

Signal Detection and Classification in Shared Spectrum: A Deep Learning Approach

Wenhan Zhang, Mingjie Feng, Marwan Krunz, and Amir Hossein Yazdani Abyaneh

Dept. Electrical & Computer Engineering, University of Arizona, Tucson, AZ
 {wenhanzhang, mingjiefeng, krunz, yazdaniabyaneh}@email.arizona.edu

Abstract—Accurate identification of the signal type in shared-spectrum networks is critical for efficient resource allocation and fair coexistence. It can be used for scheduling transmission opportunities to avoid collisions and improve system throughput, especially when the environment changes rapidly. In this paper, we develop deep neural networks (DNNs) to detect coexisting signal types based on In-phase/Quadrature (I/Q) samples without decoding them. By using segments of the samples of the received signal as input, a Convolutional Neural Network (CNN) and a Recurrent Neural Network (RNN) are combined and trained using categorical cross-entropy (CE) optimization. Classification results for coexisting Wi-Fi, LTE LAA, and 5G NR-U signals in the 5-6 GHz unlicensed band show high accuracy of the proposed design. We then exploit spectrum analysis of the I/Q sequences to further improve the classification accuracy. By applying Short-time Fourier Transform (STFT), additional information in the frequency domain can be presented as a spectrogram. Accordingly, we enlarge the input size of the DNN. To verify the effectiveness of the proposed detection framework, we conduct over-the-air (OTA) experiments using USRP radios. The proposed approach can achieve accurate classification in both simulations and hardware experiments.

Index Terms—Machine learning, signal classification, coexistence, convolutional neural networks, recurrent neural networks, dynamic spectrum access, software-defined radio.

I. INTRODUCTION

The unprecedented demand for wireless services has crowded the radio spectrum. Spectrum sharing (SS) provides a potential remedy for the operation of heterogeneous wireless systems in congested and contested spectrum environments [1]. In the opportunistic (hierarchical) form of SS, unlicensed users can use the common spectrum when licensed (primary) users are idle [2]. However, it is challenging to guarantee a given throughput level to secondary systems without knowing the usage statistics and access behavior [3], [4]. Furthermore, some SS scenarios involve heterogeneous systems that cannot coordinate with one another [5], [6]. In these scenarios, operators and end users need to sense the shared medium and identify the active signals without having to decode them.

Fast and accurate identification of the different types of coexisting signals has always been an important aspect of interleaving spectrum sharing. Common spectrum sensing techniques are likelihood or feature-based [7], [8], e.g., cyclostationarity. In these techniques, signal detection is performed under certain assumptions of the underlying waveforms, e.g.,

their modulation and coding scheme, protocol behavior, probability distributions, etc., which strongly depend on the decoded signal. In addition to relying on specific model-based assumptions, conventional sensing approaches often assume that spectrum dynamics are slowly varying. To address the above limitations, we propose a machine learning (ML) framework for signal sensing and classification in coexistence scenarios with fast-varying spectrum dynamics.

Deep learning has been successfully applied to various classification and recognition problems [9]–[11]. It can support high-dimensional inputs, sizeable neural network models, and adjustable parameters. In addition to traditional speech and image recognition, deep neural networks (DNNs) have also been applied to RF signal classification problems [12]–[15]. However, most of these works attempt to classify the data according to their modulation schemes, or they assume the signal type to be highly related to such schemes. This approach does not compute the dynamics of a typical wireless protocol, where the same signal type may have several modulation and coding schemes. Moreover, existing efforts presume the received signal is only corrupted with noise but does not consider the possibility of mixed (superposed) signals, as in a concurrent transmission scenario over an unlicensed band. As a result, it is not possible to achieve satisfactory classification performance in these scenarios.

To investigate the application of DNNs in signal classification of coexisting waveforms, we focus on Wi-Fi, LTE LAA, and 5G NR unlicensed (NR-U), as an example of SS over an unlicensed band. We design a convolutional neural network (CNN) and a recurrent neural network (RNN) to detect the underlying wireless signal based on the received I/Q samples. CNN captures the sample features through the convolution calculations, while the RNN can capture the sequence dependency structure when the previous layer's outputs are flattened. Fig. 1 describes a receive chain with a signal (or protocol) classifier. After receiving an RF signal, the signal is filtered and down-converted into IF. By applying an ADC, the waveform is then converted to the digital domain. The designed protocol classifier can distinguish the signal type for the user after the DSP block processes the signal.

To test the model accuracy, we start with an interleaving SS scenario, whereby any but only one of the three coexisting technologies is active at a time, i.e., there is no concurrent transmissions of different signal types. We adjust the channel

noise as a way to control the SNR. The average classification accuracy for the interleaving scenario is about 92.1% when the SNR is greater than 15 dB. Motivated by such results, we then extend our testing to mixed signals, i.e., concurrent transmissions are allowed. The classifier can correctly distinguish the mixture with an accuracy of around 80% when the channel is in good conditions. To further improve the accuracy of the proposed DNN classifier, we incorporate frequency-domain analysis (FDA) in its input. A short-time Fourier transform (STFT) technique is applied to segments of the I/Q samples and used to obtain their spectrogram. Based on the STFT, the DNN can extract time-frequency features by employing a sliding Kaiser–Bessel window. Such information can be combined with the original DNN input (time-domain I/Q samples), resulting in significant improvement in accuracy.

We validate the performance of the proposed classifier using hardware experiments on an software-defined radio (SDR) platform. Over-the-air (OTA) experiments are conducted with USRP radios. The signals are generated by the *Matlab Communication Toolbox* and the *5G Toolbox*, and then fed into a USRP transmitter. We set three USRP-2921 radios to transmit Wi-Fi, LTE, and 5G NR signals, respectively. During transmission, the signal source can be selected as active or inactive. Correspondingly, the signals are superposed with selected types and the channel noise when transmitted. The RF signal is then captured by the receiver and used for classification. Our contributions can be summarized as follows:

- We introduce a DNN-based classification algorithm for coexisting signals in a shared-spectrum environment. By adjusting the DNN parameters, we generate extensive datasets for Wi-Fi, LTE, and 5G NR waveforms, and use them for training and testing our classifiers.
- We integrate FDA into our CNN-LSTM framework to differentiate between various signal types. The performance of the resulting DNN-based classifier is compared with other methods from various perspectives.
- To improve the classification accuracy, STFT is applied to provide spectrogram information. The average algorithm accuracy is improved by around 14% when FDA is included.
- We verify the proposed classification techniques experimentally using OTA experiments with USRP radios. The classifier accuracy is shown to be 91% in our experiments when SNR is greater than 15 dB.

