method incorporating the whitening constraints is derived. Numerical test results are reported in Sec. V. The conclusion is provided in Sec. VI.

Notations: Bold uppercase symbols are used for matrices, bold lowercase for vectors, and calligraphic uppercase for sets. For matrix $\mathbf{X}$, $\mathbf{x}_k$ represents the k-th column, and x_{lm} denotes the (l, m)-entry. $\cdot^\top$ denotes transpose, and $\cdot^\dagger$ pseudo-inverse. $\mathbb{1}_{N\times M}$ is the N-by-M matrix with all entries equal to 1. $\|\cdot\|_F$ denotes the Frobenius norm, and $\|\cdot\|_1$ is the ℓ_1-norm, equal to the sum of the absolute values of entries. $\mathrm{tr}\{\cdot\}$ and $\lambda_{max}(\cdot)$ represent the trace and the maximum eigenvalue, respectively. bdiag$\{\cdot\}$ denotes the block diagonal matrix constructed by arranging the matrices listed in $\{\cdot\}$. $|a|$ is the absolute value of a, $|\mathcal{I}|$ the cardinality of set $\mathcal{I}$, and $*$ the convolution operator.

II. SUPERVISED DICTIONARY LEARNING

A. DICTIONARY LEARNING

Let $\mathbf{x}_i$, $i \in \mathcal{I} := \{1, 2, \ldots, N\}$, be an M-dimensional datum (sample), and $\mathbf{X} := [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N] \in \mathbb{R}^{M\times N}$ be the data set. Dictionary learning postulates that given an appropriate dictionary $\mathbf{D} \in \mathbb{R}^{M\times K}$ with K atoms (the columns of $\mathbf{D}$), the data $\mathbf{X}$ can be well represented as $\mathbf{X} \approx \mathbf{DZ}$, where the coefficient matrix $\mathbf{Z} \in \mathbb{R}^{K\times N}$ is sparse. A widely used approach for learning $\mathbf{D}$ from the data is to solve [17]

$$\min_{\mathbf{D}\in\mathcal{D},\mathbf{Z}} \|\mathbf{X} - \mathbf{DZ}\|_F^2 + \lambda\|\mathbf{Z}\|_1 \tag{1}$$

where $\lambda > 0$ is a parameter for adjusting the sparsity level of $\mathbf{Z}$. $\mathcal{D}$ is a constraint set for $\mathbf{D}$, which ensures that the solution is well-defined. For example, the columns of $\mathbf{D}$ are often constrained to have norms no greater than unity via

$$\mathcal{D} := \{[\mathbf{d}_1, \ldots, \mathbf{d}_K] \in \mathbb{R}^{M\times K} : \|\mathbf{d}_k\|_2^2 \le 1, k = 1, \ldots, K\} \tag{2}$$

which resolves the inherent scaling ambiguity of the bi-factor model $\mathbf{X} \approx \mathbf{DZ}$. That is, scaling a column in $\mathbf{D}$ by α and the corresponding row in $\mathbf{Z}$ by α^{-1} does not alter the product $\mathbf{DZ}$.

Formulation (1) is not a convex optimization problem. Thus, alternating minimization is often employed to obtain locally optimal solutions [16], [17]. DL has been shown to achieve state-of-the-art performance in a variety of image and signal processing applications such as image denoising, inpainting, and object recognition.

B. SUPERVISED DICTIONARY LEARNING FORMULATION

Through (1), a dictionary that allows faithful representation of the data is learned. On the other hand, one can also tailor the dictionary so as to capture the discriminative features that are useful for performing classification [18], [20]. Consider for now the single-label classification with C classes. Let $\mathbf{X}_c \in \mathbb{R}^{M\times N_c}$ be the collection of the class-c samples, for $c = 1, 2, \ldots, C$. That is, the columns of $\mathbf{X}_c$ are $\{\mathbf{x}_i\}_{i\in\mathcal{I}_c}$, where $\mathcal{I}_c$ denotes the index set for class-c data, and $N_c = |\mathcal{I}_c|$ is the number of the class-c samples with $\sum_{c=1}^C N_c = N$. With a slight abuse of notation, let us denote again by $\mathbf{X}$ the data matrix, where the same class samples are arranged in consecutive columns; that is, $\mathbf{X} := [\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_C] \in \mathbb{R}^{M\times N}$.

One way to obtain a discriminative dictionary is to incorporate the Fisher discriminant criterion to the learning formulation [20]. Here we make some departures from [20]. First, $\mathbf{D}$ is not partitioned to the sub-dictionaries corresponding to different classes. Therefore, in addition to the resulting simplicity of the formulation, each of the K atoms is not hard-assigned for representing the features of a particular class, but rather can flexibly represent the features of multiple classes.

Another important difference is that instead of using the coefficient matrix $\mathbf{Z}$ directly in the Fisher criterion as in [20], a linear transformation $\mathbf{W}^\top \in \mathbb{R}^{P\times K}$ is introduced, which is learned jointly with the dictionary. This transformation is important because typically not all atoms in the dictionary capture discriminative features. For example, there may be atoms that explain the common background, which is not informative for classification [27]. Projecting the K-dimensional sparse codes to a smaller P-dimensional space allows one to ignore those atoms unhelpful for discrimination, rendering further dimensionality and noise reduction, ultimately contributing to improved classification performance.

Let $\mathbf{z}_i$ be the coefficient vector corresponding to a sample $\mathbf{x}_i$, $i = 1, 2, \ldots, N$. Projecting $\mathbf{z}_i$ via $\mathbf{W}$ yields a discriminant variable $\mathbf{y}_i := \mathbf{W}^\top\mathbf{z}_i \in \mathbb{R}^P$, where P satisfies $C \le P \le K$. Let us define the class-c centroid vector and the overall sample mean vector of the sparse codes for data $\mathbf{X}$ as

$$\mathbf{m}_c := \frac{1}{N_c}\sum_{i\in\mathcal{I}_c}\mathbf{z}_i \text{ and } \mathbf{m} := \frac{1}{N}\sum_{i\in\mathcal{I}}\mathbf{z}_i \tag{3}$$

respectively. Then, one can define the within-class scatter matrix $\mathbf{S}_W$ and the between-class scatter matrix $\mathbf{S}_B$ of $\mathbf{Z}$ as

$$\mathbf{S}_W := \sum_{c=1}^C\sum_{i\in\mathcal{I}_c}(\mathbf{z}_i - \mathbf{m}_c)(\mathbf{z}_i - \mathbf{m}_c)^\top \tag{4}$$

$$\mathbf{S}_B := \sum_{c=1}^C N_c(\mathbf{m}_c - \mathbf{m})(\mathbf{m}_c - \mathbf{m})^\top \tag{5}$$

respectively. It can be easily shown that the overall scatter is the sum of the between-class scatter and the within-class scatter as [28]

$$\bar{\mathbf{S}} := \sum_{i\in\mathcal{I}}(\mathbf{z}_i - \mathbf{m})(\mathbf{z}_i - \mathbf{m})^\top = \mathbf{S}_B + \mathbf{S}_W. \tag{6}$$

Here, $\mathbf{S}_B$ corresponds to the "signal" term useful for classification, and $\mathbf{S}_W$ the "noise" term that confuses the classifier. The idea of the Fisher discriminant criterion is to maximize the between-class scatter $\mathbf{W}^\top\mathbf{S}_B\mathbf{W}$, and, at the same time, minimize the within-class scatter $\mathbf{W}^\top\mathbf{S}_W\mathbf{W}$. Thus, a reasonable penalty function to minimize is constructed as

$$f(\mathbf{W},\mathbf{Z}) := \mathrm{tr}\{\mathbf{W}^\top\mathbf{S}_W\mathbf{W}\} - \mathrm{tr}\{\mathbf{W}^\top\mathbf{S}_B\mathbf{W}\} + \alpha\|\mathbf{W}^\top\mathbf{Z}\|_F^2 \tag{7}$$

where the last term with $\alpha > 1$ is added to ensure strict convexity of $f(\mathbf{W},\mathbf{Z})$ with respect to $\mathbf{Z}$. To see this, define $N \times N$ matrices

$$\mathbf{H}_1 := \mathrm{bdiag}\left\{\frac{1}{N_1}\mathbb{1}_{N_1\times N_1}, \frac{1}{N_2}\mathbb{1}_{N_2\times N_2} \ldots, \frac{1}{N_C}\mathbb{1}_{N_C\times N_C}\right\} \tag{8}$$

TABLE 1. **Algorithm 1 for solving (P1).**

```
Input: $\{\mathbf{X}_c\}_{c=1}^{C}$, $K$, $P$, $\lambda$, $\mu$, $\alpha$, and $L_1$
Output: $\mathbf{D}$ and $\mathbf{W}$
Initialize $\mathbf{D}$ and $\mathbf{Z}$ randomly and set $\mathbf{W}$ to an arbitrary
   orthonormal matrix
Repeat
    /* Update $\mathbf{D}$ by solving (13) */
    Set $\mathbf{A} = \mathbf{Z}\mathbf{Z}^\top$ and $\mathbf{B} = \mathbf{X}\mathbf{Z}^\top$
    Set $i = 0$ and $\mathbf{D}^{(i)} := [\mathbf{d}_1^{(i)}, \ldots, \mathbf{d}_K^{(i)}] = \mathbf{D}$
    Repeat
        $i \leftarrow i + 1$
        For $k = 1, 2, \ldots, K$
            Set $\overline{\mathbf{D}} = [\mathbf{d}_1^{(i)}, \ldots, \mathbf{d}_{k-1}^{(i)}, \mathbf{d}_k^{(i-1)}, \ldots, \mathbf{d}_K^{(i-1)}]$
            Set $\tilde{\mathbf{d}}_k^{(i)} = \frac{1}{a_{kk}}\left(\mathbf{b}_k - \overline{\mathbf{D}}\mathbf{a}_k\right) + \mathbf{d}_k^{(i-1)}$
            Set $\mathbf{d}_k^{(i)} = \frac{\tilde{\mathbf{d}}_k^{(i)}}{\max\left\{\|\tilde{\mathbf{d}}_k^{(i)}\|_2, 1\right\}}$
        Next $k$
    Until convergence
    Set $\mathbf{D} = \mathbf{D}^{(i)}$
    /* Update $\mathbf{Z}$ by solving (14) */
    Set $i = 0$, $t^{(i)} = 0$, $\widetilde{\mathbf{Z}}^{(i)} := \mathbf{Z}$ and $\mathbf{Z}^{(i)} := \mathbf{Z}$
    Repeat
        $\mathbf{G}^{(i)} = -2\mathbf{D}^\top(\mathbf{X} - \mathbf{D}\mathbf{Z}^{(i)}) + 2\mu\mathbf{W}\mathbf{W}^\top\mathbf{Z}^{(i)}\mathbf{S}$
        $\mathbf{Z}^{(i+1)} = \mathcal{S}_{\lambda/L_1}\left(\widetilde{\mathbf{Z}}^{(i)} - \frac{1}{L_1}\mathbf{G}^{(i)}\right)$
        $t^{(i+1)} = \frac{1}{2}\left(1 + \sqrt{1 + 4(t^{(i)})^2}\right)$
        $\widetilde{\mathbf{Z}}^{(i+1)} = \mathbf{Z}^{(i)} + \frac{t^{(i)}-1}{t^{(i+1)}}(\mathbf{Z}^{(i+1)} - \mathbf{Z}^{(i)})$
        $i \leftarrow i + 1$
    Until convergence
    Set $\mathbf{Z} = \mathbf{Z}^{(i)}$
    Set the columns of $\mathbf{W}$ to the $P$ eigenvectors of $\mathbf{Z}\mathbf{S}\mathbf{Z}^\top$
    corresponding to the smallest eigenvalues
Until convergence
```

and $\mathbf{H}_2 := \frac{1}{N}\mathbb{1}_{N\times N}$. Then, $f(\mathbf{W}, \mathbf{Z})$ in (7) can be written as

$$\|\mathbf{W}^\top\mathbf{Z}(\mathbf{I}-\mathbf{H}_1)\|_F^2 - \|\mathbf{W}^\top\mathbf{Z}(\mathbf{H}_1-\mathbf{H}_2)\|_F^2 + \alpha\|\mathbf{W}^\top\mathbf{Z}\|_F^2 \tag{9}$$

$$= \mathrm{tr}\{\mathbf{W}^\top\mathbf{Z}\mathbf{S}\mathbf{Z}^\top\mathbf{W}\} \tag{10}$$

where $\mathbf{S} := (1+\alpha)\mathbf{I} - 2\mathbf{H}_1 + \mathbf{H}_2$. Here, the fact that $\mathbf{H}_1\mathbf{H}_2 = \mathbf{H}_2\mathbf{H}_1 = \mathbf{H}_2$ and $\mathbf{H}_2^2 = \mathbf{H}_2$ is used. It can be shown that $\mathbf{S}$ is positive definite for $\alpha > 1$ [29]. Thus, for an arbitrary $\mathbf{T}$, $\mathrm{tr}\{\mathbf{T}^\top\mathbf{S}\mathbf{T}\}$ is strictly convex in $\mathbf{T}$, and so is $f(\mathbf{W}, \mathbf{Z})$ in $\mathbf{Z}$ since $\mathbf{W}$ will be constrained to be full-rank [cf. (12)].

