CyPA: A Cyclic Prefix Assisted DNN for Protocol Classification in Shared Spectrum

Wenhan Zhang, Marwan Krunz, and Md Rabiul Hossain

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

***Abstract*—To monitor RF activity and coordinate access to a channel that is shared by heterogeneous wireless systems, network administrators and/or users must be able to identify observed transmissions rapidly and accurately. Recent research shows that deep neural networks (DNNs) can identify the underlying waveform of an RF signal based on the in-phase/quadrature (I/Q) samples without decoding them. Such DNNs take as input a fixed-size window of I/Q samples. To utilize the temporal features at various scales and improve the classification accuracy, we propose a two-stage DNN classification structure. In the first stage, DNN is designed to detect and classify long-term periodic features, such as the cyclic prefix (CP). The output of this classifier is then used as a latent variable for a second-stage protocol (technology) classifier. To evaluate this model, we consider spectrum sharing between Wi-Fi, LTE License Assisted Access (LAA), and 5G NR-unlicensed(NR-U) over the unlicensed 5GHz bands. Compared to the ResNet-18-1D, the proposed two-stage approach improves the classification accuracy from 71% to 90% while reducing the trainable parameters from 3.8 to 1.8 million. As a result, our compact design is more accurate and energy efficient than computational-intensive DNNs for mobile devices.**

***Index Terms*—Deep learning, signal classification, waveform coexistence, spectrum sharing**

I. Introduction

The demand for wireless capacity continues to outgrow spectrum availability, especially at low and mid bands (e.g., sub-7 GHz). To efficiently utilize the allocated spectrum, various spectrum-sharing architectures have been proposed [1]. For example, in the Citizens Broadband Radio Service (CBRS), a three-tier access system is employed, which enables commercial users to share spectrum with incumbent federal and non-federal users [2]. A dynamic frequency selection approach was adopted for the Unlicensed National Information Infrastructure (UNII) bands, permitting LTE License Assisted Access (LAA) and 5G NR-unlicensed (NR-U) cellular technologies to share these bands with Wi-Fi devices [3]. Spectrum sharing and coexistence inevitably introduces interference among users. Therefore, it is critical for network coordinators to rapidly classify observed signals for the purpose of reducing interference and assigning fair channel access.

Identifying the waveforms of heterogeneous protocols is difficult unless the given contending device is equipped with multiple radios. Deep neural networks (DNNs) are designed and proven to complete multi-class detection accurately. In contrast to traditional waveform-based sensing methods, DNN classifiers do not require prior knowledge about the protocol or devices and are efficient for signal detection in shared bands. In our paper, we investigate the DNN for heterogeneous wireless technology[1] classification over a shared spectrum, focusing on Wi-Fi, LTE-LAA, and 5G NR-U in the unlicensed 5 GHz bands as an example.

Several DNNs have been recently proposed to classify RF signals based on received in-phase/quadrature (I/Q) samples [4]–[8]. These approaches take a fixed-size window to sample the I/Q stream and use these samples to train the DNN. Generally, a shorter window captures short-term changes in the sequence and is able to detect the signal faster. However, the cyclic features often occur at longer time scales than a typical window of I/Q samples fed into a classical signal classifier. In addition, a longer window includes more information embedded in the waveform, such as periodicity of the cyclic prefix (CP) in OFDM waveforms, the regularly inserted pilot messages in the time-frequency resource map of the OFDMA schedule, and the alternating pattern of the time-division duplexing (TDD) cycle. In our work, we propose a two-stage classification structure to approximate the cyclic feature on a large scale and then use them for wireless protocol classification with a short sampling window size. In other words, we aim to capture additional temporal features in different scales for an accurate protocol classifier.