The rest of the paper is organized as follows. Section II presents related works in deep learning-based signal classification. Section III describes the generation of the dataset for simulations and experiments. In section IV, we introduce the DNN architecture that we designed for protocol classification. In Section V, we analyze the performance of the proposed model. Our OTA experimental results are explained in Section VI. Section VII concludes the paper.

II. RELATED WORK

Signal classification is a critical and challenging problem in SS scenarios. Correct detection allows networks coordinators

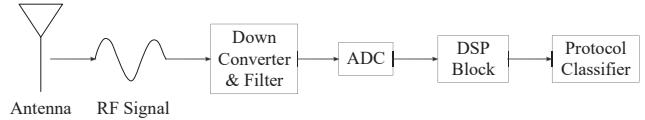


Fig. 1. Signal processing to obtain the I/Q samples for classification.

to mitigate interference between coexisting devices, which can improve the overall utilization of the network. The authors in [12] and [13] explored modulation classification problem using a deep learning (DL) algorithm. They proved that their approach can achieve more than 90% classification accuracy when the signal is in a high SNR regime. In [14], the authors studied the classification of modulation schemes for a distributed sensor network using deep learning models. A generative adversarial method was used in [15] to sort the data in a cognitive radio network. In [16], in-network users, out-network users, and jammers are distinguished by CNN to improve the in-network users' throughput and out-network users' success ratio. Unlike these works, our deep learning model targets signal protocol detection in a spectrum sharing environment. Compared with modulation schemes, the features of a protocol can be corrupted more easily and are harder to distinguish when a mixture of signals are present.

Besides their application in signal classification, neural networks have also been applied to many other time-series classification problems (e.g., [17]–[20]). It proves that neural networks have a higher degree of freedom and can achieve higher accuracy than some conventional methods. Other authors successfully employed DL methods in various wireless communications contents to improve the network efficiency (e.g., minimize congestion, improve throughput, simplify connection setup, etc.) [21]–[24]. In [25], DL was applied to coordinate multipoint downlink transmissions in 5G NR networks. The authors in [26] proposed a DL-based sharing structure for LTE and WiFi systems in the unlicensed 5 GHz bands. Their approach, however, assumes that different signal types are readily distinguished. In this paper, we provide a framework to detect the signal type without requiring any decoding even in a corrupted situation. In addition to traditional DNNs, we also include the FDA to discriminate between the signal types.

III. DATA GENERATION

The waveforms of the LTE-LAA, Wi-Fi, and 5G NR signals are generated using the *Matlab Communication Toolbox* and the *5G Toolbox*, which specifies a set of signal features, including the baseband I and Q values, channel bandwidth, modulation and coding scheme (MCS), subcarrier spacing, and allocated resource blocks. By knowing these features, we simulated sample waveforms of the three technologies under different parameter settings supported by the standards, as described in Table I. Of the various possible features, we consider the baseband I/Q samples at the receiver (with added noise) as input to the classifier. I/Q samples can be

easily obtained before decoding the signal, and they provide a rich representation of the actual waveform. Specifically, key parameters such as the MCS and the channel state information (CSI) are readily captured in the I/Q samples. The receiver architecture considered in this paper is shown in Fig. 1. The received RF signal is down-converted and digitized to obtain the raw I/Q sample set. The DSP block in Fig. 1 is used to further obtain the frequency-domain information from I/Q samples. Specifically, we employ short-time Fourier transform into the sequence analysis. By applying a sliding window, these samples are divided into multiple sequences, each consisting of 512 I/Q pairs. These sequences are used as the datasets to train and test various classifiers. Approximately, 100,000 of such segments were used, split into 80% for training and 20% for testing.

To demonstrate the viability of DL in classifying coexisting waveforms, we test the classification accuracy in both simulations and experiments. During the simulation, different types of signals are mixed under the same additive white Gaussian noise (AWGN) channel. For the hardware experiments, the baseband signals are up-converted before being transmitted over the air, so that the channel reflects the real noise of an indoor environment. We connect three USRP radios to the waveform generator, which allows us to transmit the three types of waveforms simultaneously. To distinguish all possible labels under different environments, we collect both independent and mixed (concurrent transmission) data. At the receiver, the waveforms are filtered and down-converted to I/Q samples that are then fed into the classifier. There are three types of independent signals, three types of double-mixtures, and one type of triple-mixture. For the multi-waveform scenario with concurrent transmissions, each component is transmitted with the same channel gain and the same SNR, so that they account for a roughly equal portion of the received signal. In summary, there are seven types of signals that need to be classified.

IV. NEURAL NETWORK ARCHITECTURE

In this section, we consider a combination of a CNN and an LSTM network to design our classifier, as shown in Fig. 2. The input is the segmented I/Q sequences, as described before. STFT is applied to each segment to obtain the power distribution of frequencies. Such frequency-domain data is passed to the convolutional layer along with the original I/Q value to allow the CNN capture certain features. After that, the output is fed to the pooling layer. Notice that information exchange between the convolutional layer and the pooling layer may take place several times in the structure. Such iterative process is also used in the dense layer. The output from the convolution and the pooling is then flattened to low dimensional data representation. This helps in extracting the time-series dependencies of the I/Q sequences in the bridged recurrent layers (LSTM layers). Eventually, the dense layers and a softmax layer are used to calculate the probability of each label and assign the most likely one to the given input.

CNNs are widely employed in visual imagery analysis due to the convenience in calculating the convolution for high-

TABLE I
PARAMETER OPTIONS FOR WAVEFORM GENERATOR.