The overall optimization problem can now be formulated as

$$\text{(P1)} \quad \min_{\mathbf{D}\in\mathcal{D},\mathbf{Z},\mathbf{W}} \|\mathbf{X} - \mathbf{D}\mathbf{Z}\|_F^2 + \lambda\|\mathbf{Z}\|_1 + \mu f(\mathbf{W}, \mathbf{Z}) \tag{11}$$

$$\text{subject to } \mathbf{W}^\top\mathbf{W} = \mathbf{I} \tag{12}$$

where $\lambda, \mu > 0$ are parameters. Constraint (12) is added to avoid the trivial solution $\mathbf{W} = \mathbf{0}$.

C. ALGORITHM DERIVATION

Problem (P1) is not convex as the cost function is not jointly convex with respect to $(\mathbf{D}, \mathbf{Z}, \mathbf{W})$ and constraint (12) is nonconvex. However, the optimization with respect to one variable out of $\{\mathbf{D}, \mathbf{Z}, \mathbf{W}\}$ can be solved easily, provided the remaining two are held fixed. Thus, an alternating minimization approach based on the block coordinate descent (BCD) method is proposed [30].

First, with $\mathbf{Z}$ and $\mathbf{W}$ fixed, the objective of (11) is minimized with respect to $\mathbf{D}$. This is equivalent to solving

$$\min_{\mathbf{D}\in\mathcal{D}} \|\mathbf{X} - \mathbf{D}\mathbf{Z}\|_F^2 + \lambda\|\mathbf{Z}\|_1. \tag{13}$$

This is a convex problem, which can be solved, for example, by employing another layer of the BCD method as in [17].

With $\mathbf{D}$ and $\mathbf{W}$ fixed, the sub-problem for $\mathbf{Z}$ is given by

$$\min_{\mathbf{Z}} \|\mathbf{X} - \mathbf{D}\mathbf{Z}\|_F^2 + \lambda\|\mathbf{Z}\|_1 + \mu f(\mathbf{W}, \mathbf{Z}) \tag{14}$$

which is a convex optimization problem. Recall that thanks to the term $\alpha\|\mathbf{W}^\top\mathbf{Z}\|_F^2$ in (9), $f(\mathbf{W}, \mathbf{Z})$ is convex with respect to $\mathbf{Z}$. Thus, the problem can be solved with various algorithms that can deal with the ℓ_1-norm regularizer efficiently. For example, the fast iterative shrinkage-thresholding algorithm (FISTA) can be employed [31], which requires the gradient of the smooth part of the objective and the Lipshitz constant of the gradient. The gradient is given by

$$\frac{\partial}{\partial\mathbf{Z}}\left[\|\mathbf{X} - \mathbf{D}\mathbf{Z}\|_F^2 + \mu f(\mathbf{W}, \mathbf{Z})\right]$$

$$= -2\mathbf{D}^\top(\mathbf{X} - \mathbf{D}\mathbf{Z}) + \mu\frac{\partial}{\partial\mathbf{Z}}f(\mathbf{W}, \mathbf{Z}) \tag{15}$$

where $\frac{\partial f(\mathbf{W},\mathbf{Z})}{\partial\mathbf{Z}} = 2\mathbf{W}\mathbf{W}^\top\mathbf{Z}\mathbf{S}$ [cf. (9)]. The Lipschitz constant can be shown to be

$$L_1 := \lambda_{max}(\mathbf{D}^\top\mathbf{D}) + 2\mu\lambda_{max}(\mathbf{W}\mathbf{W}^\top)\lambda_{max}(\mathbf{S}). \tag{16}$$

Finally, with $\mathbf{D}$ and $\mathbf{Z}$ fixed, the update for $\mathbf{W}$ is done via

$$\min_{\mathbf{W}:\mathbf{W}^\top\mathbf{W}=\mathbf{I}} \mathrm{tr}\left\{\mathbf{W}^\top\left[\mathbf{S}_W - \mathbf{S}_B + \alpha\mathbf{Z}\mathbf{Z}^\top\right]\mathbf{W}\right\}. \tag{17}$$

Thus, the optimal $\mathbf{W}$ can be obtained as the eigenvectors corresponding to the P smallest eigenvalues.

The aforementioned BCD steps can be repeated until convergence. Under mild conditions, the DL step yields a unique minimum for $\mathbf{D}$ [17]. Thus, it can be guaranteed that the iterates of the proposed training algorithm converge [30]. The overall algorithm is summarized in Table 1, where $\mathcal{S}_b(a) := \mathrm{sign}(a)\max\{|a| - b, 0\}$ is a shrinkage operator.

III. DL FOR MIXTURE CLASSIFICATION

A. MOTIVATION

The algorithm developed in Sec. II best suits the problem of classifying a signal that belongs to a single class. When presented with a signal that contains a mixture of components from multiple classes, it is expected that the corresponding discriminant variable $\mathbf{y}$ possesses the features of the constituent classes. Provided that the nonlinear coupling among different class features can be neglected, one can expect that $\mathbf{y}$ would lie somewhere in between the centroids of the component classes. This can confuse the classifier trained on single-label samples. On the other hand, if the feature subspaces for different classes are not overlapping significantly, but rather sufficiently decorrelated, one should be able to focus on each class subspace while nulling out the other classes' features.

In conventional multi-label classification, the classifiers are trained using mixture samples. However, this can dramatically increase the training set size as the number of combinations increases exponentially. Also, for applications such as the RF, the dynamic ranges of the component signals can be large, yielding different power ratios. Preparing the training data capturing such variety may not be straightforward.

In wireless communication, the challenge associated with detecting component signals that have widely disparate power levels is termed the *near-far problem*. When the same frequency spectrum is shared among multiple transmitter-receiver pairs (users), and a user's signal power is much higher than those of the rest, the strong signal can dominate the weaker components, rendering it difficult to detect the weaker users [23]. The remedies include performing power control and precoding, combined with linear/nonlinear equalization and multi-user detection strategies [24], [25].

Inspired by this, we take a pragmatic approach, where the distribution of the discriminant variable is engineered in such a way that the components due to different mixture classes can be easily separated and detected. In particular, in order to mitigate the near-far issue and prevent the strong signals from leaking into the weaker signal subspaces, orthogonality is encouraged among different class signals.

B. PROBLEM FORMULATION

Our novel idea is to introduce an additional penalty term in the training objective to impose orthogonality constraints among different class centroids $\{\mathbf{W}^\top\mathbf{m}_c\}_{c=1}^C$. Specifically, a matrix $\mathbf{U} \in \mathbb{R}^{P\times C}$, whose columns are orthonormal as $\mathbf{U}^\top\mathbf{U} = \mathbf{I}$, is newly introduced, and the class centroids are constrained to be close to $\mathbf{U}$.

Specifically, first collect $\{\mathbf{m}_c\}$ as $\mathbf{M} := [\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_C] \in \mathbb{R}^{K\times C}$. Upon defining $\mathbf{H}_0 := \text{bdiag}\{\frac{1}{N_1}\mathbb{1}_{N_1\times 1}, \ldots, \frac{1}{N_C}\mathbb{1}_{N_C\times 1}\} \in \mathbb{R}^{N\times C}$, one can verify that $\mathbf{M} = \mathbf{Z}\mathbf{H}_0$. The orthogonality constraints can be encoded into the penalty term

$$g(\mathbf{W}, \mathbf{Z}, \mathbf{U}) := \|\mathbf{W}^\top\mathbf{M} - \mathbf{U}\|_F^2 = \|\mathbf{W}^\top\mathbf{Z}\mathbf{H}_0 - \mathbf{U}\|_F^2. \quad (18)$$

Augmenting this to the objective in (11), the proposed formulation is given by

$$\text{(P2)} \quad \min_{\mathbf{D}\in\mathcal{D},\mathbf{Z},\mathbf{W},\mathbf{U}} \|\mathbf{X} - \mathbf{D}\mathbf{Z}\|_F^2 + \lambda\|\mathbf{Z}\|_1 + \mu f(\mathbf{W}, \mathbf{Z}) + \nu g(\mathbf{W}, \mathbf{Z}, \mathbf{U}) \quad (19)$$

$$\text{subject to } \mathbf{U}^\top\mathbf{U} = \mathbf{I} \quad (20)$$

where $\nu > 0$ is an appropriate weight. Note that constraint (12) in (P1) is removed since the orthogonality of $\mathbf{U}$ alone can avoid the trivial solution $\mathbf{W} = \mathbf{0}$.

Continuing with the analogy to wireless communication, one can interpret the class centroids $\{\mathbf{m}_c\}$ as the multi-user signal constellation, and $\mathbf{Z}$ as the noisy signals received at the receiver. The linear transformation $\mathbf{W}^\top$ then plays the role of the equalizer that combats the multi-user interference.

