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

***Abstract*—Accurate identification of the signal type in a shared-spectrum scenario is critical for fair allocation of channel access. It facilitates scheduling the transmissions of coexisting systems (e.g., Wi-Fi, LTE LAA, and 5G NR-U) and avoids collisions, especially when spectrum dynamics change rapidly. In this work, we develop deep neural networks (DNNs) to detect coexisting signal types based on In-phase/Quadrature (I/Q) samples without decoding. 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. The classification results show high accuracy for the proposed DNN, even when the received signal is under the mixture. We then exploit spectrum analysis of I/Q sequences to further improve the classification accuracy. By applying Short-time Fourier Transform (STFT), additional information in the frequency domain can be gained. 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 RF signal is transmitted between antennas and is then captured to evaluate the classification performance of our approach.**

***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 access (SA) provides a potential remedy for the operation of heterogeneous wireless systems in congested and contested spectrum environments [1]. In SA, unlicensed users can use the common channels when licensed users are idle [2]. However, it is challenging to guarantee a given throughput level to coexisting systems without knowing their usage statistics and access behavior [3], [4]. Furthermore, there are active demands for heterogeneous networks to share spectrum and avoid collisions when uncoordinated service convoluted [5], [6]. To fulfill such requirements, 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 break the above limitations, we propose a machine learning 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 input data, 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 work attempt to classify the data according to their modulation schemes, or they assume the signal category to be highly related to such schemes. It overwhelms the dynamics in a SA network, where one type of signal protocol 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) unlicensed bands. As a result, it is not possible to maintain satisfactory performance in the fast varying SA network.

To investigate the application of DNNs in signal classification of coexisting waveforms, we focus on Wi-Fi, LTE LAA, and 5G NR unlicensed, as for example, the case of operating over the unlicensed 5GHz bands. We design a convolutional neural network (CNN) and a recurrent neural network (RNN) to detect the underlying wireless technology based on the received I/Q samples. The CNN can capture the sample features by the convolution calculation, while the RNN can gain the sequence dependency when the feature is flattened. Fig. 1 describes the procedures before where we can apply the signal classifier. After the transmitted RF signal is received by the antenna, it should be filtered and down-converted into the analog signal. By applying an ADC, the waveform is then converted to digital signals. 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 SA scenario, whereby any but only one of the three techniques is active at a time, i.e., there is no superposition of different types except for channel noise. The average classification accuracy for the interleaving scenario can achieve $92.1\%$ when the SNR is greater than 15 dB. Motivated by such results, we then extend our effects to under mixed signals. The classifier can correctly distinguish the mixture signal with an accuracy of around $80\%$ when the channel is in good condition. To further improve the accuracy of the proposed DNN classifier, we incorporate the frequency-domain analysis

(FDA) in its input. A short-time Fourier transform (STFT) technique is applied to segments of the I/Q samples and 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 experiments on an software-defined radio (SDR) platform. Over-the-air (OTA) experiments are conducted with USRP sets. The signals are generated by the *Matlab communication toolbox* and *5G toolbox* and then connected to the USRP. We set three USRP-2921 with antennas correspond to the signal types Wi-Fi, LTE, and 5G NR. During the transmission, signals can be selected as active or inactive by the control of USRP. Correspondingly, the signals are superposed with selected types and with the channel noise when transmitted between antennas. The RF signal is then captured by the receiver and used for the classification. Our contributions are summarized as follows:

- We introduce a DNN-based classification algorithm for coexisting signals in a shared-spectrum scenario. By adjusting parameter settings, we generate extensive data set featured by signal types of Wi-Fi, LTE, and 5G NR and use them for training and testing classifiers.
- We propose an FDA-CNN-LSTM framework to differentiate between various signal types. The performance of the proposed DNN-based classifier is compared with other methods from various perspectives.
- To improve the classification accuracy, STFT is applied to provide extra spectrogram information. The average algorithm accuracy is improved by around $14\%$ when frequency-domain analysis is included.
- We verify the proposed experimentally detection framework with an OTA experiment conducted on USRP sets. The RF signals are transmitted and mixed between antennas and are then captured to evaluate the performance of our identification approach. The accuracy can achieve $91\%$ in our experiment when SNR is greater than 15 dB.