Protocol	Parameter	Possible Values
LTE	Reference Channel	R.1, R.2, R.3, R.4, R.5, R.6, R.7, R.8, R.9, R.10, R.11, R.12, R.13, R.14, R.25, R.26, R.27, R.28, R.31-3A, R.31-4, R.43, R.44, R.45, R.45-1, R.48, R.50, R.51, R.6-27RB, R.12-9RB, R.11-45RB
	Number of Subframes	6, 8, 10
	Modulation Schemes	QPSK, 16QAM, 64QAM
	Transmission Bandwidth [RB]	1, 6, 15, 25, 27, 39, 50, 75, 100
	Duplex Mode	FDD, TDD
5G	Frequency Range	450 MHz-6 GHz, 24.25 GHz-52.6 GHz
	Subcarrier Spacing (kHz)	15,30,60
	Modulation Schemes	QPSK, 64QAM, 256QAM
	Channel Bandwidth (MHz)	5, 10, 15, 20, 25, 30, 40, 50
	Duplex Mode	FDD, TDD
Wi-Fi	Channel Coding	BCC, LDPC
	Modulation Schemes	BPSK, QPSK, 16QAM, 64QAM, 256QAM
	Guard Interval	Short, Long
	Channel Bandwidth (MHz)	20, 40, 80, 160

dimensional input. By contrast, RNNs are used more often to solve time-series forecasting problems. The LSTM network (one type of RNNs) combines the different weights of its input sequences and calculates the prediction value by some optimization function in a recurrent way. Therefore, the outcome of the recurrent neural network captures the dependency and correlations of the sequence. To take advantage of both networks, we introduce a customized neural network that allows recurrent layers to be connected to convolutional layers. This would reinforce time-series analysis in the CNN and can help capturing the dependency between samples. To train such a combined network, we collect I/Q samples with SNR from -10 dB to 20 dB with a step increment of 2 dB following the setting described before. At each SNR value, there are roughly 10000 samples from each signal type, where each sample includes the I/Q values of 512 data points x .

A. Convolutional Neural Network

A CNN typically consists of fully connected layers, pooling layers, convolutional layers, and a softmax layer. It can be trained by associating the input samples s_t and their corresponding label y_t , $t = 1, \dots, T$, from M labels. The labeled dataset can be defined as: $\mathcal{D} =$

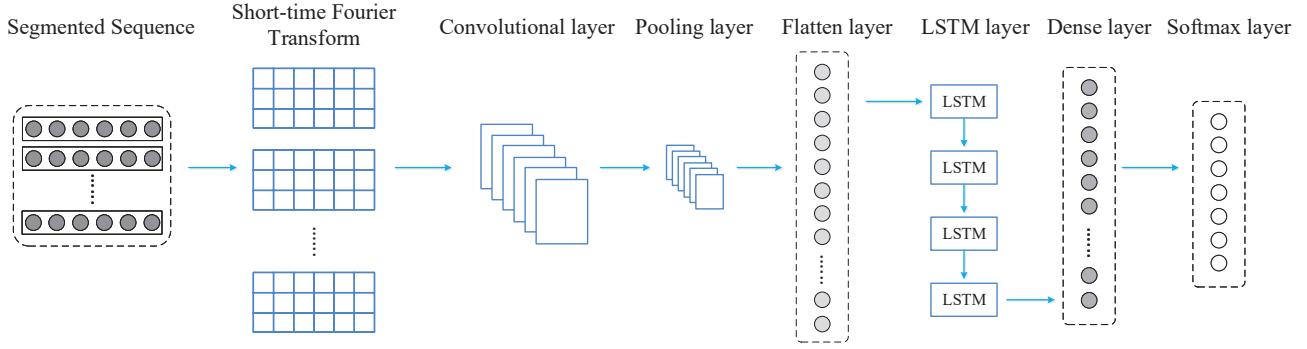


Fig. 2. Overview of the proposed classifier.

$\{(s_1, y_1), (s_2, y_2), \dots, (s_T, y_T)\}$. Let y_t be the t th sample's label, with $y_t \in L = \{l_1, l_2, \dots, l_M\}$, where l_r stands for the r th label. Under such a definition, we can train the hypothesis $\mathcal{H}(s_t) = \bar{y}_t$, which predicts the label \bar{y}_t of s_t that matches the actual one y_t .

For a sample s in a two-dimensional matrix, the convolutional layer \mathcal{V} is extracted by the convolutional kernel:

$$\mathcal{V}(\lambda)(\mu) = b(\lambda)(\mu) + \sum_{d=0}^D \sum_{h=0}^H \mathcal{K}(d)(h)s(\lambda + d)(\mu + h) \quad (1)$$

where D and H are the width and height of the convolutional kernel, \mathcal{K} is the filter kernel, and b is the bias. To avoid gradient vanishing, we also employ the scaled exponential linear unit (SELU) as an activation function in the convolutional layers:

$$\text{SELU}(u) = \lambda \begin{cases} p & \text{if } u > 0 \\ \alpha e^u - \alpha & \text{if } u \leq 0 \end{cases} \quad (2)$$

where u is the output of the convolutional layer; α and λ are constant values in the SELU setting.

After the feature is captured in the convolutional layer, it is then connected to the pooling layer. The pooling layer helps reduce computing costs by decreasing the dimensionality of the data. It also prevents over-fitting by providing an abstracted form of the representation. The results of the max-pooling are flattened and passed to the LSTM layer. We will discuss the detail of the LSTM layers in the followed section. After that, the output of the recurrent layer is passed through the dense layers. The dense layers and the softmax layer are fully connected. The softmax layer has 7 neurons, i.e., the output has 7 elements. This is because the types of signals that are being classified in this paper are: Wi-Fi, LTE, 5G, Wi-Fi+LTE, Wi-Fi+5G, LTE+5G, and Wi-Fi+LTE+5G. The probability P_{l_m} of classifying an input as label l_m , $m = 1, \dots, M$, is normalized to provide label prediction from the softmax layer:

$$P_{l_m} = \frac{e^{\hat{y}_m}}{\sum_{r=1}^M e^{\hat{y}_r}} \quad (3)$$

where \hat{y} is the output of dense layer. We let $P = [p_{l_1}, p_{l_2}, \dots, p_{l_M}]$. Our neural network can make a prediction

of the given input data x by the hypothesis \mathcal{H} under parameter setting θ : $\bar{y} = \mathcal{H}(\theta, x)$.