TABLE 2. Algorithm 2 for solving (P2).

```
Input: $\{\mathbf{X}_c\}_{c=1}^C$, $K$, $P$, $\lambda$, $\mu$, $\nu$, $\alpha$, and $L_2$
Output: $\mathbf{D}$ and $\mathbf{W}$
1: Initialize $\mathbf{D}$, $\mathbf{Z}$, and $\mathbf{W}$ randomly and set $\mathbf{U}$ to an arbitrary
   orthonormal matrix
2: Repeat
3:     Update $\mathbf{D}$ via lines 3–13 in Table 1
4:     Update $\mathbf{Z}$ by solving (21) via lines 14–22 in Table 1
       but replacing lines 16 and 17 by the following.
       16′: $\mathbf{G}^{(i)} = -2\mathbf{D}^\top(\mathbf{X} - \mathbf{D}\mathbf{Z}^{(i)}) + 2\mu\mathbf{W}\mathbf{W}^\top\mathbf{Z}^{(i)}\mathbf{S} +$
                $2\nu\mathbf{W}(\mathbf{W}^\top\mathbf{Z}^{(i)}\mathbf{H}_0 - \mathbf{U})\mathbf{H}_0^\top$
       17′: $\mathbf{Z}^{(i+1)} = \mathcal{S}_{\lambda/L_2}\left(\widetilde{\mathbf{Z}}^{(i)} - \frac{1}{L_2}\mathbf{G}^{(i)}\right)$
5:     Set $\mathbf{W} = \left(\frac{\mu}{\nu}\mathbf{Z}\mathbf{S}\mathbf{Z}^\top + \mathbf{M}\mathbf{M}^\top\right)^{-1}\mathbf{M}\mathbf{U}^\top$
6:     Update $\mathbf{U}$ using (26) or (27)
7: Until convergence
```

C. TRAINING ALGORITHM

Similar to the algorithm for (P1), the algorithm for (P2) can be derived using alternating minimization. First, with $\mathbf{Z}$, $\mathbf{W}$, and $\mathbf{U}$ fixed, the update for $\mathbf{D}$ is again equivalent to solving (1). For updating $\mathbf{Z}$, with $\mathbf{D}$, $\mathbf{W}$ and $\mathbf{U}$ fixed, the relevant subproblem is given by

$$\min_{\mathbf{Z}} \|\mathbf{X} - \mathbf{D}\mathbf{Z}\|_F^2 + \mu f(\mathbf{W}, \mathbf{Z}) + \nu g(\mathbf{W}, \mathbf{Z}, \mathbf{U}) + \lambda\|\mathbf{Z}\|_1. \quad (21)$$

It is noted that (21) minimizes a smooth convex term plus an ℓ_1-norm regularizer. Thus, the FISTA can be again employed, where the necessary gradient of the smooth part is given as

$$\frac{\partial}{\partial\mathbf{Z}}\left[\|\mathbf{X} - \mathbf{D}\mathbf{Z}\|_F^2 + \mu f(\mathbf{W}, \mathbf{Z}) + \nu g(\mathbf{W}, \mathbf{Z}, \mathbf{U})\right] = -2\mathbf{D}^\top(\mathbf{X} - \mathbf{D}\mathbf{Z}) + 2\mu\mathbf{W}\mathbf{W}^\top\mathbf{Z}\mathbf{S} + 2\nu\mathbf{W}(\mathbf{W}^\top\mathbf{Z}\mathbf{H}_0 - \mathbf{U})\mathbf{H}_0^\top \quad (22)$$

and its Lipschitz constant as

$$L_2 := L_1 + 2\nu\lambda_{max}(\mathbf{W}\mathbf{W}^\top)\lambda_{max}(\mathbf{H}_0\mathbf{H}_0^\top). \quad (23)$$

The update for $\mathbf{W}$, with all other variables fixed, is done by minimizing a convex quadratic cost, whose closed-form solution is given by

$$\mathbf{W} = \left(\frac{\mu}{\nu}\mathbf{Z}\mathbf{S}\mathbf{Z}^\top + \mathbf{M}\mathbf{M}^\top\right)^{-1}\mathbf{M}\mathbf{U}^\top \quad (24)$$

assuming that $\mathbf{Z}$ is full-rank.

Finally, the update for $\mathbf{U}$ is done by solving

$$\mathbf{U} = \arg\min_{\mathbf{U}:\mathbf{U}^\top\mathbf{U}=\mathbf{I}} \|\mathbf{U} - \mathbf{W}^\top\mathbf{M}\|_F^2. \quad (25)$$

Problem (25) is an instance of the orthogonal Procrustes problem [32]. Perform a singular value decomposition (SVD) on $\mathbf{W}^\top\mathbf{M} = \overline{\mathbf{U}}_1\mathbf{\Sigma}_1\overline{\mathbf{V}}_1^\top$, where $\overline{\mathbf{U}}_1 \in \mathbb{R}^{P\times P}$ and $\overline{\mathbf{V}}_1 \in \mathbb{R}^{C\times C}$ are orthonormal and $\mathbf{\Sigma}_1 \in \mathbb{R}^{P\times C}$ is diagonal. Then, the solution to (25) is given by [33]

$$\mathbf{U} = \overline{\mathbf{U}}_1\mathbf{I}_{P\times C}\overline{\mathbf{V}}_1^\top \quad (26)$$

where $\mathbf{I}_{P \times C} := [\mathbf{I}_C, \mathbf{0}_{C \times (P-C)}]^\top$. When $\mathbf{W}^\top \mathbf{M}$ has the full rank C, the solution can also be obtained by first performing a SVD on $\mathbf{M}^\top \mathbf{W}\mathbf{W}^\top \mathbf{M} = \overline{\mathbf{V}}_2 \mathbf{\Sigma}_2 \overline{\mathbf{V}}_2^\top$, and computing

$$\mathbf{U} = \mathbf{W}^\top \mathbf{M} \overline{\mathbf{V}}_2 \mathbf{\Sigma}_2^{-1/2} \overline{\mathbf{V}}_2^\top. \qquad (27)$$

The update steps for $\mathbf{D}$, $\mathbf{Z}$, $\mathbf{W}$, and $\mathbf{U}$ are repeated until convergence. The convergence is again guaranteed by noting that the updates for $\mathbf{D}$ and $\mathbf{W}$ correspond to the unique minima of the respective sub-problems [30]. The overall algorithm is listed in Table 2.

IV. DL WITH WHITENING

A. MOTIVATION AND PROBLEM FORMULATION

It is recalled from (6) that the overall scatter contains not only the "signal" term but also the "noise" term. In Sec. III, the focus was on engineering the "signal" term such that the centroids are approximately orthogonal. In this section, the "noise" part is manipulated.

In particular, note that if the "noise" scatter $\mathbf{W}^\top \mathbf{S}_W \mathbf{W}$ is elongated along the direction connecting any two class centroids $\mathbf{W}^\top \mathbf{m}_c$ and $\mathbf{W}^\top \mathbf{m}_{c'}$ for $c \neq c'$, then the discriminative performance between these two classes can be degraded. In [34], a discriminant component analysis technique is proposed, where the discriminant projection $\mathbf{W}$ is learned under the constraint that the noise is whitened. That is, $\mathrm{tr}\{\mathbf{W}^\top \mathbf{S}_B \mathbf{W}\}$ is maximized subject to the constraint

$$\mathbf{W}^\top \mathbf{S}_W \mathbf{W} = \mathbf{I}. \qquad (28)$$

That is, the noise distribution becomes spherical after projection to $\mathbf{W}$. However, (28) constrains the noise scatter averaged over all classes [cf. (4)]. Therefore, when per-class noise powers are disparate, it may be influenced much by the few classes with strong noise power.

Thus, we take one step further and whiten the per-class scatter of the discriminants $\{\mathbf{W}^\top \mathbf{z}_i\}_{i \in \mathcal{I}_c}$ for each class $c = 1, 2, \ldots, C$. That is, for class c, it is constrained that

$$\sum_{i \in \mathcal{I}_c} \mathbf{W}^\top (\mathbf{z}_i - \mathbf{m}_c)(\mathbf{z}_i - \mathbf{m}_c)^\top \mathbf{W} \approx \delta_c^2 \mathbf{I} \qquad (29)$$

for an appropriate scaling parameter $\delta_c > 0$. The whitening constraint shapes the data cloud of class c to be spherical with radius δ_c around the centroid $\mathbf{W}^\top \mathbf{m}_c$. It is necessary to determine δ_c from data as well, since the centroids themselves are placed around the unit-radius sphere, per (18) and (20).

One could use the squared error between the left-hand and the right-hand sides of (29) as the penalty term, but this will lead to a quartic function, which is not convex. Instead, let us introduce an additional matrix variable $\mathbf{V}_c \in \mathbb{R}^{P \times N_c}$, which satisfies $\mathbf{V}_c \mathbf{V}_c^\top = \mathbf{I}$. Let $\mathbf{Z}_c \in \mathbb{R}^{K \times N_c}$ denote the collection of class-c sparse coefficients $\{\mathbf{z}_i\}_{i \in \mathcal{I}_c}$. Define also $\mathbf{M}_c := [\mathbf{m}_c, \mathbf{m}_c, \ldots, \mathbf{m}_c] = \mathbf{m}_c \mathbb{1}_{1 \times N_c} \in \mathbb{R}^{K \times N_c}$. Then, one can use the constraints

$$\mathbf{W}^\top (\mathbf{Z}_c - \mathbf{M}_c) \approx \delta_c \mathbf{V}_c, \quad c = 1, 2, \ldots, C. \qquad (30)$$

Thus, upon defining $\mathbf{V} := [\mathbf{V}_1, \ldots, \mathbf{V}_C]$ and $\boldsymbol{\delta} := [\delta_1, \ldots, \delta_C]^\top$, a whitening penalty is given by

$$s(\mathbf{W}, \mathbf{Z}, \mathbf{V}, \boldsymbol{\delta}) = \sum_{c=1}^{C} \|\mathbf{W}^\top (\mathbf{Z}_c - \mathbf{M}_c) - \delta_c \mathbf{V}_c\|_F^2. \qquad (31)$$

The overall optimization problem can be formulated as

$$\text{(P3)} \quad \min_{\mathbf{D} \in \mathcal{D}, \mathbf{Z}, \mathbf{W}, \mathbf{U}, \mathbf{V}, \boldsymbol{\delta}} \|\mathbf{X} - \mathbf{D}\mathbf{Z}\|_F^2 + \lambda \|\mathbf{Z}\|_1 + \mu f(\mathbf{W}, \mathbf{Z})$$
$$+ \nu g(\mathbf{W}, \mathbf{Z}, \mathbf{U}) + \omega s(\mathbf{W}, \mathbf{Z}, \mathbf{V}, \boldsymbol{\delta}) \qquad (32)$$
$$\text{subject to } \mathbf{U}^\top \mathbf{U} = \mathbf{I}, \ \mathbf{V}_c \mathbf{V}_c^\top = \mathbf{I}, \ c = 1, 2, \ldots, C \qquad (33)$$

where $\omega > 0$ is another weighting parameter.

B. ALGORITHM DERIVATION

Problem (P3) is not convex jointly in all the variables. Thus, the BCD method is again employed with the blocks $\mathbf{D}$, $\mathbf{Z}$, $\mathbf{W}$, $\mathbf{U}$, $\mathbf{V}$, and $\boldsymbol{\delta}$. For each of these blocks, the sub-problem with the rest of the block variables fixed can be solved optimally. For updating $\mathbf{D}$, the sub-problem is exactly the same as before.