The rest of the paper is organized as follows. Section II presents the related work in deep learning-based signal classification. Section III describes the generation of the dataset for simulation and experiment. In section IV, we introduce the neural network architecture that we designed for protocol classification. In Section V, We analyze the performance of the proposed model. The details and results of the corresponding SDR experiment are explained in Section VI. Section VII concludes the paper.

## II. Related Work

Signal classification is a critical and challenging problem in a dynamic resource allocation network. Correct detection allows the coordinator to avoid the interference sources, which can actively improve the total network resources' efficiency. The authors explore the modulation scheme classification problem with deep learning (DL) algorithm in [12], [13] and prove it can achieve more than $90\%$ classification accuracy when the signal is in a high SNR environment. After that, the deep learning based signal classification problem become extensively popular. In [14], the distributed sensor network's modulation schemes are detected and classified by deep learning models. [15] shows the generative adversarial method can 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 work, our deep learning model is trained for signal protocol detection in a coexisting environment. Instead of the modulation scheme, the protocol features are easier corrupted and harder distinguishable by the mixture of other sources' signals. As a result, we apply a CNN and an RNN to characterize the pattern from the I/Q samples dynamically.

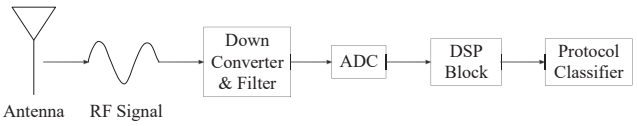



Fig. 1. Signal processing to obtain the I/Q samples for classification.

In addition to the previously mentioned signal classification scenario, neural networks have also been profitably applied to other time-series classification problems in [17]–[20]. It proves that neural networks have a higher degree of freedom and can achieve even higher accuracy than conventional methods. Other authors also successfully employed in other wireless communication areas [21]–[24] to help improve the network's efficiency (e.g., minimize congestion, improve throughput, and simplify the connection setup). In [25], deep learning is applied to coordinate multipoint downlink transmission in NR heterogeneous networks. [26] propose and deep learning based unlicensed bands sharing structure for LTE and WiFi system. However, these approaches ignore the importance of successfully detect the required signal type in coexisting environments. Moreover, it's even more arduous to recognize the signal in a spectrum access or heterogeneous network. In this paper, we provide a framework to detect the signal without requiring decoding them even in a corrupted situation. In addition to the traditional deep neural network, we also include the frequency analysis to discriminate the signal types accurately. We then build the SDR platform to make sure the model is also viable in a real environment. The testing result is impressive in both the simulation and the experiment.

## III. Dataset Generation

The waveforms of these signals were generated using the Matlab 5G and communication toolbox, which specifies a set of signal features, including the baseband In-phase and Quadrature values, channel bandwidth, modulation and coding scheme (MCS), subcarrier spacing, and allocated resource blocks. We generated sample waveforms of the three tech-

nologies based on different parameter settings supported by LTE, Wi-Fi, and 5G NR standards, as described in Table I. Of the various possible features, we considered the baseband I/Q samples at the receiver (noise added) as the 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. Accordingly, instead of taking all the signal parameters as inputs to the artificial neural network (ANN), we only rely on the measured I/Q samples, hence significantly reducing the size of the ANNs. 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. By applying a sliding window, these I/Q samples are divided into multiple sequences, each consisting of 512 I/Q pairs. These sequences are used as the data sets to train various classifiers. Approximately 100,000 such segments were used, split into 80% sequences for training and 20% sequences for testing.