To measure the difference between an estimated label \bar{y} and the real label y , cross-entropy is introduced, which allows the neural network to capture this error. In our case, the cross-entropy function is used as the loss function during training:

$$\mathcal{L}(\theta) = - \sum_r \mathcal{B}_r \log(\bar{y}_r) \quad (4)$$

where $\{\mathcal{B}_r\}_{r=1}^m$ are binary variables; $\mathcal{B}_r = 1$ if the label r is correct among m categories and \bar{y}_r is the corresponding probability of the correction. The CNN minimizes the loss function during the training by calculating the gradient of θ at each step j . Then, θ gets updated with the corresponding learning rate η as follows:

$$\theta_j = \theta_{j-1} - \eta \nabla_{\theta} \mathcal{L}(\theta). \quad (5)$$

B. Recurrent Neural Network

A typical LSTM network consists of multiple LSTM cells. We explain the steps taken to calculate different parameter values of one LSTM cell. At each time step j , we define x_j as the system input and δ_j as the output of the LSTM cell. The cell output at the previous time step, δ_{j-1} , is then combined with the current system input x_j to form the input of the current cell. The state of the cell is C_j , which records the system memory and gets updated at each time step. To control information flow between cells, several gates are applied, including an input gate (i_j), an output gate (o_j), and a forget gate (f_j). Each gate generates an output between 0 and 1, where the value of the output is calculated by a sigmoid (σ) function. An output of 0 indicates that the input of the gate is totally blocked, while an output of 1 indicates all information of the input is kept in the cell. The input, output, and forget gates are calculated as follows:

$$\begin{aligned} i_j &= \sigma(W_i x_j + U_i \delta_{j-1} + b_i) \\ o_j &= \sigma(W_o x_j + U_o \delta_{j-1} + b_o) \\ f_j &= \sigma(W_f x_j + U_f \delta_{j-1} + b_f) \end{aligned} \quad (6)$$

where W_i , W_o , and W_f are weights assigned to the three gates; U_i , U_o , and U_f are the corresponding recurrent weights; and b_i , b_o , and b_f are the bias values of the three gates.

Similar to the gate function, we combine the current inputs and previous cell state C_{j-1} to update the cell state. The difference is that instead of a sigmoid, the inputs will be processed by a hyperbolic tangent function that generates a value output between -1 and 1 :

$$\tilde{C}_j = \tanh(W_c x_j + U_c \delta_{j-1} + b_c) \quad (7)$$

after the update, \tilde{C}_j is multiplied by the output of the input gate, which is then used as the first component to update the cell state. Another component for updating the cell state is the state of the previous cell, which is processed by the forget gate. This component determines how past data is to be utilized. With the two components, the cell state at time j is updated as follows:

$$C_j = f_j C_{j-1} + i_j \tilde{C}_j. \quad (8)$$

The output of the cell δ_j , which will be used at time $j+1$, is calculated by the multiplication of the output gate and the tanh function of the current cell state:

$$\delta_j = o_j \tanh(C_j). \quad (9)$$

C. Frequency-domain Analysis

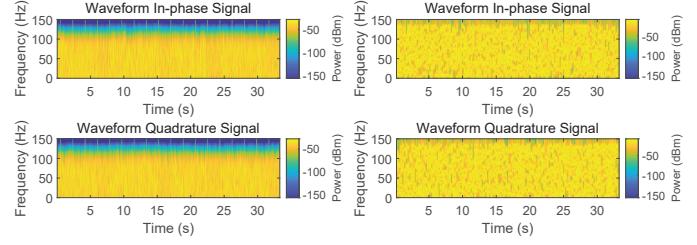
Because Wi-Fi, LTE, and 5G-NR transmissions show substantial similarity in time-domain, we augment our classification approach with the spectrogram, which allows these waveforms to be more distinguishable, as shown in Fig. 3. In such a spectrogram, a 'frequency' refers to the relative occurrence of components with specific power density. Frequency-domain analysis is commonly employed in image processing and wireless communications. It can be used to compensate for overlooked hidden information in neural networks when the input consists of a time-series of I/Q samples. FDA-based classification places more emphasis on the spectral characteristics of periodic patterns that are hard to capture by time-domain analysis. ANNs have the flexibility to accommodate different types of inputs, including spectrogram information. According, we apply Fourier transform to the data before feeding it along with the I/Q samples into the neural network.

Our FDA is applied to the baseband signal. Specifically, a short-time Fourier Transform (STFT) is applied to each I/Q sequence:

$$X(\tau, \omega) = \int_{-\infty}^{\infty} x(k)w(k-\tau)e^{-i\omega k}dk \quad (10)$$

where $x(k)$ is the time-domain signal to be transformed and $w(\tau)$ is the Kaisar-Bessel window function. $X(\tau, \omega)$ is essentially the Fourier transform of $x(k)w(k-\tau)$, which is a complex function representing the phase and magnitude of the signal over time and frequency. The Kaisar-Bessel window is used to extract the time-frequency features:

$$w(o) = \frac{I_0(\beta \sqrt{1 - (\frac{o-O/2}{O/2})^2})}{I_0(\beta)} \quad (11)$$



(a) Spectrogram for 5G NR signal, (b) spectrogram for Wi-Fi signal.

Fig. 3. Example of frequency-domain analysis for 5G NR and Wi-Fi signals.

where I_0 is the zeroth-order Bessel function of the first kind, O is the window length, and β is the shape factor, which can be determined by a side-lobe attenuation α . STFT is used to determine the sinusoidal frequency and phase content of local sections of a signal as it changes over time. It uses short sequence segments to analyze the spectrogram so that it matches well with our neural network training, which also divides a long signal sequence into several equal-length training samples. The original network is extended to include frequency information as another set of features. Unlike image classification, in time-domain I/Q samples classification, time-series features play an essential role in determining the classifier's performance. The amplitude fluctuation, phase change, and signal dependency pattern are hidden in the I/Q sequences, which cannot be replaced by the spectrogram.