In a 5G NR waveform, the CP is inserted after each OFDM symbol. The symbol duration (including the CP) ranges from 4.47 μsec to 71.4 μsec, depending on the numerology (equivalently, the subcarrier spacing). At a sampling rate of 100 M samples/sec, these symbol durations correspond to 447 to 7140 samples. Even if one were to reduce the sampling rate by an order of magnitude (i.e., 10 M samples/sec), thousands of samples per window are still needed to capture multiple CPs for use in one classification instance. Hence, the window sizes of 128 or 512 I/Q pairs, as utilized in [4], [5], are inadequate for capturing temporal correlations associated with CP. In this paper, we first investigate the feasibility of CP duration approximation with a sufficient long sampling window. By incorporating additional temporal features from CP estimation with raw I/Q samples, we show that the protocol classification accuracy effectively gets enhanced.

[1]We use wireless protocol and wireless technology interchangeably in our paper.

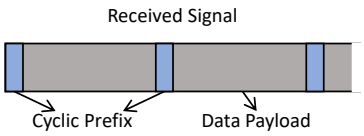


Fig. 1. Example of OFDM cyclic prefix.

Even though there are other advanced DNN structures for accurate classification, these structures usually require more computational resources. In [9], the author proposes to use the number of trainable parameters to compare the complexity of models. The more parameters indicate more energy consumption and more memories required. In wireless mobile networks, such complexity and consumption are practical concerns. Our proposed structure shows that a concise multi-layer perceptron (MLP) with a few trainable parameters can achieve high accuracy, making DNN classifiers feasible for deployment in wireless networks.

II. Cyclic Prefix for Wireless Technologies

CP acts as a buffer region or guard interval to protect the OFDM signals from inter-symbol interference, as shown in Figure 1. It repeats the end of the symbol so the linear convolution of a frequency-selective multi-path channel can be modeled as circular convolution, which in turn may transform to the frequency domain via a discrete Fourier transform. LTE waveforms have a subcarrier spacing (SCS) of 15 kHz with two optional CP durations corresponding to normal and extended CPs. The normal CP is intended to support propagation conditions with a delay spread up to 4.7 μs, while the extended CP support up to 16.7 μs. In general, the normal CP is more commonly found. There are 10 subframes in a 10-msec LTE frame, and each subframe contains 6 or 7 OFDM symbols, depending on whether a normal or extended CP is used. In the case of a normal CP, the first symbol has a duration of 5.2 μs while the remaining 6 symbols have durations of 4.69 μs each. For simplicity, we treat both as one CP type. The CP parameters for LTE are summarized in Table I.

TABLE I
CP and symbol durations for various types of LTE signals

LTE	
Subcarrier Spacing	15 KHz
OFDM Symbols/Subframe	7/6 (normal/extended CP)
CP Length (Normal)	5.2 μs (first symbol)/ 4.69 μs (six following symbols)
CP Length (Extended)	16.67 μs
Symbol Duration (without CP)	66.67 μs

3GPP standards define several 5G NR waveforms with different CP and symbol durations, depending on the numerology (or SCS), as shown in Table II. Some of these durations are meant for Frequency Range 2 (FR2), which is mmWave spectrum. Given that we focus on spectrum sharing with LTE and Wi-Fi, we only consider a subset of the 5G NR CPs in FR1 ("sub-6 GHz bands"). Note when SCS = 60 kHz, the CP duration have two options depending on the normal or extended settings, similar to the LTE. Other than that, one SCS has only one corresponding CP duration setting.

TABLE II
CP and symbol duration for various types of 5G NR signals (i is the index of an OFDM symbol in a frame)

5G NR			
SCS	Duration	CP for Long Symbols	CP for Other Symbols
15 KHz	66.67 μs	5.2 μs (for i = 0 or 7)	4.69 μs
30 KHz	33.33 μs	2.86 μs (for i = 0 or 14)	2.34 μs
60 KHz	16.67 μs	1.69 μs (NCP), 4.17 μs (ECP), for i = 0 or 28	1.17 μs (NCP), 4.17 μs (ECP)
120 KHz	8.33 μs	1.11 μs (for i = 0 or 56)	0.59 μs
240 KHz	4.17 μs	0.81 μs (for i = 0 or 112)	0.29 μs

Wi-Fi signals also have cyclic features, called guard intervals (GIs). For instance, in Wi-Fi 5, specified by the IEEE 802.11ac standard, long and short GIs can be used, with corresponding durations of 0.8 and 0.4 μs, respectively. In both cases, the OFDM symbol duration without the GI is 3.2 μs. Wi-Fi 6, as specified by the IEEE 802.11ax standard, uses a much longer symbol duration of 12.8 μs and, correspondingly, a smaller SCS of 78.125 kHz. Three different GIs are available, as shown in Table III.