To demonstrate the viability of the deep learning algorithm in the coexisting signal situation, we test the classification accuracy in both simulation and experiment. During the simulation, different types of signals are mixed under the same additive white Gaussian noise (AWGN) channel. While, during the experiment, the baseband signals are up-converted RF signals then transmitted over the air, so that the channel reflects the real noise of an indoor environment. We connect three USRP sets to the waveform generator, which allows us to transmit all three types of the waveforms simultaneously. To distinguish all possible labels under different environments, we collect both independent and mixture transmission data. At the receiver side, the antenna keeps receiving the RF signals. Then the waveforms are filtered and down-converted to I/Q samples and used for classifying. There are three types of independent signals, three types of double-mixtures, and one type of triple-mixture. For the multi-waveform situation, each component is transmitted with the same channel-gain, so that they take a roughly equal portion of the received commixture. As a result, there are seven types of the signal need to be categorized in the dataset.

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 |

## IV. NEURAL NETWORK ARCHITECTURE

In this section, we consider a CNN and an LSTM network as our classifier to distinguish the signal types, as shown in Fig. 2. The input is the segmented I/Q sequences, as we describe in Section III. After the input is fed, STFT is applied in these sequences to obtain the frequency power strength distribution. Such frequency-domain strength data is passed into the convolutional layer along with the original I/Q value to allow the CNN capture features. Then the output is connected to the pooling layer. Notice that the information exchange between the convolutional layer and the pooling layer may repeat several times in the structure, so we don't draw all the layers out. It happens in the dense layer as well. The output from convolution and pooling is then flattened to low dimensional data. It helps better extract the time-series dependencies from the sequences in the bridged the recurrent layers (LSTM layers). After that, the dense layer and softmax layer calculate the probability and assign the label for the input sequence.

CNN is widely employed in visual imagery analysis by its convenience to calculate the convolution for high dimensional input. On the other hand, an LSTM network is used considerably to solve time-series problems. LSTM network combines the different weights of the sequence value and calculates the value by some optimization function in a recurrent way. Therefore, the outcome of the network cares more about the dependency and correlation of the sequence. To combine the advantages of a CNN and an LSTM together, we customized a framework which allows a recurrent layer connected to convolutional layers. It could reinforce time-series analysis into CNN and could help capture the dependency between samples. To train such a network, we collected the data from $-10$ dB to 20 dB with a step increment of 2 dB following the setting we described previously. At each SNR, there are roughly 1000 samples $\{s_t\}_{t=1}^{T}$ from each signal type, where each sample includes the I/Q values of 512 data points $x$.

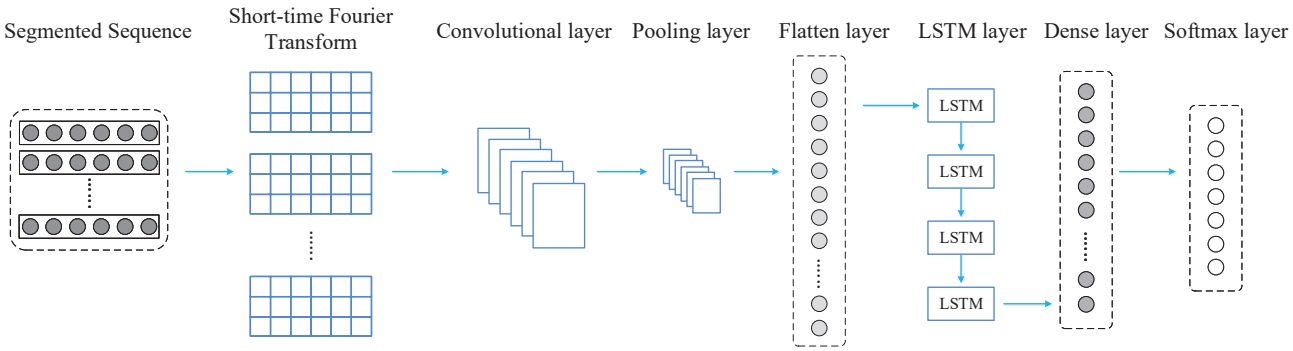



Fig. 2. Overview of the proposed FDA-CNN-LSTM classifier.