V. PERFORMANCE EVALUATION

A. Benchmark Algorithm

We compare our classification algorithm with several traditional machine learning (ML) algorithms and also with individual CNN and LSTM classifiers. For the traditional ML algorithms, we consider the support vector machine (SVM) and random forests (RF). The traditional SVM is successfully applied in class separation by determining the decision boundary while maximizing the margin between classes. Because the classes in our signal classification problem are not linearly separable, we employ a soft margin in the SVM model to further improve its accuracy. By introducing slack variables ξ , the objective function can include multiplications of the soft margin. After applying a Lagrangian relaxation, we transform the above constrained problem into an unconstrained one. Furthermore, the objective function of the resulting unconstrained problem becomes a quadratic function. The optimal weight matrix is obtained during the training by solving the derivative of each mathematical operation in the objective function at every epoch. After such a procedure, the trained SVM is then tested. For RF methods, we evaluate the performance for different numbers of constituent trees. We compare our proposed methods with these ML methods to show the merits of our approach. Because the proposed approach includes both CNN and RNN architectures, the performance of independent CNN and RNN is also studied. Basically, the independent

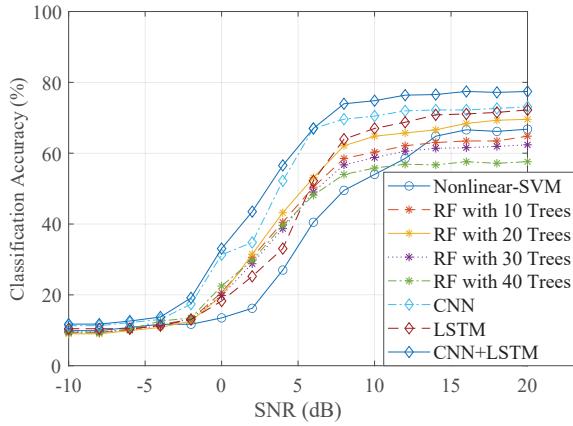


Fig. 4. Signal classification accuracy vs. SNR over an AWGN channel.

CNN model architecture is described as in section IV-A but without any LSTM layer. The independent RNN is the LSTM network defined in section IV-B, where the output is passed to the dense layers to generate the predicted label for the input data.

B. Classification based on I/Q Samples

In our simulations, we assume the signal is transmitted over an additive white Gaussian noise (AWGN) channel. In general, analytic feature extractors perform well under this assumption. Before introducing STFT, we first take the input as time-domain I/Q pairs. Data generation was described in Section III. During the testing phase, the received SNR was increased in steps of 2 dB. As can be observed in Fig. 4, the nonlinear-SVM has the worst accuracy when the SNR is less than 10 dB. The accuracy is less than 70% for such an algorithm. The accuracy for random forests increases sharply when the SNR goes from 0 to 10 dB. The performance is improved when the number of trees increases from 10 to 20, but seems to saturate for a larger number of trees. Accuracy can be optimized when the number of trees is around 20. The LSTM-based classifier does not perform well when the SNR is below 6 dB; however, its accuracy increases rapidly at higher SNR values. CNN can achieve similar accuracy to LSTM when SNR is higher than 14 dB. Furthermore, CNN performs even better than other algorithms under low SNRs. Our proposed architecture, which incorporates an LSTM layer into the CNN architecture, can further improve the classification performance. It achieves higher accuracy than CNN and LSTM; meanwhile, it behaves better in almost all SNR conditions.

The confusion matrix for the proposed combined neural network is depicted in Fig. 5. As shown in Fig. 5(a), the correct rate for single LTE, Wi-Fi, and 5G NR is more than 63%, while the accuracy for the mixed signal is around 50%. Similar results happen in Fig. 5(b) and Fig. 5(c), where the classifier can achieve higher classification accuracy for independent signals. The reason is that the independent signal features are more noticeable compared with the signal mixed with other types. When signals are under a coexisting environment, the

waveforms are corrupted with each other. Such corruption makes the amplitude and phase of the received I/Q samples deviate from the predetermined pattern. After the noise is introduced, it becomes even more difficult to distinguish these signals. Another observation is that the proposed approach can avoid the misdetection of highly similar types so that each type's misclassification is distributed evenly. By contrast, the classifiers like SVM and RF are harder to separate the categories when some types have higher closeness. Therefore, the false-positive rate would be much higher between LTE and 5G NR, for the closeness of them is greater than that of Wi-Fi. In our proposed algorithm, the false prediction doesn't concentrate on specific types, which means the combined neural network can distinguish the classes even though they are extremely comparable. Finally, from the plot, we can find that the classifier accuracy increases fast between 4 dB and 8 dB but slows down between 8 dB and 12 dB. This is because the influence the noise makes is almost negligible with very high SNR, and the signals have less improvement in purity when SNR is greater than 10 dB. As a result, the performance of the classifier becomes more steady in such an SNR range.

C. Impact of Frequency-domain Analysis

We then add the FDA into the I/Q samples, as introduced in Section IV-C. To be specific, STFT is applied to I/Q sequences, and the results are reflected as the spectrogram. Such frequency strength is also fed into the neural network along with the original I/Q pairs to train the model. The FDA is only based on the I/Q samples, so there is no extra information required for the input data. We then compare the performance of neural networks at 20 dB and summarize the results in Fig. 6. The classification accuracy is improved for all learning algorithms, proving that the FDA provides more information that regular machine learning algorithms can not obtain from the time-series input. The improvement is less evident for the RF algorithm, which means the influence of frequency strength may weaken under the trees. In fact, the RF algorithm makes a decision based on all the trees' predictions. However, spectrogram information can not guarantee to improve the performance of all of them. Thus, the effect is averaged among trees, which weakens the advance. For CNN, LSTM, and their combination, the accuracy is enhanced by approximately 15%. The proposed algorithm can achieve 93% accuracy after including spectrogram analysis.

1) *STFT Resolution*: STFT resolution quantitatively relates to the mainlobe width of the transform of the window. To analyze the influence of resolution, we capture a period of a received LTE signal with 10 sub-carriers and use it as an example for our following analysis. The computed power spectrum of the input shown in Fig. 7(a) visualizes the fraction of time that a particular frequency component is present in a signal. When time resolution T_{res} increased from 200 ms to 700 ms, both the power density and strength decrease in all frequencies. T_{res} controls the duration of the segments used to compute the short-time power spectra that form the spectrogram so that it decides the precision of signal energy