TABLE III
CP and GI durations for various types of Wi-Fi 5/6 signals

802.11ac (Wi-Fi 5)		802.11ax (Wi-Fi 6/6E)	
Symbol Duration	3.2 μs	Symbol Duration	12.8 μs
Subcarrier Spacing	312.5 KHz	Subcarrier Spacing	78.125 KHz
Long GI	0.8 μs	Normal GI	0.8 μs
Short GI	0.4 μs	Double GI	1.6 μs
		Quadruple GI	3.2 µs

III. Wireless Technology Classifier Design with CP as Latent Vector

Our wireless technology classifier employs a two-stage prediction approach. The protocol prediction hinges on two consecutive classifiers functioning at distinct time scales, as depicted in Figure 2. In the initial stage, the CP/symbol duration classifier[2] utilizes a lengthy window W_{cp}, illustrated in black, to sample the I/Q stream. Subsequently, the protocol classifier in the second stage utilizes a shorter window W^*, indicated in red, to predict the protocol type of the received signal. For regular classifiers in [4]–[8], the window size is equivalent to W^*. In contrast, our two-stage approach approximates the CP types from W_{cp} and generates the prediction as the latent variable for protocol classification. Such latent vector is padded, reshaped, and appended to W^*, forming $W^* + N$ input to the second (protocol) classifier, as shown in green, where N depends on the number of CP types. In our work, we consider that the sampling windows are non-overlapping

[2]For simplicity, we abbreviate the CP/symbol duration classifier as the CP classifier in our paper.

for both the CP and protocol classifiers. Because V the same latent vector will be used for multiple insta second classifier. For simplicity, we set W_{cp} to be multiple of W^*.

A. Training and Testing Process

We split the data into non-overlapping two sets, ' of data is for training and the rest 20% is for test are two stages in the training process. In the first I/Q pairs are sampled into windows of W_{cp}. Thes are used to train the CP classifier with CP labels. W the trained classifier to generate the CP predictic training set. These predictions are the latent vect next stage classifier. In the second stage, we train tl classifier using the I/Q pairs and latent vectors within W^*+N . Correspondingly, the labels become the protocol types. During the testing phase, we also evaluate our model in two stages. Firstly, the trained CP classifier uses the I/Q pairs within W_{cp} to generate latent vectors for the testing set. After that, the I/Q pairs associated with latent vectors $(W^* + N)$ are used as input for protocol classification. Finally, we compare the output of the second-stage classifier with the true protocol label to evaluate our model's accuracy.

B. Cyclic Prefix Assisted (CyPA) Latent Vector

A DNN can be represented by the mapping $z = g(x;\theta)$, where x is a window of I/Q samples and θ is the set of learnable DNN parameters. The input x is in $\mathbb{R}^{2\times W}$, where W is the window size (in consecutive samples) and the first (second) row represents the sequence of I (Q) values, respectively. The output z is in $\mathbb{R}^K$, and K is the number of classes. The input matrix x is passed through the DNN and is represented by a feature vector that is the result of a projection and nonlinear (activation) function, $\phi(\cdot)$. For our CP classifier, the activation function for the last layer (i.e., output layer) is the softmax function σ: $\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$.