### A. Convolutional Neural Network

A CNN typically consists of fully-connected layers, polling layers, convolutional layers, and a softmax layer. It can be trained by the input samples $s_t$ and the corresponding category $y_t$ from $M$ labels. Then, the labeled dataset can be defined as: $D = \{(s_1, y_1), (s_2, y_2), ..., (s_T, y_T)\}$. Let $y_t$ be the $t$th sample's category with $y_t \in \{L\} = \{l_1, l_2, ..., l_M\}$, where $l_r$ stands for the label $r$. Under such definition, we can train and get the hypothesis $\mathcal{H}(s_t) = \overline{y}_t$, which predicts the category $\overline{y}_t$ of $s_t$ that matches the actual one $y_t$.

Samples' feature can be obtained by the convolutional layers during the training. For sample $s$ in two dimensional matrix, convolutional layer $\mathcal{V}$ is extracted by the convolutional kernel:

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

where $W$ 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 scaled exponential linear unit (SELU) as activation functions 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; $\alpha$ and $\lambda$ are the constant value in the SELU setting.

After the feature captured in the convolutional layer, it's connected to the max pooling layer. A max pooling layer helps reduce computing costs by decreasing the dimensionality of the data; besides, it helps prevent the over-fitting problem by providing an abstracted form of the representation. The results of the max polling is flattened and then passed to the LSTM layer to allow the recurrent. The detail of the LSTM layers will be discussed in the followed section. Then the output of the recurrent layer gets through a dense layer. In dense layers, features are mapped until it satisfies the requirement of the output shape. Therefore, the last dense layer in our neural network has seven neurons to pass each class's likelihood to the softmax layer. The probability of each label is normalized to provide category prediction in the softmax layer:

$$P = \{p_{l_1}, p_{l_2}, ..., p_{l_M}\} = \frac{e^{\widehat{y}_m}}{\sum_{r=1}^{M} e^{\widehat{y}_r}} \quad (3)$$

where $\widehat{y}$ is the output of dense layer, and $p_{l_m}$ is the probability of the input to be classified into label $m$.

Our neural network can make a prediction of the given input data $x$ by the hypothesis $\mathcal{H}$ under parameter setting $\theta$: $\overline{y} = \mathcal{H}(\theta, x)$. To measure the difference between an estimated label $\overline{y}$ and the real label $y$, the cross-entropy is introduced to allow the neural network self-involving by this error. In our case, the cross-entropy function is used as the loss function during the training:

$$\mathcal{L}(\theta) = -\sum_{r} \mathcal{B}_r \log(\overline{y}_r) \quad (4)$$

where $\{\mathcal{B}_r\}_{r=1}^{m}$ is a binary variable which will be 1 if the label $r$ is correct among $m$ categories and $\overline{y}_r$ is the corresponding probability of the correction. The CNN minimizes the loss function during the training by calculating the gradient of $\theta$ at each step $j$. Then, $\theta$ gets updated with the corresponding learning rate $\eta$ as follows:

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

### B. Recurrent Neural Network

A typical LSTM network is consists of multiple LSTM cells that determine the parameters of the hidden layers. At each time step $j$, $x_j$ is system input and $\delta_j$ is the output of the LSTM cell. The cell output at the previous time step, $\delta_{j-1}$, is combined with the current system input $x_j$ to form the input for the current cell. The state of each cell is $C_j$, which records the system memory. $C_j$ is updated at each time step. To control the information flow through the cell, several gates are applied, including an input gate ($i_j$), output gate ($o_j$), and forget gate ($f_j$). Each gate generates an output between 0 and 1, where the value of the output is calculated by a sigmoid ($\sigma$) 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:

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

where $W_i$, $W_o$, and $W_f$ are the weights of 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 an output between $-1$ and 1:

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

After the update, $\widetilde{C}_j$ 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 previous cell state, which is processed by the forget gate to determine how past data is to be utilized. With the two components, the cell state at time $j$ is updated as:

$$C_j = f_j C_{j-1} + i_j \widetilde{C}_j \tag{8}$$

The output of the cell $h_j$, 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:

$$h_j = o_j \tanh(C_j) \tag{9}$$

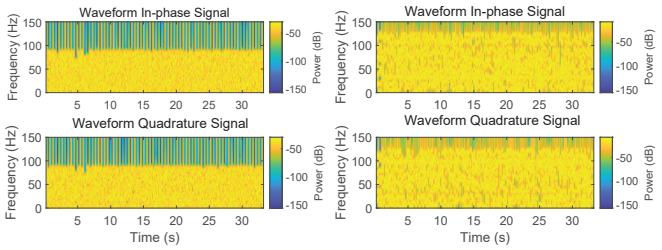
### C. Frequency-domain Analysis

Although the signal shows substantial similarity in time-domain, the spectrogram may still be distinguishable after we apply a Fourier transform, as shown in Fig. 3. The frequency-domain analysis is commonly employed in image processing as well as wireless signal transforming. Such investigation can compensate for the overlook of hidden information in neural networks when the input is time-series I/Q sequences. Besides, frequency-domain signal classification emphasizes the spectrum characteristic on the periodic pattern, which is hard to fully captured by time-series analysis. On the other hand, one convenience of the neural network is its flexibility in input's shape, which makes it possible for us to include the spectrogram information into the network by increasing the input dimension. After such a transform, the frequency analysis consequence can be fed with the I/Q sample into the neural network.

The frequency-domain analysis is conducted in the base-band signal. In particular, a short-time Fourier Transform (STFT) is applied to the I/Q sequences:

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

where $x(k)$ is the time-series 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




(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.

complex function representing the phase and magnitude of the signal over time and frequency. The Kaisar-Bessel window is described as follows to extract the time-frequency features:

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

where $I_0$ is the zeroth-order Bessel function of the first kind. $O$ is the window length, and $\beta$ is the shape factor, which can be determined by side-lobe attenuation $\alpha$. 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 sequence signal into several equal length training samples. The original network is extended to a higher resolution one, including frequency information as other dimension features. Unlike figure classification, time-series I/Q samples, as well as their time-series features, still play an essential role as part of the input. The amplitude fluctuation, phase change, and signal dependency pattern are hidden in the I/Q sequences, which can not be replaced by the spectrogram.

## V. Algorithm Performance Analysis

### A. Benchmark Algorithm

We compare our algorithm with some traditional machine learning (ML) algorithms and independent CNN and LSTM network. As for ML algorithms, we investigate the support vector machine (SVM) and random forests (RF) to verify that ML methods are viable to solve such a signal classification problem. The traditional SVM is successfully applied in class distinction by determining the boundary while maximizing the margin between classes. As our signal classification is not linearly separable, we employ a soft margin in the SVM model to further improve its accuracy. By introducing slack variables $\xi$, the multiplications of the soft margin and $\xi$ can be added to the objective function. After using Lagrangian relaxation, we transform the above constrained problem to an unconstrained one. Furthermore, the objective function of the resulting unconstrained problem becomes a quadratic function. The optimal weight matrix can be obtained during the training by solving each part's derivative in the objective function