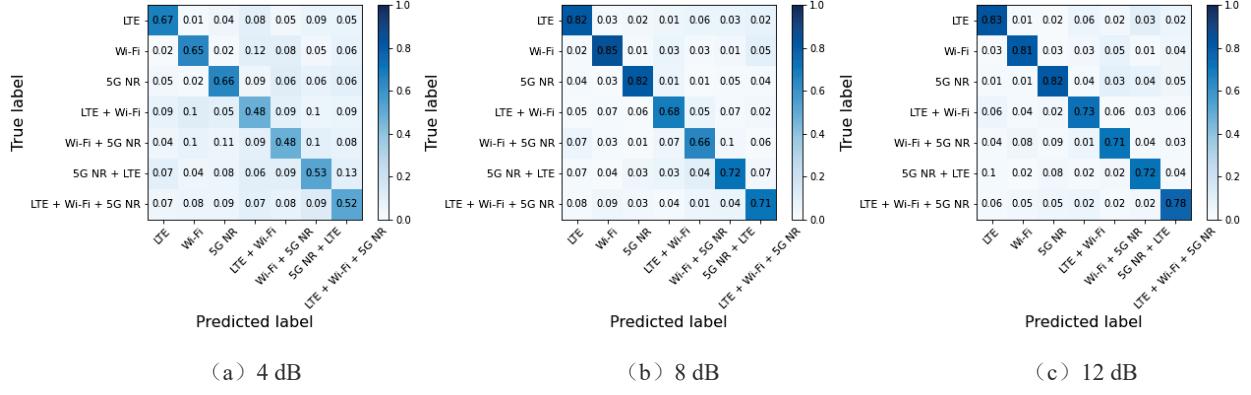


Fig. 5. Confusion matrices for superimposed signals without FDA.

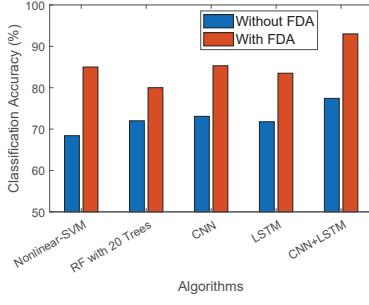


Fig. 6. Impact of FDA on different algorithms.

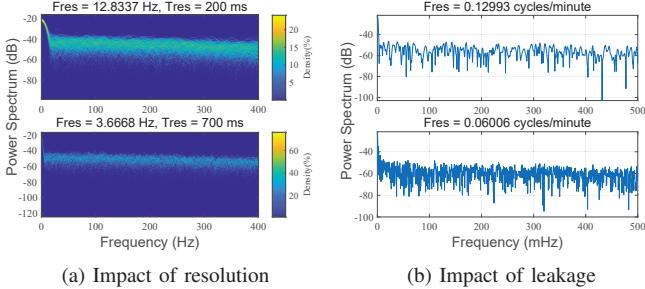


Fig. 7. Spectrogram under different STFT resolutions and spectral leakages.

distribution in the frequency space. T_{res} directly relates to the number of samples, which means if the system has a higher sample rate, the duration can be reduced. In our case, the signal power becomes less dominant with lower T_{res} . However, such fuzzy figures enlarge the spectrogram difference between different types of signals. As a result, the higher T_{res} can not guarantee better classifier performance. During the test, we find the proposed classifier can achieve the best accuracy when the T_{res} is around 600 ms. The results are as described in Table II.

2) *Spectral Leakage:* Spectral leakage occurs when a non-integer number of periods of a signal is sent to the STFT. One reason for such leakage is that the spectrum is the convolution between frequency function and sample sequences, which inevitably creates new frequency components. These compo-

nents are directly affected by the spectral windowing function; thus, they are considered as results of spectral leakage. To control the spectral leakage, we introduce the leakage coefficient, which is a real numeric scalar between 0 and 1. It restraints the Kaiser window sidelobe attenuation, that is also relative to the mainlobe width. When adjusting such coefficients, the resolution frequency F_{res} changes correspondingly. When leakage is 1, F_{res} is 0.06006 cycles/minute, while, when leakage is 0.65, F_{res} becomes 0.12993 cycles/minute. As depicted in Fig. 7(b), the power spectrum records more changes in frequency when leakage is 1. Such details expand the diversity between signal types; however, these changes also include more noise for the classifier. As a result, leakage doesn't have the monotonous classification accuracy. By comparing different leakage settings as depicted in Table II, the proposed classifier performs better when leakage is between 0.6 and 0.8 during the simulation.

TABLE II
ACCURACY OVER TIME RESOLUTION AND SPECTRAL LEAKAGE.

Time Resolution (ms)	Accuracy	Spectral Leakage	Accuracy
100	89.3%	0.1	83.5%
200	90.2%	0.2	87.2%
300	90.8%	0.3	89.4%
400	91.3%	0.4	90.8%
500	92.1%	0.5	91.3%
600	93.4%	0.6	92.2%
700	93.0%	0.7	93.7%
800	92.6%	0.8	92.1%
900	91.7%	0.9	90.9%
1000	91.3%	1	88.6%

D. Impact of RNN Layer

The LSTM layer is one typical type of RNN layer, and we propose to investigate an integrated CNN-LSTM design that takes advantage of both CNNs and LSTM networks. CNN employs convolution layers to extract multidimensional data features and achieve accurate classification, but it cannot capture the dependency pattern in the data. In contrast, an LSTM network captures such dependency by recurring the previous cell states in the hidden layers. After we introduce

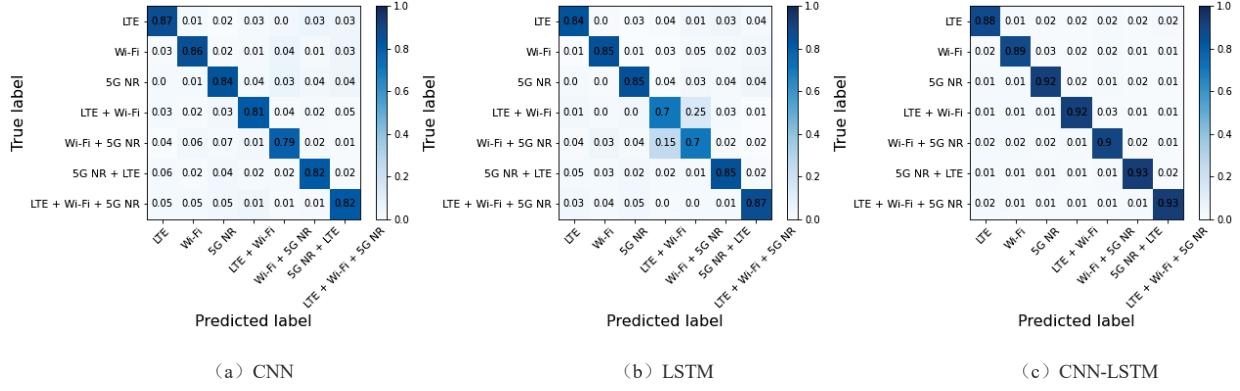


Fig. 8. Confusion matrices for superimposed signals with FDA.