For updating $\mathbf{Z}$, the sub-problem is given by

$$\min_{\mathbf{Z}} \|\mathbf{X} - \mathbf{D}\mathbf{Z}\|_F^2 + \mu f(\mathbf{W}, \mathbf{Z}) + \nu g(\mathbf{W}, \mathbf{Z}, \mathbf{U})$$
$$+ \omega s(\mathbf{W}, \mathbf{Z}, \mathbf{V}, \boldsymbol{\delta}) + \lambda \|\mathbf{Z}\|_1. \qquad (34)$$

Note that the first four terms in the objective of (34) are convex quadratic in $\mathbf{Z}$. Thus, we have a convex cost plus a non-smooth ℓ_1-norm term, again solved by the FISTA. To facilitate the computation of the derivative of the smooth part w.r.t. $\mathbf{Z}$, define

$$\overline{\mathbf{H}}_c := \mathbf{I} - \frac{1}{N_c} \mathbb{1}_{N_c \times N_c} \in \mathbb{R}^{N_c \times N_c} \qquad (35)$$
$$\widetilde{\mathbf{H}}_c := [\mathbf{0}_{N_c \times \sum_{i=1}^{c-1} N_i}, \mathbf{I}_{N_c}, \mathbf{0}_{N_c \times \sum_{i=c+1}^{C} N_i}]^\top \in \mathbb{R}^{N \times N_c} \qquad (36)$$

as well as $\widehat{\mathbf{H}}_c := \widetilde{\mathbf{H}}_c \overline{\mathbf{H}}_c$. Then, it is noted that $\mathbf{Z}_c = \mathbf{Z}\widetilde{\mathbf{H}}_c$, and $\mathbf{W}^\top (\mathbf{Z}_c - \mathbf{M}_c) = \mathbf{W}^\top \mathbf{Z} \widetilde{\mathbf{H}}_c \overline{\mathbf{H}}_c = \mathbf{W}^\top \mathbf{Z} \widehat{\mathbf{H}}_c$. Thus, (31) can be re-written as

$$s(\mathbf{W}, \mathbf{Z}, \mathbf{V}, \boldsymbol{\delta}) = \sum_{c=1}^{C} \|\mathbf{W}^\top \mathbf{Z} \widehat{\mathbf{H}}_c - \delta_c \mathbf{V}_c\|_F^2. \qquad (37)$$

The derivative needed for the FISTA can be expressed as

$$\frac{\partial}{\partial \mathbf{Z}} \Big[\|\mathbf{X} - \mathbf{D}\mathbf{Z}\|_F^2 + \mu f(\mathbf{W}, \mathbf{Z}) + \nu g(\mathbf{W}, \mathbf{Z}, \mathbf{U}) + \omega s(\mathbf{W}, \mathbf{Z}, \mathbf{V}, \boldsymbol{\delta}) \Big]$$
$$= -2\mathbf{D}^\top (\mathbf{X} - \mathbf{D}\mathbf{Z}) + 2\mu \mathbf{W}\mathbf{W}^\top \mathbf{Z}\mathbf{S} + 2\nu \mathbf{W}(\mathbf{W}^\top \mathbf{Z}\mathbf{H}_0 - \mathbf{U})\mathbf{H}_0^\top$$
$$+ 2\omega \sum_{c=1}^{C} \mathbf{W}(\mathbf{W}^\top \mathbf{Z} \widehat{\mathbf{H}}_c - \delta_c \mathbf{V}_c) \widehat{\mathbf{H}}_c^\top \qquad (38)$$

and the Lipschitz constant is given by

$$L_3 := L_2 + 2\omega \lambda_{max}(\mathbf{W}\mathbf{W}^\top) \sum_{c=1}^{C} \lambda_{max}(\widehat{\mathbf{H}}_c \widehat{\mathbf{H}}_c^\top). \qquad (39)$$

TABLE 3. **Algorithm 3 for solving (P3).**

```
Input: ${\mathbf{X}_c}_{c=1}^C$, $K$, $P$, $\lambda$, $\mu$, $\nu$, $\omega$, $\alpha$, and $L_3$
Output: $\mathbf{D}$ and $\mathbf{W}$
1: Initialize $\mathbf{D}$, $\mathbf{Z}$, and $\mathbf{W}$ randomly and set $\mathbf{U}$ and $\{\mathbf{V}_c^\top\}_{c=1}^C$ to arbitrary
   orthonormal matrix
2: Repeat
3:     Update $\mathbf{D}$ via lines 3–13 in Table 1
4:     Update $\mathbf{Z}$ by solving (34) via lines 14–22 in Table 1
       but replacing lines 16 and 17 by the following.
       16′: $\mathbf{G}^{(i)} = -2\mathbf{D}^\top(\mathbf{X} - \mathbf{D}\mathbf{Z}^{(i)}) + 2\mu\mathbf{W}\mathbf{W}^\top\mathbf{Z}^{(i)}\mathbf{S}$
         $+2\nu\mathbf{W}(\mathbf{W}^\top\mathbf{Z}^{(i)}\mathbf{H}_0 - \mathbf{U})\mathbf{H}_0^\top$
         $+2\omega\sum_{c=1}^C \mathbf{W}(\mathbf{W}^\top\mathbf{Z}^{(i)}\widehat{\mathbf{H}}_c - \delta_c\mathbf{V}_c)\widehat{\mathbf{H}}_c^\top$
       17′: $\mathbf{Z}^{(i+1)} = \mathcal{S}_{\lambda/L_3}\left(\widetilde{\mathbf{Z}}^{(i)} - \frac{1}{L_3}\mathbf{G}^{(i)}\right)$
5:     Set $\mathbf{W} = \left(\mu\mathbf{Z}\mathbf{S}\mathbf{Z}^\top + \nu\mathbf{M}\mathbf{M}^\top + \omega\sum_{c=1}^C \mathbf{Z}\widehat{\mathbf{H}}_c\widehat{\mathbf{H}}_c^\top\mathbf{Z}^\top\right)^{-1}$
           $\left(\nu\mathbf{M}\mathbf{U}^\top + \omega\sum_{c=1}^C \mathbf{Z}\widehat{\mathbf{H}}_c\delta_c\mathbf{V}_c^\top\right)$
6:     Update $\mathbf{U}$ using (26) or (27)
7:     Update $\mathbf{V}_c$ using (45) for $c = 1, 2, \ldots, C$
8:     Set $\delta_c = \frac{\mathrm{tr}\{\mathbf{V}_c^\top\mathbf{W}^\top\mathbf{Z}\widehat{\mathbf{H}}_c\}}{\|\mathbf{V}_c\|_F^2}$ for $c = 1, 2, \ldots, C$
9: Until convergence
```

For updating $\mathbf{W}$, the relevant sub-problem is given by

$$\min_{\mathbf{W}} \mu f(\mathbf{W}, \mathbf{Z}) + \nu g(\mathbf{W}, \mathbf{Z}, \mathbf{U}) + \omega s(\mathbf{W}, \mathbf{Z}, \mathbf{V}, \boldsymbol{\delta}) \quad (40)$$

whose solution can be obtained simply by taking the derivative of the objective in (40) and setting it to zero. This yields a closed-form solution for $\mathbf{W}$ as

$$\mathbf{W} = \left(\mu\mathbf{Z}\mathbf{S}\mathbf{Z}^\top + \nu\mathbf{M}\mathbf{M}^\top + \omega\sum_{c=1}^{C}\mathbf{Z}\widehat{\mathbf{H}}_c\widehat{\mathbf{H}}_c^\top\mathbf{Z}^\top\right)^{-1} \times \left(\nu\mathbf{M}\mathbf{U}^\top + \omega\sum_{c=1}^{C}\mathbf{Z}\widehat{\mathbf{H}}_c\delta_c\mathbf{V}_c^\top\right). \quad (41)$$

The $\mathbf{U}$ update is the same as in (26)–(27). The sub-problem for $\mathbf{V}$ is

$$\min_{\mathbf{V}} s(\mathbf{W}, \mathbf{Z}, \mathbf{V}, \boldsymbol{\delta}) \text{ subject to } \mathbf{V}_c\mathbf{V}_c^\top = \mathbf{I}, \; c = 1, 2, \ldots, C \quad (42)$$

which is equivalent to solving for each $c = 1, 2, \ldots, C$

$$\min_{\mathbf{V}_c} \|\mathbf{W}^\top(\mathbf{Z}_c - \mathbf{M}_c) - \delta_c\mathbf{V}_c\|_F^2 \text{ subject to } \mathbf{V}_c\mathbf{V}_c^\top = \mathbf{I}. \quad (43)$$

It can be seen that (43) is again an instance of the orthogonal Procrustes problem since it is equivalent to

$$\min_{\mathbf{V}_c} \|(\mathbf{Z}_c - \mathbf{M}_c)^\top\mathbf{W}/\delta_c - \mathbf{V}_c^\top\|_F^2 \text{ subject to } \mathbf{V}_c\mathbf{V}_c^\top = \mathbf{I}. \quad (44)$$

Let the SVD of $(\mathbf{Z}_c - \mathbf{M}_c)^\top\mathbf{W}/\delta_c = \widehat{\mathbf{H}}_c^\top\mathbf{Z}^\top\mathbf{W}/\delta_c$ is given by $\widetilde{\mathbf{U}}_c\widetilde{\boldsymbol{\Sigma}}_c\widetilde{\mathbf{V}}_c^\top$, where $\widetilde{\mathbf{U}}_c \in \mathbb{R}^{N_c \times N_c}$ and $\widetilde{\mathbf{V}}_c \in \mathbb{R}^{P \times P}$ are orthonormal, and $\widetilde{\boldsymbol{\Sigma}}_c \in \mathbb{R}^{N_c \times P}$ is diagonal. Then, the solution to (43) is given by

$$\mathbf{V}_c = \widetilde{\mathbf{V}}_c\mathbf{I}_{P \times N_c}\widetilde{\mathbf{U}}_c^\top. \quad (45)$$

The sub-problem for $\boldsymbol{\delta}$ can be solved for each element δ_c independently as well. It is a simple least-squares problem

$$\min_{\delta_c} \|\mathbf{W}^\top\mathbf{Z}\widehat{\mathbf{H}}_c - \delta_c\mathbf{V}_c\|_F^2 \quad (46)$$

with the solution given by

$$\delta_c = \frac{\mathrm{tr}\{\mathbf{V}_c^\top\mathbf{W}^\top\mathbf{Z}\widehat{\mathbf{H}}_c\}}{\|\mathbf{V}_c\|_F^2}. \quad (47)$$

The overall algorithm is listed in Table 3.

V. NUMERICAL TESTS

A. DATA SET AND PREPROCESSING

The performance of the proposed methods is tested on real RF data sets collected in the 2.4 GHz band using a software defined radio from Ettus Research. The downconverted complex signals of 40 MHz bandwidth were acquired inside a RF shield box. The transmissions of Wi-Fi, Bluetooth, and Bluetooth Low Energy (BLE) protocols were generated using a vector signal generator from Rohde & Schwarz. The Wi-Fi signal actually has two categories: Wi-Fi with high occupancy (denoted as Wi-Fi1 in the sequel) and Wi-Fi with low occupancy (Wi-Fi2). The high occupancy signals capture intensive Wi-Fi usage such as downloading a large file, whereas the low occupancy signals represent a more sporadic use case. Moreover, two types of drone controllers were used as transmitters, which generated unique proprietary frequency hopping spread spectrum (FHSS) waveforms, which we term FHSS1 and FHSS2.

For each of the 6 classes, the samples were collected for 45 s. Then, 449 temporal snapshots of duration 200 ms each were extracted per class by allowing overlaps no longer than 100 ms. Of the 449 snapshots, 300 were used for training, 74 for cross-validation, and 75 for testing.

Instead of using the raw snapshots for DL, they were preprocessed to obtain the deep scattering spectrum (DSS) features [35]. The DSS can not only produce features that are locally translation-invariant and stable to deformations, but also capture scale interactions and higher-order statistics. Its computation is done by a multi-layer architecture that resembles a CNN, but it does not require training.