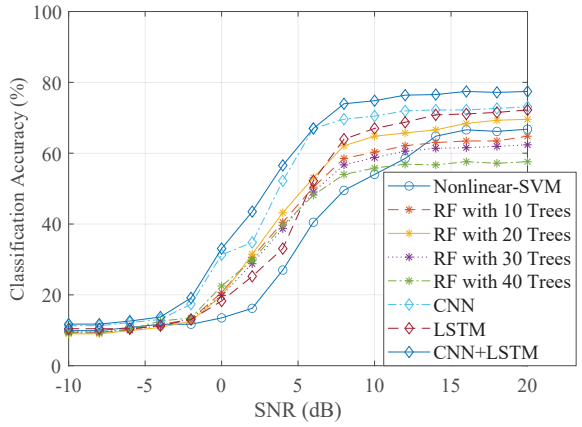



Fig. 4. Classification performance comparison under AWGN.

at every epoch. After such a procedure, the trained SVM is then used for testing the classification accuracy. Meanwhile, RF methods are also investigated in our signal classification problem. The decision tree in RF provides a prediction strategy that traces through all sub-decisions that lead to a particular solution so that the tree's number plays a vital role in RF methods. We also evaluate such performance by setting the RF under different tree numbers.

We compare the proposed methods with these ML methods, and the results show the merits of our approach. Due to the proposed approach includes both CNN and RNN architectures, the performance of independent CNN and RNN is also considered as comparison candidates. Basically, the independent CNN model architecture is described as the same in IV-A but without any LSTM layer. While the independent RNN is the LSTM network defined in IV-B, which passes the output to the dense layers to generate the predicted label for input data.

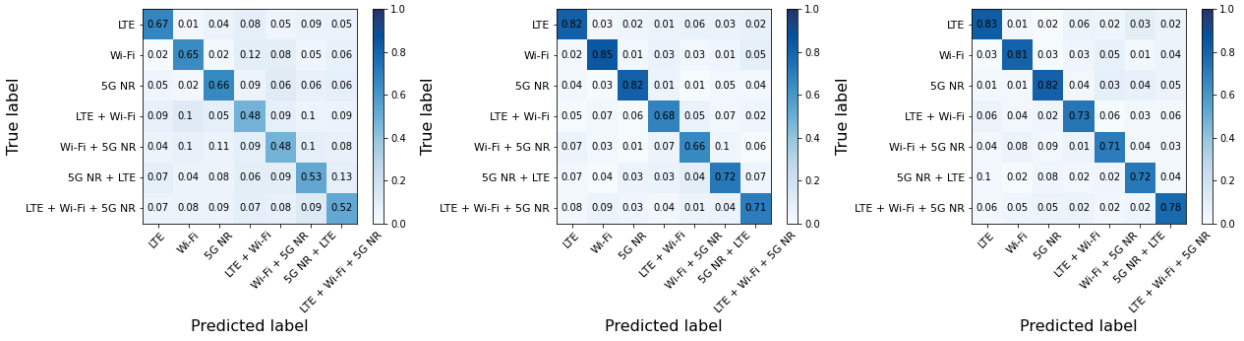
### *B. Classification based on I/Q Samples*

In the simulation, we assume the signal is transmitted under additive white Gaussian noise (AWGN) channel. It is the condition under which analytic feature extractors should generally perform well and matches the analytic model assumptions. Before introducing STFT, we first consider the input as I/Q pairs without frequency analysis. The data generation details are described in Section III. During the testing, the SNR increases 2 dB at each step, and all the mentioned ML/DL algorithm is compared under the same environment setting. As we can see in Fig. 4, the accuracy of nonlinear-SVM is always lower when SNR is less than 10 dB. The achievable accuracy rate is less than 70% for such an algorithm. The random forest can improve the accuracy prominently from 0 to 10 dB intervals. The performance improved when the number of trees increases from 10 to 20, but we can't guarantee further improvement when we keep expanding the tree number. Accuracy can be optimized when the tree number is set to around 20. LSTM is overwhelmed when SNR is less than 6 dB; however, it's accuracy increases faster when SNR is greater. It indicates that LSTM can have better performance under better channel conditions. CNN can achieve similar accuracy when SNR is higher than 14 dB in our case. Besides, CNN performs even better under the lower SNR range compared with other algorithms. The proposed architecture, which includes the LSTM layer into CNN, can further improve the neural network's performance. It achieves higher accuracy than CNN and LSTM; meanwhile, it behaves better in almost all SNR conditions. From our observation, the algorithms obtain the highest accuracy when SNR is greater than 16 dB.