Where $i \in \{1, 2, ..., K\} \doteq \mathcal{K}$. After activation function, the classifier output results $\{\sigma(z)_1, \sigma(z)_2, ..., \sigma(z)_K\}$. These results have been normalized by softmax function so we treat them as the *confidence* for each CP class. To generate the prediction, the classifier assigns a label $f(x;\theta) = \arg\max_k(\sigma(z)_k)$ to the received input, where $k \in \mathcal{K}$. To make use of the hard outcome and keep it the same length of as the confidence vector, we apply *one-hot encoding* [10]. Such a output vector consists of all '0's except for one element that has a value '1'. The location of bit '1' within the output vector indicates the most likely CP type. These two types of output of the CP classifier are used as the latent vector that consists of either probabilities (soft outcome) or binary values (hard outcome), indicating the CP duration belongs to a given type.

C. Hyperparameters for Classifiers

We use MLPs for both the CP and protocol classifiers. The CP classifier includes four layers with sizes 512, 256, 64, and 7. In contrast, the protocol classifier has the same structure except three neurons in the output layer. This is because

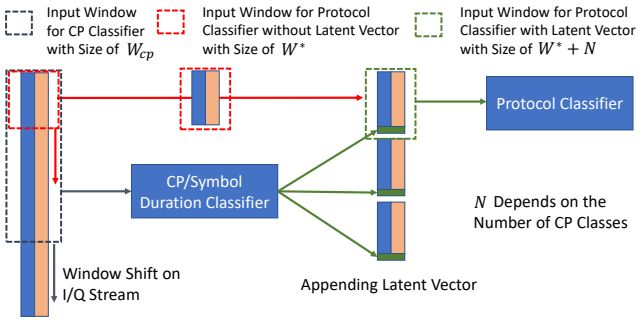


Fig. 2. Overall structure of the two-stage protocol classifier.

TABLE IV
HYPERPARAMETERS FOR PROPOSED TWO-STAGE CLASSIFICATION STRUCTURE

Activation Function	ReLU
Loss	Categorical Cross-entropy
Overfitting Prevention	Early Stopping (Patience = 10)
Batch Size	512
Optimizer	Adam
Max Training Epochs	100

we consider three possible protocols associated with seven types of CPs. Other hyperparameters for classifier training are summarized in Table IV. We've studied the performance with/without dropout and normalization in these classifiers. We found both to have negligible impact on the classification accuracy. For brevity, we present the results when normalization and dropout are not performed.

IV. Data Generation

Data generation for the three technologies is conducted using *Matlab* LTE, 5G, and WLAN communication toolboxes. We produce a dataset for 7 different CP "types": two related to LTE signals (Normal and Extended CPs), two to 5G NR signals (15 kHz and 30 kHz SCS), two to 802.11ax signals (Normal and Double GIs), and one to 802.11ac signals (long GI). For each type of CP, we generate 15 AWGN channel realizations, where 12 are used for training and 3 for testing. Each realization includes about $300,000$ I/Q pairs. Among the three types of signals, LTE with Extended CP has the longest OFDM+CP duration (83.34 μs). At a sampling rate of 30.72 MS/sec, this duration corresponds to about 2560 I/Q samples. Thus, we set $W_{cp} = 2560$, which ensures that any window sample will cover at least one OFDM duration plus its CP or GI for all protocols. Considering non-overlapping windows, each realization results in 307200/2560=120 windows, ending with 1,440 sample windows for training and 360 sample windows for testing for each CP type. In total, we have 10,080 and 2,520 windows of samples for training and testing, respectively, across all 7 CP types. These CP types correspond to the SCS value by definition.

The channel realizations are produced at SNR = 15 dB, where the SNR value is controlled by channel noise. The I/Q

samples (with noise included) are normalized by their mean energy over the whole realization. Since our final objective is detecting the wireless protocol types, we restrain the same bandwidth of 20 MHz and the same modulation scheme of 64-QAM for all three protocols to eliminate their impact. Some other parameter values are selected to ensure that the realizations of different technologies are equal in length, as measured in the number of I/Q samples/realization. While this is not a requirement, it ensures the results for different technologies have the same statistical confidence.