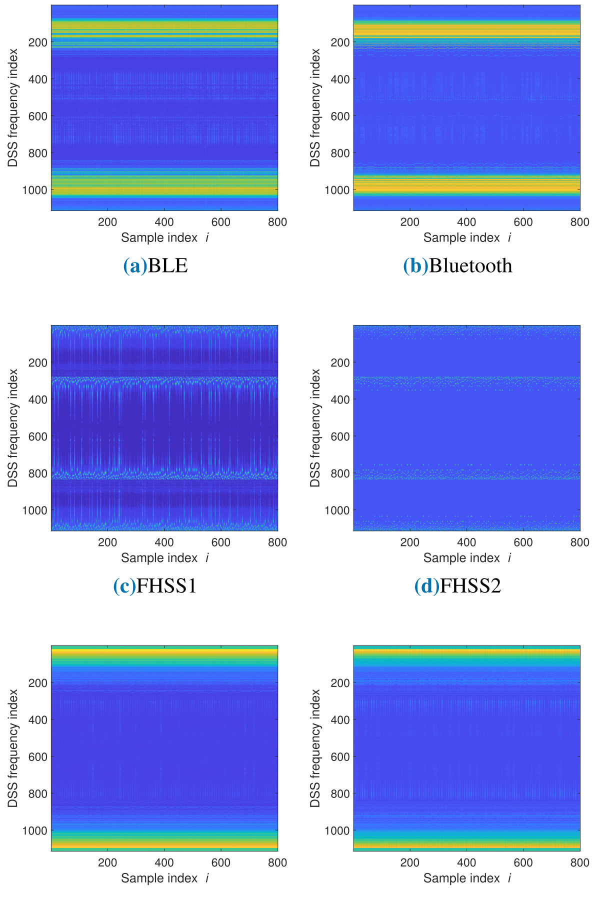
As the baseband RF samples are complex, the DSS features were constructed for the real and the imaginary parts separately. Let $\bar{x}_j(t)$ denote the real part of the j-th RF snapshot $x_j(t)$. Let $\{\psi_{\alpha_1}(t)\}_{\alpha_1 \in \mathcal{A}_1}$ be a set of analytic wavelet filters. That is, $\psi_{\alpha_1}(t)$ is a bandpass filter centered at frequency α_1 with bandwidth α_1/Q_1, where Q_1 is the number of wavelets per octave, and $\mathcal{A}_1$ the set of center frequencies. Then, with a lowpass filter $\phi(t)$ with bandwidth $1/T$, the first layer output of the DSS is simply a lowpass-filtered version of the input signal, given by

$$S_0(\bar{x}_j) := \bar{x}_j(t) * \phi(t). \quad (48)$$

Stable and locally translation-invariant features are obtained by taking the moduli of the wavelet filter bank outputs, followed by lowpass filtering, as in

$$S_1(\bar{x}_j, \alpha_1) := |\bar{x}_j(t) * \psi_{\alpha_1}(t)| * \phi(t). \quad (49)$$

The second layer output of DSS due to $\bar{x}_j(t)$ is the collection $\{S_1(\bar{x}_j, \alpha_1)\}_{\alpha_1 \in \mathcal{A}_1}$. To recover the information lost due to the



(a)BLE (b)Bluetooth

(c)FHSS1 (d)FHSS2

(e)Wi-Fi1 (f)Wi-Fi2

FIGURE 1. DSS of different RF signal classes. The second layer outputs in the DSS frequency index range [279, 836] are seen to be quite useful for discrimination.

lowpass filtering, one can go deeper with more layers. For example, the next layer can be defined based on another set of wavelets $\{\psi_{\alpha_2}(t)\}_{\alpha_2 \in \mathcal{A}_2}$ with Q_2 wavelets per octave as

$$S_2(\bar{x}_j, \alpha_1, \alpha_2) := ||\bar{x}_j(t) * \psi_{\alpha_1}(t)| * \psi_{\alpha_2}(t)| * \phi(t). \quad (50)$$

This process can be repeated with more layers. In this work, we used up to S_2. Thus, the DSS is defined as the collection of $\overline{M} := 1 + |\mathcal{A}_1| + |\mathcal{A}_1||\mathcal{A}_2|$ signals given by

$$S(\bar{x}_j) := [S_0(\bar{x}_j), \{S_1(\bar{x}_j, \alpha_1)\}_{\alpha_1 \in \mathcal{A}_1}, \{S_2(\bar{x}_j, \alpha_1, \alpha_2)\}_{\alpha_1 \in \mathcal{A}_1, \alpha_2 \in \mathcal{A}_2}]. \quad (51)$$

By sampling the analog signals in (51) at the Nyquist rate, the discrete-time DSS can be obtained as a matrix $\overline{\mathbf{X}}_j \in \mathbb{R}^{\overline{M} \times \overline{N}}$. Likewise, one can extract the DSS from the imaginary part $\tilde{x}_j(t)$ to construct $\widetilde{\mathbf{X}}_j \in \mathbb{R}^{\overline{M} \times \overline{N}}$. Define the concatenation $\widehat{\mathbf{X}}_j := [\overline{\mathbf{X}}_j^\top, \widetilde{\mathbf{X}}_j^\top]^\top \in \mathbb{R}^{M \times \overline{N}}$, where $M = 2\overline{M}$. The input vectors $\{\mathbf{x}_i\}$ to the DL algorithms are the columns of $\mathbf{X} := [\widehat{\mathbf{X}}_1, \widehat{\mathbf{X}}_2, \ldots, \widehat{\mathbf{X}}_J] \in \mathbb{R}^{M \times N}$, where J is the number of RF snapshots, and $N = J\overline{N}$.

In our experiment, $Q_1 = 16$, $Q_2 = 0.05$, and $T = 100$ ms were used. A Morlet wavelet was employed for the bandpass filters and Gabor for the lowpass filter. Overall, the dimension of $\widehat{\mathbf{X}}_j$ turns out to be $M = 1,114$ and $\overline{N} = 8$.

In Fig. 1, the sample DSS of the 6 classes are depicted using 100 snapshots for each class. In each DSS plot, the vertical axis represents the DSS frequency index. Index 1 corresponds to S_0, indices $2 \sim 278$ to S_1, and indices $279 \sim 557$ to S_2, for the real part $\bar{x}_j(t)$. Likewise, index $1,114$ represents S_0, indices $1,113 \sim 837$ to S_1, and $836 \sim 558$ to S_2, for the imaginary part $\tilde{x}_j(t)$. It is interesting to see that highly discriminative features are often extracted through S_2, as is the case, for instance, when discriminating between Wi-Fi1 and Wi-Fi2.

B. CLASSIFICATION STRATEGIES

In this section, the strategies employed for classifying the mixture components are described. For a given test snapshot $\widehat{\mathbf{X}} \in \mathbb{R}^{M \times \overline{N}}$, one first needs to perform sparse coding using the trained dictionary $\mathbf{D}$. One way to do this is to solve

$$\min_{\hat{\mathbf{Z}}} \|\widehat{\mathbf{X}} - \mathbf{D}\hat{\mathbf{Z}}\|_2^2 + \bar{\lambda}\|\hat{\mathbf{Z}}\|_1 \quad (52)$$

where $\bar{\lambda}$ may be different from the value of λ used for training, and thus is tuned separately. An alternative approach that does not require tuning of $\bar{\lambda}$ is a Bayesian sparse coding method, such as the relevance vector machine (RVM) [36]. The test results of both approaches are reported in Sec. V-E.

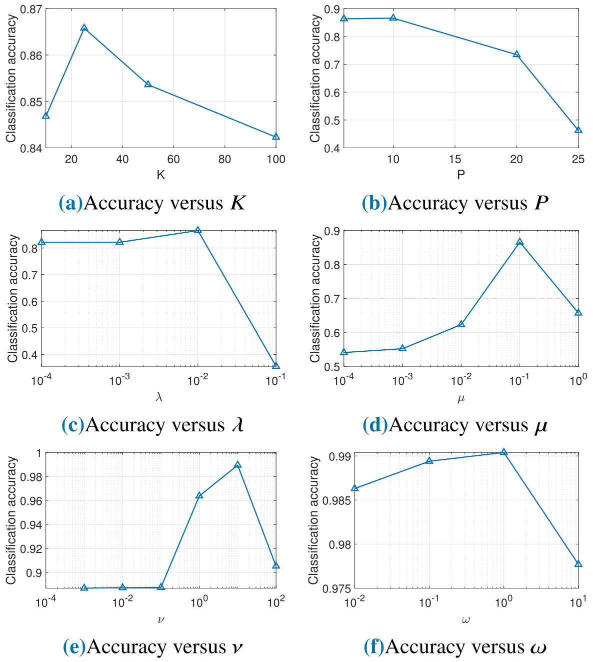
The resulting sparse code matrix $\hat{\mathbf{Z}}$ is of dimension $K \times \overline{N}$. In order to impart with temporal invariance (i.e., we do not care at which time point the discriminative features appear), average pooling is done to form a summary $\hat{\mathbf{z}} := \overline{N}^{-1} \sum_{n=1}^{\overline{N}} \hat{\mathbf{z}}_n$. Then, the discriminant variable $\hat{\mathbf{y}}$ is obtained as $\hat{\mathbf{y}} := \mathbf{W}^\top \hat{\mathbf{z}}$.

Employing $\hat{\mathbf{y}}$ as the input feature, we tested different classifiers, namely, the logistic regression (LR), and the neural network (NN) classifiers, as well as the matched filter (MF) and the zero-forcing (ZF) equalizer. The LR classifier is a linear classifier. With parameters $\{\boldsymbol{\beta}_c\}_{c=1}^{C-1}$, the posterior class probabilities are computed as [37]

$$\pi_c^{LR} := \frac{\exp(\boldsymbol{\beta}_c^\top \hat{\mathbf{y}})}{1 + \sum_{c'=1}^{C-1} \exp(\boldsymbol{\beta}_{c'}^\top \hat{\mathbf{y}})}, \quad c = 1, 2, \ldots, C-1 \quad (53)$$

and $\pi_C^{LR} := 1 - \sum_{c=1}^{C-1} \pi_c^{LR}$. Given that the mixture contains L components, the class labels for the components can be determined as the classes corresponding to the L largest values of $\{\pi_c^{LR}\}_{c=1}^{C}$. If L is not known, one can adopt a simple strategy of thresholding π_c^{LR} to detect the presence of individual signal classes. That is, the c-th class is detected when $\pi_c^{LR} \geq \theta$, for $c = 1, \ldots, C$. The NN classifier is a nonlinear classifier, constructed in our experiments using 10 hidden layers with sigmoidal nonlinearity, followed by the cross-entropy loss as the training objective. The LR and NN classifiers are machine learning algorithms, whose parameters are trained using the (non-mixture) training data.