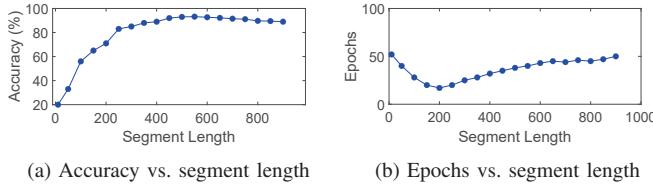


Fig. 9. Achievable accuracy and required epochs under different segment length.

such layers into the CNN, the performance improves as plotted in the confusion matrix. The overall accuracy improves from 87% when LSTM layers are connected to convolutional layers. In Fig. 8, the LSTM network can achieve similar accuracy with CNN. However, it misclassifies between LTE+ Wi-Fi and Wi-Fi+5G NR heavily. By combining CNN with LSTM, such misclassification is eliminated. The average accuracy is further improved to 92%.

E. Segment Length

The length of input sequences also influences the performance of the neural network. Fig. 9(a) describes the achievable accuracy trends with the range of segments. As we can see, the accuracy is only around 20% when the segment period is short, and it increases fast with the input duration until 400. The growth then slows down after that point. This is because more features that are hidden in the sequence can be captured when the segment is more extended, and the signal types are more distinguishable when the difference of the sequences is expanded due to its length. However, the performance is not ensured to be enhanced, for that too many features included can incur the over-fitting problem. It is reflected in Fig. 9(a) that the accuracy stops rapid raising and even reduces after the length is greater than 600.

Fig. 9(b) depicts the epochs required to achieve expected accuracy when the length of segments changes. When the segment period is short, the proposed network's performance fluctuates considerably and takes more epochs to be steady. It's due to the insufficient input sequence duration and the

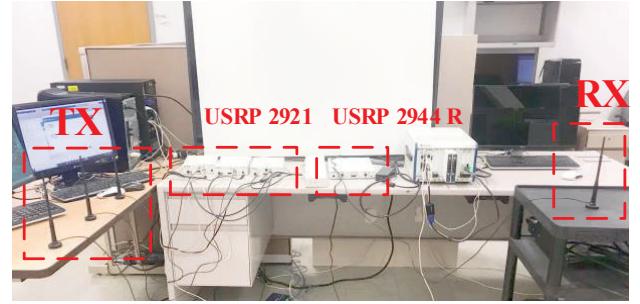


Fig. 10. Experiment setup used for performance evaluation.

low expected accuracy. With the length growing, the epochs reduced to around 20 when sequence duration is 200, which means the network can obtain a stable prediction condition with such length. Then the epochs increase because the neural network needs more training when the input is more extensive.

VI. EXPERIMENT

A. USRP Settings

We further evaluate the proposed classification model on a testbed consisting of three NI USRP-2921s and one NI USRP-2944R. Our indoor experiment setup is shown in Fig. 10, where the distance between the transmitter and receiver is roughly 2 meters, and each of them is equipped with 8 dBi antennas operating at the frequency of 5 GHz. The transmitters are synchronized by OctoClock CDA-2990 if transmitting different types of signals simultaneously. There are 500 Mbytes worth of experimental traces for 7 different classes of signals that are collected, namely Wi-Fi, LTE, 5G, Wi-Fi+LTE, Wi-Fi+5G, LTE+5G, and Wi-Fi+LTE+5G. In all these experiments, signals are transmitted at the center frequency of 5 GHz with a bandwidth of 20 MHz. The receiver has a gain of 30 dB and a sampling rate of 20 Msps centered at the 5 GHz center frequency, with collecting time equals to 250 ms. The Wi-Fi waveform is transmitted by generating baseband samples of 802.11 ac (VHT) with BPSK modulation

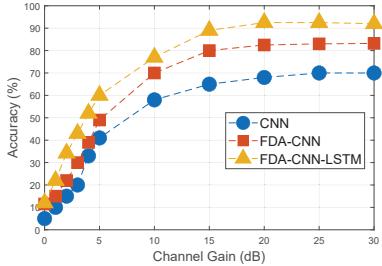


Fig. 11. Average accuracy vs. channel gain.

and 1/2 rate with a PSDU length of 1024 bytes, and it consists of 26080 samples. The LTE waveform is generated by downlink RMC with the reference channel of R.9, which has a 64 QAM modulation and is of size 250000. We also generate 5G waveforms using 5G DL FRC with QPSK modulation, a rate of 1/3 with a subcarrier spacing of 15 kHz, and a size of 250000. As a result, we gather I/Q samples for different transmission gains ranging from 0 – 30 dB and later use them for our DL processes.

B. Evaluation of The Proposed Integrated Approach

As depicted in Fig. 11, the achievable accuracy increases fast with channel gain until 15 dB. Then, it slows down and converges to a steady-state. We assume the noise power is at the same level when we control receiver gains. Thus, by adjusting the amplifier which controls the gain, the SNR changes accordingly. Similar to the previous simulation, the performance is hard to improve when the SNR reaches the bound. It's probable because the features have already been fully obtained, and the noise effects are neglectable after such a point. To compare the influence of the FDA, a basic CNN is used to predict the signal type. The proposed FDA approach can enhance the accuracy by more than 10%. It is due to the FDA that translates expand the I/Q samples from the time-domain to the frequency-domain, which includes more features into the inputs of the classifier and amplifies the difference between signal types. We also analyze the effect of the LSTM layer, and we find it can raise the accuracy by around 10% when channel gain is greater than 20 dB compared with the non-LSTM model. Besides, such RNN layers can improve the performance under all channel gains in our experiment.