V. Performance Evaluation

Our proposed CyPA structure includes two stages. In the first stage, we use a CP/symbol duration classifier, as described in Section III, to classify CP and generate latent variables. We depict the classification results of the 7 CP types in the confusion matrix shown in Figure 3. The CP classifier can accurately detect the GI duration of a Wi-Fi signal with 100% accuracy. This can be explained by the fact that the GI durations for Wi-Fi (0.8 or 1.6 μs) are quite distinct and are much shorter than the CP durations used for LTE and 5G NR. The classifier can also successfully distinguish between Wi-Fi 6 and Wi-Fi 5, even if both have the same GI duration of 0.8 μs. In contrast, differentiating between the CP durations of LTE and 5G NR is not as easy. In fact, the average classification accuracy for the four CP types of LTE and 5G NR is just 51.5%. The overall testing accuracy for the CP classifier is 72.34%.

We also considered other state-of-art classifiers, including LSTM, bidirectional RNN, VT-CNN2, and ResNet [4], [5], [11]–[13]. However, these classifiers have limited accuracy improvement but a much more significant increase in model size. We compare these models with CyPA in Section V-C. The inaccurate prediction also indicates that the regular DNN structure is challenging to fully utilize the CP information. As a result, we propose a two-stage design for protocol classification, where we develop the second-stage classifier using the latent variable generated from the CP classifier.

A. Precision, Recall, and F1 Score

We consider using precision, recall, and F1 score to evaluate the second-stage classifier. For each protocol, the result becomes binary. There are True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN), four types of classification depending on the actual and predicted labels. Precision is defined as TP/(TP+FP) to illustrate the accuracy among all positive predictions. In contrast, recall equals TP/(TP+FN), showing accuracy among all actual positive inputs. A higher precision and recall represent a better classifier. F1 score is generated from precision and recall as $2\times$precision$\times$recall/(precision + recall). This is approximately the harmonic mean of precision and recall. A high F1 score indicates the classifier is less bias in FN and FP.

1) Evaluating the Performance of a Single-Stage Protocol Classifier (without CP): Before we study the combined two-classifier design, as a reference point we first study the performance of a protocol classifier that does not rely on the CP duration classifier, i.e., it only uses a single classification stage. In this case, the protocol classifier has three possible outcomes (labels): LTE, 5G, and Wi-Fi. The training and testing data are the same as the ones used to study the 7-label CP duration classifier, i.e., realizations of different variants of the three protocols. We set $W^* = 2560$. An MLP classifier is used, and the corresponding results are presented as 'Without CP' in Table V. Although the Wi-Fi signal can be accurately identified, the classifier is not as accurate in differentiating between LTE and 5G NR signals. The F1 score is only around 60% for these two labels, indicating the classifier has high FN and FP predictions. The overall testing accuracy for the 3-label protocol classifier is 77.17%.

2) Evaluating the Performance of the Two-Classifier Structure with Hard CP: We evaluate the overall performance of the two-classifier design. We first train the CP classifier, as shown in Figure 3. After that, we train the second (protocol) classifier using the true values of the latent vector (binary). This is because the true CP labels of the input are known during the training phase. These labels provide more accurate information for the protocol classifier training. However, during the testing, we can only rely on the first classifier to predict values of the latent vector (also binary). The classification accuracy is summarized as 'With Hard CP' in Table V. Compared to a