On the other hand, MF and ZF detectors were developed originally in the detection theory framework. The MF detector correlates the input $\hat{\mathbf{y}}$ with the class centroids $\{\mathbf{W}^\top \mathbf{m}_c\}$ to



(a) Accuracy versus K

(b) Accuracy versus P

(c) Accuracy versus λ

(d) Accuracy versus μ

(e) Accuracy versus ν

(f) Accuracy versus ω

FIGURE 2. Results for parameter tuning. Coarse grid searches were performed based on the classification accuracy using 5-fold cross-validation.

obtain $\boldsymbol{\xi}_{MF} := (\mathbf{W}^\top \mathbf{M})^\top \hat{\mathbf{y}}$. If the class centroids were truly orthogonal, then the MF detector would be able to null out the interference completely. Since the orthogonality is only approximate, one may improve the detection performance by employing a linear equalizer. The ZF equalizer applies the pseudo-inverse of the centroid matrix to get $\boldsymbol{\xi}_{ZF} := (\mathbf{W}^\top \mathbf{M})^\dagger \hat{\mathbf{y}}$. Thus, the interference signals are nulled out via linear projection. The class labels of the L components are determined based on the L largest absolute values of the elements in $\boldsymbol{\xi}_{MF}$ or $\boldsymbol{\xi}_{ZF}$. Note that the MF and ZF detectors are obtained without any separate training. When L is not available, one can again adopt the thresholding strategy. For example, in the ZF case, the class probabilities are first obtained from the transformation

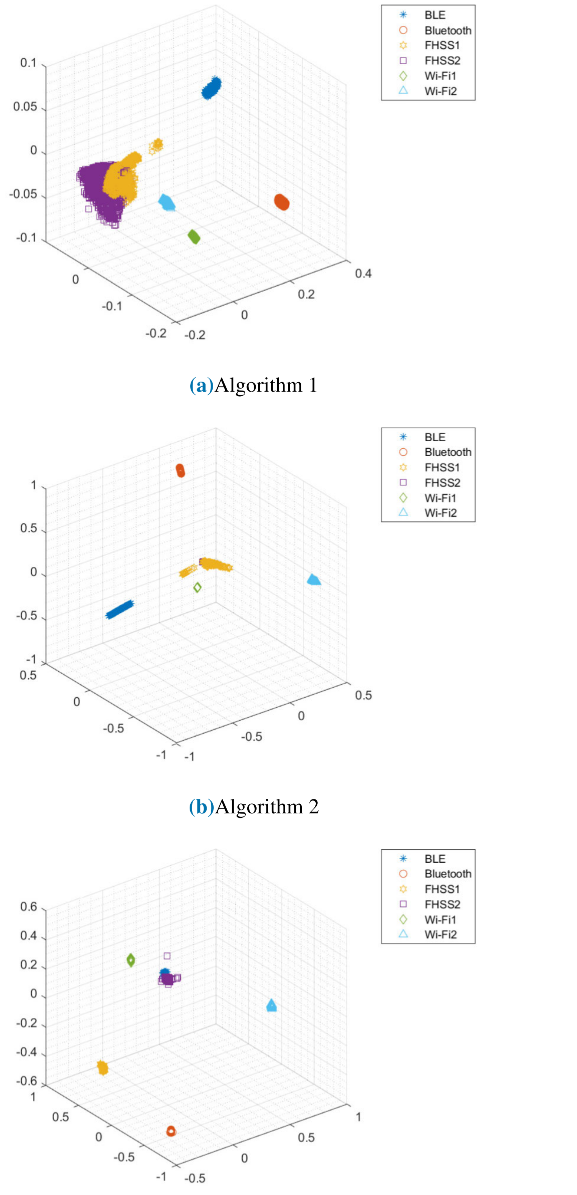
$$\pi_c^{ZF} := \frac{\exp(\xi_c^{ZF})}{\sum_{c=1}^{C} \exp(\xi_c^{ZF})}, \quad c = 1, 2, \ldots, C \tag{54}$$

where ξ_c^{ZF} is the c-th component of $\boldsymbol{\xi}_{ZF}$. The class-c signal is detected if $\pi_c^{ZF} \geq \theta$. The MF detection is done similarly.

C. PARAMETER TUNING

The DL formulations contain parameters that need to be tuned, namely, K, P, λ, μ (for (P1), (P2), and (P3)), ν (for (P2) and (P3)), and ω (for (P3)). We performed coarse grid searches based on 5-fold cross-validation using the classification accuracy as the metric.

For tuning (P1)'s parameters, the k-nearest neighbor (kNN) classifier was used to obtain the classification accuracies on the validation data. The kNN classifier is a nonparametric classifier, which does not assume any parametric model of the classification boundaries [37]. In our experiments, five nearest neighbors of $\hat{\mathbf{y}}$ in Euclidean distance were found from the training data features $\{\mathbf{y}_i\}_{i=1}^N$. Then, the label was predicted as the majority label of the five. We added white Gaussian noise with a signal-to-noise power ratio (SNR) of -20 dB to the DSS, since, without noise, the classification accuracy always turned out to be 100%. The tuned number of atoms K was 25, and the dimension of the feature P was 10. The accuracy curves for varying K and P values are depicted in Figs. 2(a) and 2(b), respectively. Similarly, tuning of the



(a) Algorithm 1

(b) Algorithm 2

(c) Algorithm 3

FIGURE 3. MDS of the learned features. The 3-D embeddings of $\{\mathbf{y}_i = \mathbf{W}^\top \mathbf{z}_i\}$ are depicted. (a) The FHSS features from Algorithm 1 do not seem to cluster well. (b) This is mitigated by Algorithm 2, but some classes exhibit elongated point clouds. (c) Algorithm 3 obtains well-separated and spherical point clouds.

remaining parameters yielded $\lambda = 10^{-2}$ and $\mu = 10^{-1}$, as shown in Figs. 2(c) and 2(d), respectively.

For tuning the parameters of (P2), we used a small number of mixture signals containing $L = 2$ equal-power components. This is necessary since the optimal ν would be 0 if tuned with non-mixture samples, as the orthogonality penalty always impairs the performance in the non-mixture case. Specifically, 100 snapshots for each of the $\binom{6}{2} = 15$ mixture types were generated, resulting in $1,500$ snapshots in total. The ZF classifier was used to get the accuracies. Retaining $K = 25$ from the tuning of (P1) for simplicity, we performed the grid search over $P \in \{6, 10, 20, 25\}$, $\lambda \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$, $\mu \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$, and $\nu \in \{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^{2}\}$. The tuned values are $P = 20$, $\lambda = 0.1$, $\mu = 10^{-3}$, and $\nu = 10$. The tuning curve for ν is shown in Fig. 2(e).

For the parameters in (P3), all the parameters were fixed at the values obtained for (P2) for simplicity, except the new parameter ω. A 1-dimensional search over ω found the best value $\omega = 1$, as shown in Fig. 2(f).

D. VISUALIZATION OF LEARNED FEATURES

An intuitive way to understand the differences of the proposed DL formulations is to visualize the learned features. The classical multidimensional scaling (MDS) is employed to visualize the P-dimensional discriminant vectors $\{\mathbf{y}_i = \mathbf{W}^\top \mathbf{z}_i\}$ in the 3-dimensional Euclidean space. Fig. 3 shows the MDS results for Algorithms 1–3. From Fig. 3(a), it can be seen that the point clouds are mostly well separated thanks to Fisher criterion, except for FHSS1 and FHSS2. It turns out that the representations for FHSS signals occupy higher dimensions (require more atoms) than the other classes, and the learned features do not cluster well. When Algorithm 2 is used, the features become approximately orthonormal, which can be observed to some degree in Fig. 3(b). Note that orthogonality in a higher dimensional space can never be accurately depicted in the 3-dimensional visualization. On the other hand, some of the class clouds such as for BLE and FHSS1 are seen to be quite elongated in certain directions, which may hurt the classification performance. Using Algorithm 3, it can be seen from Fig. 3(c) that the distribution of the features have become much more spherical.

E. TEST RESULTS

1) TESTS WITH NON-MIXTURE SIGNALS

First, the performance of the proposed algorithms for classifying non-mixture (single-label) signals was evaluated. Fig. 4 shows the classification accuracies at different SNR levels. Circularly symmetric complex Gaussian noise was added to the time-domain snapshots $x_j(t)$, before computing the DSS. The classification was done using a 5-NN classifier. The sparsity parameter $\bar{\lambda}$ in (52) was tuned using 5-fold cross-validation. Different values of $\bar{\lambda}$ were allowed at different SNR levels. The accuracies were computed on the test data set using the 5 dictionaries trained from 5 different training/cross-validation data splits. A robust average of the 5 accuracies was calculated, by throwing away the maximum and the minimum values before averaging. Note that the DL training was done using samples without any noise added (i.e. at SNR $= \infty$), but tested at various SNRs. It can be seen from Fig. 4 that the accuracies for Algorithms 2 and 3 are comparable to that of Algorithm 1 at high SNRs, but slightly worse at lower SNR levels. However, as we will see next, they have much more robust performance in the mixture case.

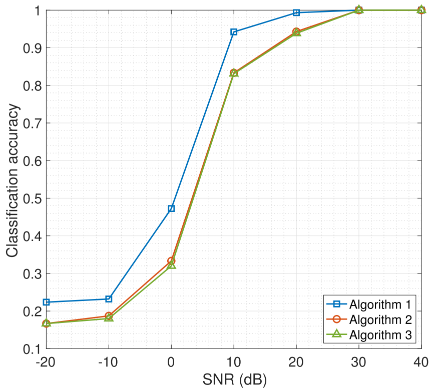
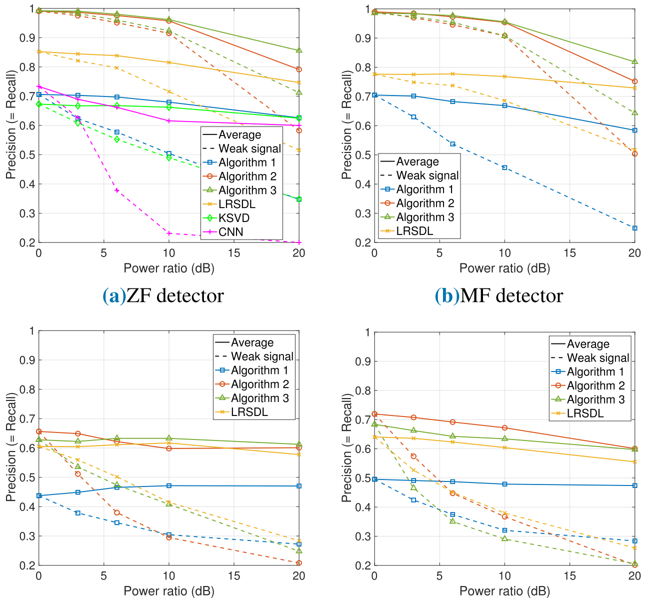


FIGURE 4. Classification accuracies in the non-mixture case. Training was done at SNR = ∞. Algorithms 2 and 3 perform worse than Algorithm 1 as they incorporate additional feature shaping constraints useful for the mixture case.



(a)ZF detector

(b)MF detector

(c)LR classifier

(d)NN classifier

FIGURE 5. Classification performance for $L = 2$. Algorithms 2 and 3 perform robustly with the ZF or MF strategies. Algorithm 3 further improves the performance in the near-far situations especially for the weak components.

2) TESTS WITH MIXTURE SIGNALS

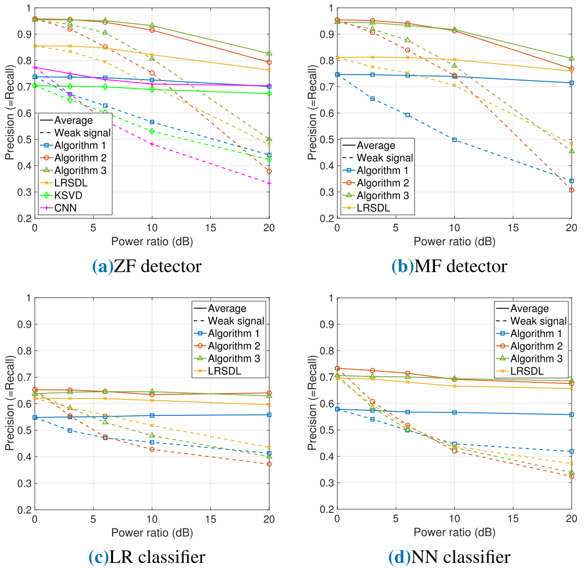
Next, the algorithms were tested using the mixture signals. First, assume that the number L of the components in the mixture is known. In this case, the number of false positive (FP) predictions (that is, predicting a signal to be present when it is actually absent) becomes the same as the number of false negative (FN) predictions (predicting absence when in fact present). Thus, the precision, which is defined as the number of true positives (TPs) over the total positives (TP + FP), becomes equal to the recall, which is defined as $\frac{\text{TP}}{\text{TP+FN}}$. Moreover, the accuracy, defined as the total correct predictions (both TPs and true negatives (TNs)) over the total predictions (TP+FP+FN+TN), becomes simply $1 - \frac{2L}{C}(1 - \text{Precision})$. Thus, we focus on the precision as our performance metric in the ensuing discussion.