VII. CONCLUSION

In this work, we develop deep neural networks to detect coexisting signal types by I/Q samples without having to decode them. With segmented sample sequences, the CNN is combined with the LSTM network and then trained. The classification result shows competitive accuracies by neural networks when the received signal is a mixture of signals transmissions of different wireless technologies. We then apply STFT on I/Q sequences to further improve the classification accuracy. Neural networks show considerable improvement after including the spectrogram information. Moreover, to

verify that the proposed detection framework is viable in a real environment, an OTA experiment is then conducted with USRP sets. The results show that the proposed deep neural architecture can achieve accurate classification in both simulations and experiments.

VIII. ACKNOWLEDGEMENTS

This research was supported in part by NSF (grants CNS-1563655, CNS-1731164, CNS-1813401, and IIP-1822071) and by the Broadband Wireless Access & Applications Center (BWAC). Any opinions, findings, conclusions, or recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of NSF.

REFERENCES

- [1] "IEEE standard for definitions and concepts for dynamic spectrum access: terminology relating to emerging wireless networks, system functionality, and spectrum management," *IEEE Std 1900.1-2019 (Revision of IEEE Std 1900.1-2008)*, vol. no., pp.1-78, 23 April 2019.
- [2] Q. Wang et al., "Robust large-scale spectrum auctions against false-name bids," *IEEE Transactions on Mobile Computing*, vol. 16, no. 6, pp. 1730-1743, June 2017.
- [3] Qualcomm, "Extending LTE advanced to unlicensed spectrum," *Qualcomm Incorporated White Paper*, pp. 1-12, Dec. 2013.
- [4] 3GPP, "Study on licensed-assisted access using LTE," *3GPP Work Item Description*, RP-141664, Sep. 2014.
- [5] K. Kosek-Szott, J. Gozdecki, K. Loziak, M. Natkaniec, L. Prasnal, S. Szott, and M. Wagrowski, "Coexistence issues in future WiFi networks," *IEEE Network*, vol. 31, no. 4, pp. 86-95, July 2017.
- [6] M. Hirzallah, W. Afifi and M. Krunz, "Full-duplex-based rate/mode adaptation strategies for Wi-Fi/LTE-U coexistence: a POMDP approach," *IEEE JSAC*, vol. 35, no. 1, pp. 20-29, Jan. 2017.
- [7] O. A. Dobre, A. Abdi, and a. W. S. Y. Bar-Ness, "Survey of automatic modulation classification techniques: classical approaches and new trends," *IET Communications*, vol. 1, no. 2, pp. 137-156, 2007.
- [8] W. C. Headley and C. R. C. M. d. Silva, "Asynchronous classification of digital amplitude-phase modulated signals in flat-fading channels," *IEEE Transactions on Communications*, vol. 59, no. 1, pp. 7-12, January 2011.
- [9] J. Nam, K. Choi, J. Lee, S. Chou and Y. Yang, "Deep learning for audio-based music classification and tagging: teaching computers to distinguish rock from bach," *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 41-51, Jan. 2019.
- [10] M. Kim, B. Cao, T. Mau and J. Wang, "Speaker-independent silent speech recognition from flesh-point articulatory movements using an LSTM neural network," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 12, pp. 2323-2336, Dec. 2017.
- [11] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE CVPR'16*, Las Vegas, NV, June 2016, pp. 770-778.
- [12] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," In *Proc. International conference on engineering applications of neural networks*, 2016, pp. 213-226, Springer.
- [13] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-air deep learning based radio signal classification," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 168-179, Feb. 2018.
- [14] S. Rajendran, W. Meert, D. Giustiniano, V. Lenders and S. Pollin, "Deep learning models for wireless signal classification with distributed low-cost spectrum sensors," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 3, pp. 433-445, Sept. 2018.
- [15] B. Tang, Y. Tu, Z. Zhang and Y. Lin, "Digital signal modulation classification with data augmentation using generative adversarial nets in cognitive radio networks," *IEEE Access*, vol. 6, pp. 15713-15722, 2018.
- [16] Y. Shi, K. Davaslioglu, Y. E. Sagduyu, W. C. Headley, M. Fowler and G. Green, "Deep learning for RF signal classification in unknown and dynamic spectrum environments," in *Proc. IEEE DySPAN'19*, Newark, NJ, USA, 2019, pp. 1-10.

- [17] M. G. Baydogan, G. Runger and E. Tuv, "A bag-of-features framework to classify time series," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2796-2802, Nov. 2013.
- [18] J. Long, E. Shelhamer and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. CVPR'15*, Boston, MA, 2015, pp. 3431-3440.
- [19] F. Karim, S. Majumdar, H. Darabi and S. Chen, "LSTM fully Convolutional networks for time series classification," *IEEE Access*, vol. 6, pp. 1662-1669, 2018.
- [20] F. Karim, S. Majumdar and H. Darabi, "Insights into LSTM fully convolutional networks for time series classification," *IEEE Access*, vol. 7, pp. 67718-67725, 2019.
- [21] A. H. Y. Abyaneh, M. Hirzallah and M. Krunz, "Intelligent-CW: AI-based framework for controlling contention window in WLANs," in *Proc. IEEE DySPAN'19*, Newark, NJ, USA, 2019, pp. 1-10.
- [22] A. Berian, I. Aykin, M. Krunz and T. Bose, "Deep learning based identification of wireless protocols in the PHY layer," in *Proc. IEEE ICNC*, Big Island, HI, USA, 2020, pp. 287-293.
- [23] A. Albanna and H. Yousefi'Zadeh, "Congestion minimization of LTE networks: a deep learning approach," *IEEE/ACM Transactions on Networking*, vol. 28, no. 1, pp. 347-359, Feb. 2020.
- [24] X. Du, H. Van Nguyen, C. Jiang, Y. Li, F. R. Yu and Z. Han, "Virtual relay selection in LTE-V: a deep reinforcement learning approach to heterogeneous data," *IEEE Access*, vol. 8, pp. 102477-102492, 2020.
- [25] F. B. Mismar and B. L. Evans, "Deep learning in downlink coordinated multipoint in new radio heterogeneous networks," *IEEE Wireless Communications*, vol. 8, no. 4, pp. 1040-1043, Aug. 2019.
- [26] J. Tan, L. Zhang, Y. Liang and D. Niyato, "Intelligent sharing for LTE and WiFi systems in unlicensed bands: a deep reinforcement learning approach," *IEEE Transactions on Communications*, vol. 68, no. 5, pp. 2793-2808, May 2020.