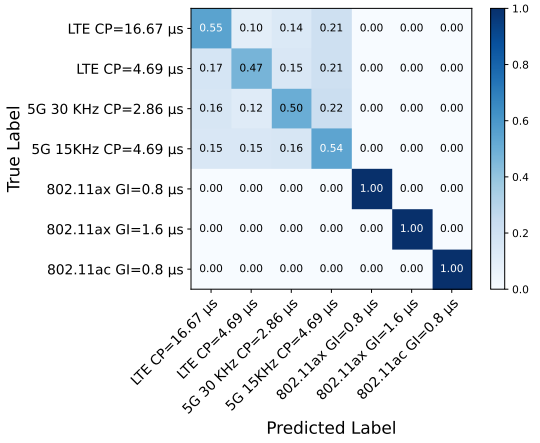
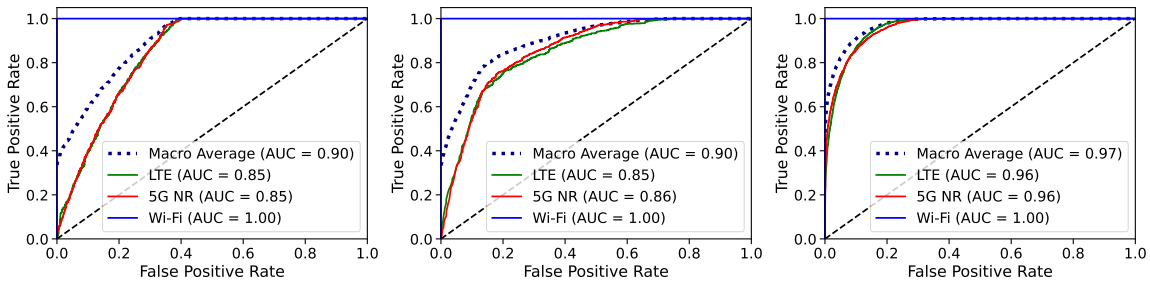


Fig. 3. Confusion matrix for the CP classifier, considering 7 CP types (SNR = 15 dB).

TABLE V
Precision, Recall and F1 Score for Classifiers with Different CP Types as Latent Variables

Metrics	Protocol	Without CP (%)	With Hard CP (%)	With Soft CP(%)
Precision	LTE	63.52	64.15	79.61
	5G	57.47	70.82	80.39
	Wi-Fi	100	100	100
Recall	LTE	59.86	68.73	80.23
	5G	61.17	66.37	79.76
	Wi-Fi	100	100	100
F1 Score	LTE	61.65	66.36	79.92
	5G	59.26	68.53	80.08
	Wi-Fi	100	100	100



(a) (b) (c)

Fig. 4. ROC curve for wireless technology (protocol) classifiers with different CP types as appended latent variables, trained and tested under data with multiple CP types per protocol. (a) Without CP, (b) with hard CP, and (c) with soft CP.

protocol classifier without CP, appending hard CPs improves the precision, recall, and F1 score for 5G signals by 13.35%, 5.2%, and 9.27%, respectively. The precision and F1 score for LTE improves slightly, but the increment in the recall is 8.87%. In addition, the average accuracy over three classes increases from 77.17% to 81.42%.

3) Evaluating the Performance of the Two-Classifier Structure with Soft CP: Although the appended latent vector improves the classification accuracy, the appended vectors in the training and testing phases are different (one is true while the other is predicted). Instead, we consider appending the predicted soft output of the first classifier to both training and testing samples used in the second classifier. This approach is justified by the fact that predictions on CP type are only 72.34% accurate on average, implying that the predicted and true latent vectors can differ significantly, thus impairing the second classifier. Even though the soft-predicted latent vector is not 100% accurate during the training part, it still approximates the distribution of the CP types. Using this approach, we obtained the results as 'With Soft CP' in Table V. The precision, recall, and F1 scores for LTE and 5G are further improved by another 10% than hard CPs. On average, the classification accuracy for the 2-classifier design increases to 88.57%.

The above evaluation for the protocol classifier has a window length of $W^* = 2056$. However, in some cases, predictions can be made based on a smaller window size. Accordingly, we investigate the accuracy of protocol classification under two other values of W^* (with $W_{cp} = 2560$), considering these three CP types, and show the results in Figure 5. When $W^* = 512$, we can see that the classification accuracy improves from 71.75% when only one classifier is used to 80.54% when appending hard CPs. It means the latent variable about the CP generated from the first classifier is valuable for the protocol classifier. Moreover, the accuracy further improves to 89.99% when appending the soft CPs. This indicates appending the confidence about the CP from the first classifier can effectively enhance the accuracy of protocol classification, even if the protocol classifier has a much shorter window size. We further vary the length of such window, and a similar trend is observed when $W^* = 640$.