The test data set was generated by linearly combining multiple single-label signals in the time domain. Fig. 5 depicts the precision metric when $L = 2$ signals are present in the mixtures. The signals were combined using a certain power ratio. For instance, a power ratio of 20 dB means that one component signal is 20 dB stronger than the other component. A total of 30 different types of mixtures were generated at each power ratio by considering $\binom{6}{2} = 15$ signal combinations and 2 different ways of picking the strong/weak components. For each type, 100 snapshots were generated for testing the classification performance. In Fig. 5, each panel shows the performance of one of the four classification strategies, namely, ZF, MF, LR, and NN classifiers, at different power ratios that range from equal power (0 dB power ratio) to more near-far situations (up to 20 dB power ratio). The solid curves represent the robust precision averaged across both the strong and the weak components. The dashed curves depict the precisions of only the weak component.

a: PERFORMANCE OF THE PROPOSED ALGORITHMS

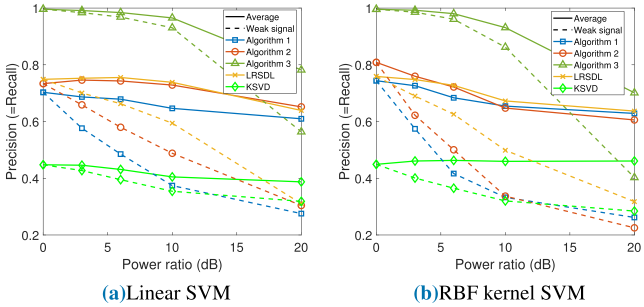
In Fig. 5, the square, circle, and triangle markers indicate Algorithms 1, 2, and 3, respectively. It can be seen that the performance of Algorithm 1 is severely degraded by the mixture signals, regardless of the classification strategies or the power ratios. In other words, the features from Algorithm 1 are not robust against mixture classification, although they achieve good performance in the non-mixture case [cf. Sec. V-E1]. Interestingly, the performance of Algorithm 2 based on the ZF and the MF strategies is seen to remain robust to the mixture signals. This is because the feature centroids from Algorithm 2 are almost orthonormal to one another, allowing the ZF/MF filters to effectively null out the interfering signals. On the other hand, the LR and the NN classifiers are seen not as robust, since they do not exploit the inherent orthogonality in the features. Comparing the LR and the NN classifiers, the NN improves the performance slightly thanks to its nonlinear decision boundaries.

The performance of Algorithm 3 stays similar to that of Algorithm 2 when the power ratio is close to unity. However, in the near-far situation, Algorithm 3 with the ZF or the MF detectors provides significant improvement in the classification performance. It should be noted that the improvement is particularly pronounced for the weak signals. For example, in Fig. 5(a), using the ZF detector at the power ratio 20 dB, the overall improvement from Algorithm 2 to Algorithm 3 is around 8%, whereas for the weak signals, the improvement is as high as 22%. It is also worth noting from Figs. 5(a) and (b) that the ZF detector performs slightly better than the MF detectors. This is because the ZF can further mitigate the leakages from imperfect orthogonality among the features.



(a)ZF detector (b)MF detector
(c)LR classifier (d)NN classifier

FIGURE 6. Performance for $L = 3$. The trend is similar to the $L = 2$ case.



(a)Linear SVM (b)RBF kernel SVM

FIGURE 7. Performance with SVM classifiers when $L = 2$. It is interesting to see that the features from Algorithm 3 are robust with SVMs.

We repeated the experiment for the case of $L = 3$. When the power ratio was not 0 dB, one strong component and two weak components were created, where the two weak ones were at the same power level. Fig. 6 shows the resulting classification performance. One can see a trend similar to the $L = 2$ case. That is, Algorithms 2 and 3 exhibit robust performances while Algorithm 1 degrades sharply. The ZF and the MF detection strategies work much better than the LR or the NN counterparts. Algorithm 3 improves upon Algorithm 2 when the power ratio is high, especially for the weak signals.[1]

[1] Compared to Fig. 5, it can be seen that the performances of Algorithms 2 and 3 are slightly worse in $L = 3$ than in $L = 2$, which is expected since correctly predicting the presence of three signals is harder than two. However, it is also noted that the performance of Algorithm 1 is improved when $L = 3$ compared to $L = 2$, which can be explained as follows. Consider a random prediction strategy, which picks L classes out of C candidates at random. The performance of this strategy when $L = 2$ can be shown to be $\frac{1}{3}$, while with $L = 3$, it actually becomes $\frac{1}{2}$. The performance of Algorithm 1 is already quite bad. Thus it seems it is affected by this higher baseline when $L = 3$.

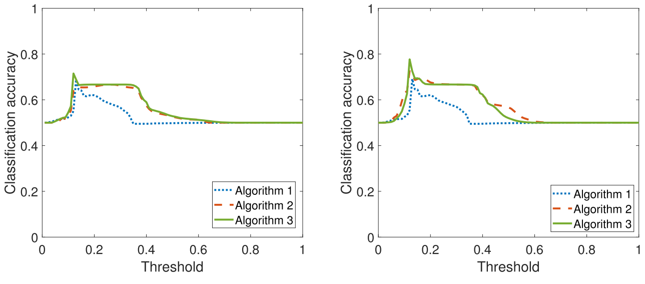
b: COMPARISON WITH THE EXISTING METHODS

Next, the performance of the proposed algorithms is compared to that of the existing DL and recent CNN-based methods. First, the basic K-SVD algorithm [16] and the low-rank shared dictionary learning (LRSDL) algorithm are considered [27], [38]. Since the ideas in other discriminative DL algorithms, such as LC K-SVD [39], discriminative K-SVD [19], DLSI [40] and DL-COPAR [41], were already incorporated and compared to LRSDL in [38], we do not explicitly compare with those.

In Figs. 5 and 6, the LRSDL performance for mixture classification is shown. In Figs. 5(a) and 6(a), we also added the curves for K-SVD followed by ZF detection. The inputs to LRSDL and K-SVD algorithms are the DSS as before. Since the classification strategies proposed in [38] is not suitable for multi-label classification, the ZF, MF, LR, and NN strategies are again employed. Also, since the performances of K-SVD combined with MF, LR, and NN classifiers are inferior to the ZF case, they are not shown to reduce the clutter. For a fair comparison, the number of atoms K in K-SVD algorithm was fixed to 25, which is the same as the size used for our proposed methods. Similarly, the dictionary for LRSDL contains 24 discriminative atoms and 3 common atoms, yielding 27 in total, which is close to our number 25.

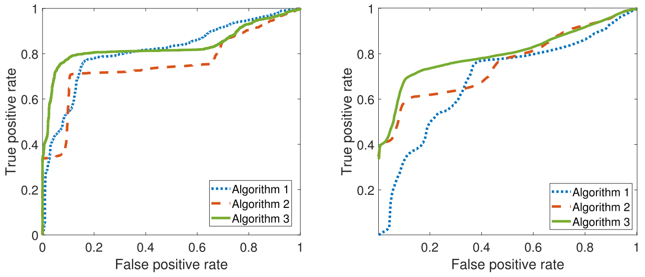
From Fig. 5(a), it is seen that when $L = 2$, K-SVD performs as well as Algorithm 1, but much worse than LRSDL, Algorithms 2 and 3. This is because K-SVD does not obtain discriminative features but learns the dictionary based on a reconstruction criterion. A similar observation can be made for $L = 3$ in Fig. 6(a). As for LRSDL, it can be seen from Fig. 5 that the performance is the best with the ZF detection. In Fig. 5(a), LRSDL is seen to outperform Algorithm 1. This is expected since Algorithm 1 does not employ an orthogonality constraint, while the LRSDL formulation actually includes a penalty term related to a notion of orthogonality. That is, for a class-c sample, the portion explained by the class-c' sub-dictionaries, $c' \neq c$, is encouraged to be small. Still, LRSDL is inferior to Algorithms 2 and 3, since the learned features in LRSDL are not constrained to be orthonormal, unlike the formulations for Algorithms 2 and 3. This is important for mixture signal classification, since the leakage from stronger signals can still affect the weaker components unless they are of approximately the same "power" in the feature domain. A similar conclusion can be drawn from Fig. 6 when $L = 3$.

We also tested the algorithms with the support vector machine (SVM) classifier instead of the ZF, MF, LR, or NN classifiers. Since there are six classes, six binary classifiers were trained in a one-versus-all-others fashion, again using non-mixture samples. Both linear and kernel SVMs were tested. In Fig. 7(a), it is seen that the K-SVD followed by the linear SVM does not perform well, as K-SVD does not yield discriminative features. It is also noted that Algorithms 1 and 2, as well as LRSDL, perform much worse than with the ZF classifier. This is because the SVM does not anticipate the mixture signals, whereas the ZF or the MF classifiers are designed with the mixture signals in mind. It is interesting to see that Algorithm 3 performs quite well with SVMs. The use of a radial basis function (RBF) kernel improves the performances slightly in Fig. 7(b), except for Algorithm 3, for which linear classification seems to work better.



(a)Sparse coding based on (52) (b)Sparse coding based on RVM

FIGURE 8. Classification accuracy with the ZF equalizer. The number $L = 3$ of the components is not known to the classifier. A class-c signal is detected when $\pi_c^{ZF} \geq \theta$. Algorithm 3 achieves the best performance regardless of the sparse coding method.



(a)Sparse coding based on (52) (b)Sparse coding based on RVM

FIGURE 9. ROCs with the ZF equalizer. $L = 3$ is unknown to the classifier and the power ratio is 20 dB.

We also compared with a CNN architecture designed for RF signal classification in [42], where 2.4 GHz-band Wi-Fi, Zigbee, Bluetooth signals were classified, similar to our work. Instead of using the FFT-based features as the input to the CNN (which achieved the highest classification accuracy in [42] among other features), we used the DSS features as the input, for fair comparison. The outputs from the CNN are the class probabilities. In our experiments involving the mixtures of L different transmissions, the class labels corresponding to L highest probabilities were picked. The resulting performances are shown in Figs. 5(a) and 6(a), for $L = 2$ and $L = 3$, respectively. As the results show, the CNN trained using non-mixture samples is not robust for the mixture classification task. In particular, in the near-far situation, it can be seen that the CNN can barely detect the weak signals.

3) MIXTURE SIGNALS WITH UNKNOWN NUMBER OF COMPONENTS

In many situations, the number of the components in the mixture may not be available. As explained in Sec. V-B,

TABLE 4. AUC performances.