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. It's because the influence the noise is almost neglectable 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,




（a）4 dB　　（b）8 dB　　（c）12 dB

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

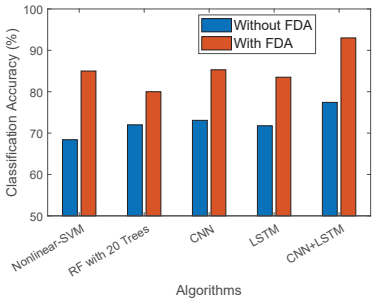
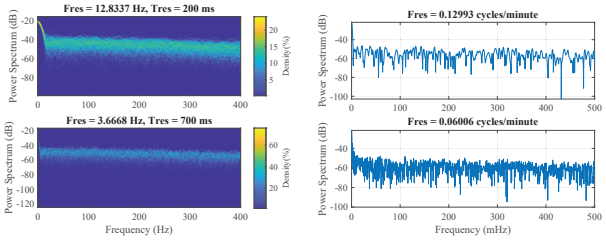



Fig. 6. Impact of FDA on different algorithms.




(a) Impact of resolution　　(b) Impact of leakage

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

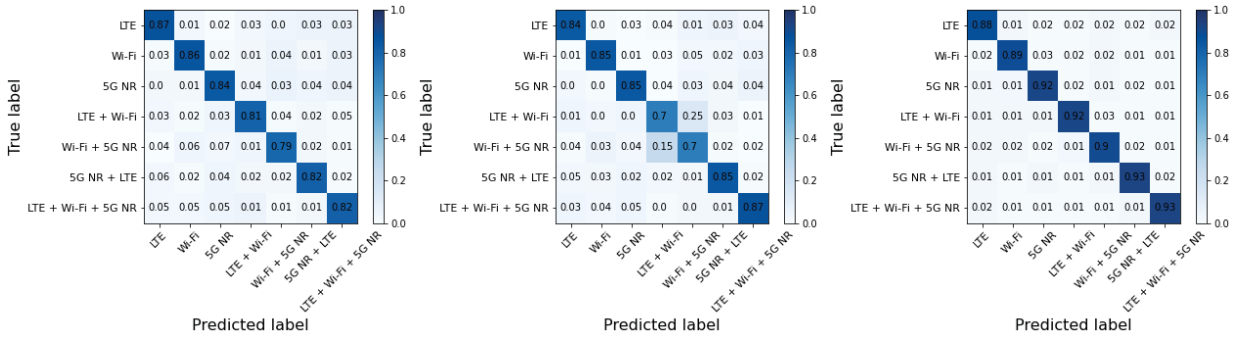
spectrogram information can not guarantee to improve the performance of all of them. Thus, the effect is averaged among trees, which weakens the advance. For CNN, LSTM, and their combination, the accuracy is enhanced by approximately 15%. The proposed algorithm can achieve 93% accuracy after including 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 signal from received LTE with 10 sub-carriers and use it as an example. 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 $\mathrm{T_{res}}$ increased from 200 ms to 700 ms, both the power density and strength decrease in all frequencies. $\mathrm{T_{res}}$ 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 distribution in the frequency space. In our case, the signal power becomes vaguer with lower $\mathrm{T_{res}}$. However, such fuzzy figures enlarge the spectrogram difference between different types of signals. As a result, the higher $\mathrm{T_{res}}$ can not guarantee better classifier performance. During the test, we find when the $\mathrm{T_{res}}$ is around 600 ms, the proposed classifier can achieve the best accuracy. 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 the new frequency components. These components are directly affected by the spectral windowing function; thus, they are considered as results of spectral leakage. To control the spectral leakage, we introduce by the leakage coefficient, which is a real numeric scalar between 0 and 1. It restraints the Kaiser window sidelobe attenuation which is also relative to the mainlobe width. When adjusting such coefficients, the resolution frequency $\mathrm{F_{res}}$ changes correspondingly. When leakage is 1, $\mathrm{F_{res}}$ is 0.06006 cycles/minute, while, when leakage is 0.65, $\mathrm{F_{res}}$ 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 a monotonous with 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.