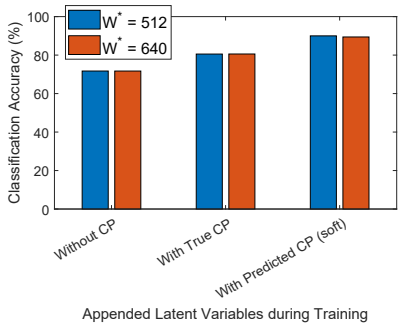


Fig. 5. The impact of true and soft CP information on classification accuracy of the protocol classifier.

B. Receiver Operating Characteristic (ROC) Curve

We compare the ROC for these three CP types to illustrate the diagnostic ability of the classifier as its discrimination threshold is varied, as shown in Figure 4. The ROC curve is plotted by True Positive Rate (TPR) and False Positive Rate (FPR). The higher Area Under the Curve (AUC) means a better model's performance distinguishing between the positive and negative classes. Among these three classes, the ROC curve for Wi-Fi is perfect. This is because the Wi-Fi signal significantly differs in the CP duration from cellular signals. It aligns with the observation in Figure 3, where Wi-Fi signals have a 100% classification accuracy. In addition, we observe that classifiers without CP and with hard CP have a similar average AUC, even if the appended hard CP improves the classification accuracy. By including the soft CP, the average AUC is improved to 0.97, bringing a better classification for both LTE and 5G NR.

C. Trainable Parameters

Our CyPA design can effectively increase classification accuracy. Meanwhile, the model is controlled to be compact with less trainable parameters [9] than the other models. We compared our model with some state-of-art approaches [4], [5], [11]–[13] for three-label protocol classification with the same dataset. Our CyPA design requires $W_{cp}* = 2560$; all the rest classifiers also apply the window size of 2560 for the fair comparison. ResNet [14] allows shortcut in DNN blocks and

TABLE VI
NUMBER OF PARAMETERS AND CORRESPONDING ACCURACY COMPARISON

Method/Model	Parameters	Testing Accuracy (%)
VT-CNN2 [4]	52,474,451	88.04
Bi-LSTM [5]	13,284,483	86.86
LSTM [11]	6,728,835	74.43
ResNet-50-1D [12]	7,245,123	71.85
ResNet-18-1D [13]	3,859,651	71.46
Ours	**1,874,314**	**89.99**

TABLE VII
PROTOCOL CLASSIFICATION ACCURACY COMPARISON UNDER DIFFERENT W_{cp}

W_{cp}	Accuracy without CP	Accuracy with Predicted CP (soft)
2560	77.17%	89.99%
3840	84.30%	94.01%
5120	85.47%	93.69%

facilitate the neural network to become deeper. However, such an operation in the block reduces the size of the intermediate output for the next layer due to the filter function. In our case, the inputs have the size of $2 \times W$, and the shape of 2 does not support the procedure as in the image inputs. Therefore, we replace all *Conv2D* and *Pooling2D* layers to *Conv1D* and *Pooling1D* layers as authors did in [12]. We call the modified model ResNet-1D and train them for protocol classification.

We summarize the parameters and testing accuracy of these models in Table. Models with more parameters tend to have higher accuracy for protocol classification. VT-CNN2 has the most parameters with 88.04% accuracy. While bidirectional LSTM has only one-quarter parameters of VT-CNN2, it has a less accurate prediction. LSTM and ResNet-50-1D reduce the parameters to less than ten million, sacrificing the classification accuracy to less than 80%. Comparing ResNet-18-1D with ResNet-50-1D, the parameter amount gets reduced to almost half, but the accuracy keeps similar. Among these model, our design has the least parameters of 1,874,314 and achieve the highest accuracy of 89.99%.