	$L=1$			$L=2$			$L=3$		
	Alg. 1	Alg. 2	Alg.3	Alg. 1	Alg. 2	Alg.3	Alg. 1	Alg. 2	Alg.3
ZF	0.8714	**1**	**1**	0.7592	**0.9977**	0.9973	0.6987	0.8544	**0.9066**
MF	0.9379	**1**	**1**	0.8765	0.9962	0.9970	0.8341	0.8453	0.9017
LR	0.9994	0.9960	**1**	0.7828	0.7927	0.7593	0.7016	0.6266	0.6538
NN	**1**	**1**	**1**	0.8495	0.9294	0.8587	0.7517	0.7660	0.7873

(a) AUC for the equal power case

	$L=2$			$L=3$		
	Alg. 1	Alg. 2	Alg.3	Alg. 1	Alg. 2	Alg.3
ZF	0.7319	0.8259	**0.8539**	0.6848	0.7552	**0.8008**
MF	0.7611	0.8304	0.8474	0.6998	0.7554	0.7861
LR	0.7697	0.7772	0.7663	0.6913	0.6969	0.6749
NN	0.766	0.7879	0.7891	0.6899	0.6994	0.7108

(b) AUC for the case with the power ratio 20 dB

a thresholding strategy can then be employed. Fig. 8 shows the detection performance of the proposed algorithms using the ZF detector. Only the ZF detection is shown as it was observed to achieve the best performance in Sec. V-E2. The test data set is the same as in Sec. V-E2 with $L = 3$, but this L is unknown to the detector. There are one strong component and two weak components, with the power radio set to 20 dB. Fig. 8 depicts the classification accuracy, defined as $\frac{\text{TP+TN}}{\text{TP+FP+FN+TN}}$, achieved at different threshold values θ. In particular, Fig. 8(a) shows the accuracies when the sparse coding is done based on (52), and Fig. 8(b) corresponds to employing the RVM for the sparse coding step. In the case of using (52), $\bar{\lambda}$ needs to be tuned manually. We tuned it as in Sec. V-E2 using a small set of mixture data with $L = 2$ equal-power components. It can be seen from Fig. 8 that the best performance is achieved by Algorithm 3 regardless of the sparse coding method. The RVM is seen to yield slightly better performance compared to the one using (52). This is reasonable since $\bar{\lambda}$ was tuned for $L = 2$ but the performance was tested on $L = 3$, whereas the RVM can automatically adjust the sparsity level. Fig. 9 shows the receiver operating characteristic (ROC) curves for the ZF detection. Again, it can be seen that Algorithm 3 shows the most robust performance. Interestingly, the ROC obtained using (52) seems to be better than the ROC from the RVM, presumably because the tuning of $\bar{\lambda}$ still provides useful information as the mismatch in L between the tuning and the testing is not large.

Table 4 summarizes the area under the ROC curve (AUC) values for the ROCs obtained using the ZF, MF, LR, and NN classifiers based on the RVM sparse coding, under various scenarios involving different L values and power ratios. When the detection performance is high, a high TP rate is achieved at a low FP rate, which would render a ROC curve pushed to the upper-left corner of the plot, leading to a high AUC value. Table 4(a) corresponds to the equal-power case with $L = 1, 2$ and 3, and Table 4(b) is for the case with the power ratio equal to 20 dB and $L = 2$ or 3. When $L = 1$, virtually all detectors are performing perfectly. However, when $L > 1$, Algorithm 1 is much degraded, while Algorithms 2 and 3 yield robust performances. It can be observed that Algorithm 3 significantly outperforms the rest in difficult scenarios, such as with $L = 3$ and the power ratio 20 dB, especially when employing the ZF detection scheme.

VI. CONCLUSION

RF signal classification algorithms based on data-driven feature learning have been developed, which can detect and classify the individual component signals when the observation is a mixture of concurrent transmissions. In order to reduce the data collection and training burden, the training was done using the single-label non-mixture samples, rather than the mixture samples. This was achieved by discriminative DL formulations that incorporated the labels in a Fisher discriminant cost, while shaping the learned features to be orthogonal across different classes and spherically distributed within each class. Exploiting the orthogonality, simple MF and ZF detectors were employed to effectively null out interfering class signals, much like the multiuser detection in wireless communication. The developed algorithms were tested with real wideband RF measurements to verify that the orthogonality and the sphering constraints significantly improved the robustness of classification, even in the challenging scenarios where there were large power differences among the component signals and the number of the components was not known a priori.

ACKNOWLEDGMENT

The authors would like to thank Tom Chatt at Lockheed Martin for collecting the RF measurement data.

REFERENCES

[1] S. K. Jayaweera, *Signal Processing for Cognitive Radios*. Hoboken, NJ, USA: Wiley, 2015.

[2] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *Proc. Int. Conf. Eng. Appl. Neural Netw.*, Aberdeen, U.K., Aug. 2016, pp. 213–226.

[3] 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.

[4] Z. Zhang, C. Wang, C. Gan, S. Sun, and M. Wang, "Automatic modulation classification using convolutional neural network with features fusion of SPWVD and BJD," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 5, no. 3, pp. 469–478, Sep. 2019.

[5] M. Schmidt, D. Block, and U. Meier, "Wireless interference identification with convolutional neural networks," in *Proc. IEEE 15th Int. Conf. Ind. Inform. (INDIN)*. Emden, Germany, Nov. 2017, pp. 180–185.

[6] Y. Pan, S. Yang, H. Peng, T. Li, and W. Wang, "Specific emitter identification based on deep residual networks," *IEEE Access*, vol. 7, pp. 54425–54434, 2019.

[7] K. Youssef, L. Bouchard, K. Haigh, J. Silovsky, B. Thapa, and C. V. Valk, "Machine learning approach to RF transmitter identification," *IEEE J. Radio Freq. Identificat.*, vol. 2, no. 4, pp. 197–205, Dec. 2018.

[8] 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, 2019.

[9] S.-J. Kim, E. Dall'Anese, J. A. Bazerque, K. Rajawat, and G. B. Giannakis, "Advances in spectrum sensing and cross-layer design for cognitive radio networks," in *Academic Press Library in Signal Processing*, R. Chellappa and S. Theodoridis, Eds. Waltham, MA, USA: Academic, 2014, ch. 9, pp. 471–502.

[10] Z. Li, Z. Yang, C. Song, C. Li, Z. Peng, and W. Xu, "E-eye: Hidden electronics recognition through mmWave nonlinear effects," in *Proc. 16th ACM Conf. Embedded Netw. Sensor Syst.*, Shenzhen, China, Nov. 2018, pp. 68–81.

[11] 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, USA, Nov. 2008, pp. 1–5.

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

[13] C.-H. Lee and W. Wolf, "Blind signal separation for cognitive radio," *J. Signal Process. Syst.*, vol. 63, no. 1, pp. 67–81, Apr. 2011.

[14] G. Tsoumakas, I. Katakis, and L. Vlahavas, "Random k-labelsets for multilabel classification," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 7, pp. 1079–1089, Jul. 2011.

[15] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 8, pp. 1819–1837, Aug. 2014.

[16] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[17] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *J. Mach. Learn. Res.*, vol. 11, pp. 19–60, Mar. 2010.

[18] J. Mairal, F. Bach, and J. Ponce, "Task-driven dictionary learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 791–804, Apr. 2012.

[19] Q. Zhang and B. Li, "Discriminative K-SVD for dictionary learning in face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, San Francisco, CA, USA, Jun. 2010, pp. 2691–2698.

[20] M. Yang, L. Zhang, X. Feng, and D. Zhang, "Fisher discrimination dictionary learning for sparse representation," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 543–550.

[21] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *Proc. 4th Int. Conf. Learn. Represent.*, San Juan, Puerto Rico, Jan. 2016, pp. 1–16.

[22] T. Angles and S. Mallat, "Generative networks as inverse problems with scattering transforms," in *Proc. 6th Int. Conf. Learn. Represent.*, Vancouver, BC, Canada, Apr./May 2018, pp. 1–10.

[23] J. G. Proakis, *Digital Communications*, 3rd ed. Singapore: McGraw-Hill, 1995.

[24] M. Honig, U. Madhow, and S. Verdú, "Blind adaptive multiuser detection," *IEEE Trans. Inf. Theory*, vol. 41, no. 4, pp. 944–960, Jul. 1995.

[25] S. Verdú, *Multiuser Detection*. Cambridge, U.K.: Cambridge Univ. Press, 1998.

[26] H. Chen, S.-J. Kim, and T. Chatt, "Discriminative dictionary learning for mixture component detection with application to RF signal recognition," in *Proc. 52nd Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, Oct. 2018, pp. 835–839.

[27] T. H. Vu and V. Monga, "Learning a low-rank shared dictionary for object classification," in *Proc. IEEE Int. Conf. Image Process.*, Phoenix, AZ, USA, Sep. 2016, pp. 4428–4432.

[28] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. Hoboken, NJ, USA: Wiley, 2012.

[29] S. Li and Y. Fu, "Learning robust and discriminative subspace with low-rank constraints," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 11, pp. 2160–2173, Nov. 2016.

[30] P. Tseng, "Convergence of a block coordinate descent method for non-differentiable minimization," *J. Optim. Theory Appl.*, vol. 109, no. 3, pp. 475–494, Jun. 2001.

[31] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, 2009.

[32] P. H. Schönemann, "A generalized solution of the orthogonal procrustes problem," *Psychometrika*, vol. 31, no. 1, pp. 1–10, Mar. 1966.

[33] R. Lai and S. Osher, "A splitting method for orthogonality constrained problems," *J. Sci. Comput.*, vol. 58, no. 2, pp. 431–449, 2014.

[34] S.-Y. Kung, "Discriminant component analysis for privacy protection and visualization of big data," *Multimedia Tools Appl.*, vol. 76, no. 3, pp. 3999–4034, Feb. 2017.

[35] J. Andén and S. Mallat, "Deep scattering spectrum," *IEEE Trans. Signal Process.*, vol. 62, no. 16, pp. 4114–4128, Aug. 2014.

[36] M. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, pp. 211–244, Jan. 2001.

[37] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. New York, NY, USA: Springer, 2001.

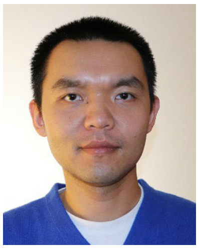
[38] T. H. Vu and V. Monga, "Fast low-rank shared dictionary learning for image classification," *IEEE Trans. Image Process.*, vol. 26, no. 11, pp. 5160–5175, Nov. 2017.

[39] Z. Jiang, Z. Lin, and L. S. Davis, "Label consistent K-SVD: Learning a discriminative dictionary for recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2651–2664, Nov. 2013.

[40] I. Ramirez, P. Sprechmann, and G. Sapiro, "Classification and clustering via dictionary learning with structured incoherence and shared features," in *Proc. IEEE Conf. CVPR*, San Francisco, CA, USA, Jun. 2010, pp. 3501–3508.

[41] S. Kong and D. Wang, "A dictionary learning approach for classification: Separating the particularity and the commonality," in *Proc. Eur. Conf. Comput. Vis.*, Florence, Italy, Oct. 2012, pp. 186–199.

[42] M. Kulin, T. Kazaz, I. Moerman, and E. D. Poorter, "End-to-end learning from spectrum data: A deep learning approach for wireless signal identification in spectrum monitoring applications," *IEEE Access*, vol. 6, pp. 18484–18501, 2018.



HAO CHEN (Graduate Student Member, IEEE) received the B.S. degree in electronics and information technology from the Nanjing Agricultural University, Nanjing, China, in 2011, and the M.S. degree in electrical engineering from the University of Cincinnati, Cincinnati, OH, USA, in 2014. He is currently pursuing the Ph.D. degree in electrical engineering with the University of Maryland at Baltimore County, Baltimore, MD, USA. His current research interests include machine learning and signal processing.

SEUNG-JUN KIM (Senior Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from Seoul National University, Seoul, South Korea, in 1996 and 1998, respectively, and the Ph.D. degree in electrical engineering from the University of California at Santa Barbara, Santa Barbara, CA, USA, in 2005. From 2005 to 2008, he was with NEC Laboratories America, Inc., Princeton, NJ, USA. From 2008 to 2014, he was with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN, USA, where his final title was a Research Associate Professor. Since 2014, he has been with the Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, where he is currently an Associate Professor. His research interests include statistical signal processing, machine learning, and optimization, with applications to wireless communication and networking, future power systems, and data science. He is serving as a Senior Area Editor for the IEEE Signal Processing Letters.