### *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

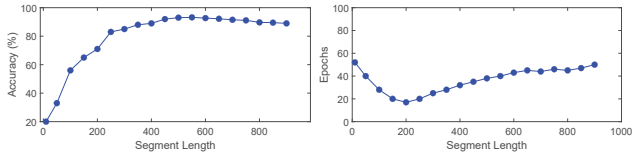



（a）CNN （b）LSTM （c）CNN-LSTM

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

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% |




(a) Accuracy vs. segment length (b) Epochs vs. segment length

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

capture the dependency pattern in the data. In contrast, an LSTM network captures such dependency by storing the memory in the hidden layers, but it produces a regression instead of a probabilistic outcome. After we introduce such layers into the CNN, the performance improves as plotted in the confusion matrix. The overall accuracy improves from 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. The accuracy

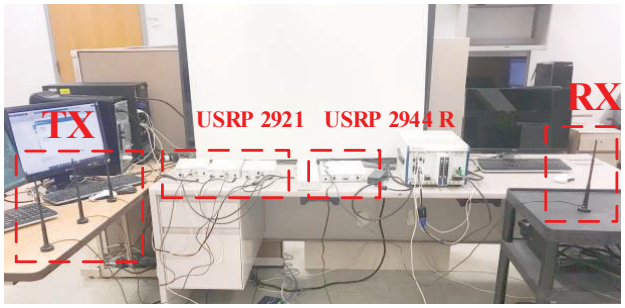



Fig. 10. Experiment setup used for performance evaluation.

is only around 20% when the segment period is short, and it increases fast with input duration until 400. The growth then slows down after that. It's because more features 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's 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 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 table 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-2921 and one NI USRP-

2944R. The 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 5 GHz frequency. 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 WiFi-only, LTE-only, 5G-only, WiFi-LTE, WiFi-5G, LTE-5G, and WiFI-LTE-5G. In all the experiments, signals transmit 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 WiFi waveform is transmitted by generating the WiFi waveform using baseband samples of 802.11 ac (VHT) with BPSK modulation 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 IQ 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. For the environment setting is kept the same during the whole experiment, we assume the noise power is at the same level when we control channel 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 may 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's due to the FDA expand the I/Q samples from the time-domain to the frequency-domain, which 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 non-LSTM layer 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 co-existing signal types by I/Q samples without having to decode them. With segmented sample sequences, CNN is combined with RNN and then trained. The classification results show competitive accuracies by neural networks when the received signal is under the mixture. We then apply STFT on I/Q sequences to further improve the classification accuracy. Neural networks show considerable improvement after including the spectrogram. Moreover, to verify that the proposed detection framework is viable in a real environment, an OTA experiment is then conducted with USRP sets. The proposed deep neural architecture can achieve accurate classification in both the simulation and the experiment.

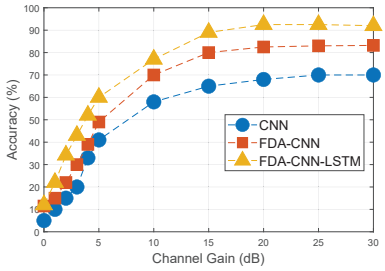



Fig. 11. Average accuracy vs. channel gain.

## 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.