D. Impact of CP Window Size

We have evaluated the proposed approach with $W_{cp} = 2560$ to guarantee the CP window includes at least one cyclic period. In addition to this window size, we extend our evaluation to $W_{cp} = 3840$ and 5120, with the same neural network structure as before. The results of the protocol classification are summarized in Table VII. We observe that the longer W_{cp} can improve the accuracy of the protocol classifier without CP prediction. Moreover, the proposed CP prediction embedded approach can still increase classification accuracy by around 10% in these CP window sizes. The accuracy improvement in all three window sizes validates the effectiveness of the CyPA design.

VI. DISCUSSIONS

Applying a CP-duration classifier as a first stage, followed by a protocol classifier, leads to significant performance gain in the classification accuracy. In addition, such a structure is more resource efficient since it requires much fewer parameters than DNN models with the similar accuracy performance. Although we implement our approach to classifying LTE, 5G, and Wi-Fi signals, such an idea can be used to classify other wireless technologies with periodic frame/packet duration settings (e.g., WiMAX and Bluetooth). In addition to advantages, there are also limitations in our work. For example, our results are all based on the simulation data and have not considered real channels. We will study and address these limitations in our future work.

REFERENCES

[1] A. M. Voicu, L. Simić, and M. Petrova, "Survey of spectrum sharing for inter-technology coexistence," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1112–1144, 2019.

[2] N. Soltani, V. Chaudhary, D. Roy, and K. Chowdhury, "Finding waldo in the CBRS band: Signal detection and localization in the 3.5 GHz spectrum," in *Proc. of IEEE Global Communications Conference (GLOBECOM)*, 2022, pp. 4570–4575.

[3] W. Zhang, M. Feng, M. Krunz, and A. H. Y. Abyaneh, "Signal detection and classification in shared spectrum: A deep learning approach," in *Proc. of the IEEE Conference on Computer Communications (INFOCOM)*, May 2021, pp. 1–10.

[4] 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, 2018.

[5] W. Zhang and M. Krunz, "Machine learning based protocol classification in unlicensed 5 GHz bands," in *Proc. of the IEEE International Conference on Communications Workshops*, May 2022, pp. 1–6.

[6] S. Hanna, C. Dick, and D. Cabric, "Signal processing-based deep learning for blind symbol decoding and modulation classification," *IEEE Journal on Selected Areas in Communications*, vol. 40, no. 1, pp. 82–96, 2022.

[7] Y. Kumar, M. Sheoran, G. Jajoo, and S. K. Yadav, "Automatic modulation classification based on constellation density using deep learning," *IEEE Communications Letters*, vol. 24, no. 6, pp. 1275–1278, 2020.

[8] A. P. Hermawan, R. R. Ginanjar, D.-S. Kim, and J.-M. Lee, "CNN-based automatic modulation classification for beyond 5G communications," *IEEE Communications Letters*, vol. 24, no. 5, pp. 1038–1041, 2020.

[9] A. Thakur, V. Abrol, P. Sharma, T. Zhu, and D. A. Clifton, "Incremental trainable parameter selection in deep neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2022.

[10] M. K. Dahouda and I. Joe, "A deep-learned embedding technique for categorical features encoding," *IEEE Access*, vol. 9, pp. 114 381–114 391, 2021.

[11] T. Ergen, A. H. Mirza, and S. S. Kozat, "Energy-efficient LSTM networks for online learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 8, pp. 3114–3126, 2020.

[12] T. Jian, B. C. Rendon, E. Ojuba, N. Soltani, Z. Wang, K. Sankhe, A. Gritsenko, J. Dy, K. Chowdhury, and S. Ioannidis, "Deep learning for RF fingerprinting: A massive experimental study," *IEEE Internet of Things Magazine*, vol. 3, no. 1, pp. 50–57, 2020.

[13] F. He, T. Liu, and D. Tao, "Why ResNet works? Residuals generalize," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 12, pp. 5349–5362, 2020.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.