An Efficient Specific Emitter Identification Method Based on Complex-Valued Neural Networks and Network Compression

Yu Wang, *Graduate Student Member, IEEE*, Guan Gui, *Senior Member, IEEE*,
Haris Gacanin, *Fellow, IEEE*, Tomoaki Ohtsuki, *Senior Member, IEEE*,
Octavia A. Dobre, *Fellow, IEEE*, and H. Vincent Poor, *Life Fellow, IEEE*

***Abstract*— Specific emitter identification (SEI) is a promising technology to discriminate the individual emitter and enhance the security of various wireless communication systems. SEI is generally based on radio frequency fingerprinting (RFF) originated from the imperfection of emitter's hardware, which is difficult to forge. SEI is generally modeled as a classification task and deep learning (DL), which exhibits powerful classification capability, has been introduced into SEI for better identification performance. In the recent years, a novel DL model, named as complex-valued neural network (CVNN), has been applied into SEI methods for directly processing complex baseband signal and improving identification performance, but it also brings high model complexity and large model size, which is not conducive to the deployment of SEI, especially in Internet-of-things (IoT) scenarios. Thus, we propose an efficient SEI method based on CVNN and network compression, and the former is for performance improvement, while the latter is to reduce model complexity and size with ensuring satisfactory identification performance. Simulation results demonstrated that our proposed CVNN-based SEI method is superior to the existing DL-based methods in both identification performance and convergence speed, and the identification accuracy of CVNN can reach up to nearly 100% at high signal-to-noise ratios (SNRs). In addition, SlimCVNN just has 10%$\sim$30% model sizes of the basic CVNN, and its computing complexity has different degrees of decline at different SNRs; there is almost no performance gap between SlimCVNN and CVNN. These results demonstrated the feasibility and potential of CVNN and model compression.**

***Index Terms*— Specific emitter identification (SEI), complex-valued neural network (CVNN), sparse structure selection (Triple-S), knowledge distillation (KD).**

I. Introduction

IN WIRELESS networks, the device authentication employs IP or MAC address as the identification mark, which is easy to be counterfeited and tampered, leading to security problems. For the purpose of enhancing communication security, specific emitter identification (SEI) based on radio frequency fingerprinting (RFF) extracted from the signal [1]–[3] has been proposed. RFF for SEI is inherent and unique for every emitter, because it is derived from the imperfection of emitter's hardware, such as the nonlinearity of power amplifier (PA) [4], [5], and RFF is hard to be counterfeited.

SEI has been widely investigated in cognitive radios [6], self-organized networks [7] and Internet of things (IoT) [8], [9] for improving the security of authentication systems, as shown in Fig. 1. Based on the collected signal for identifying emitters, SEI can be categorized into two classes [10]. 1) SEI method based on the transient signal, which generally appears during the emitter turning on or off. 2) SEI method with the steady-state signal, which is easier to be captured than the transient signal. The second class is the main research direction at present. However, there is no difference between two kinds of SEI methods in terms of identification steps, which mainly includes RFF extraction and device identification [11]. Specifically, RFF extraction extracts the manmade features from signals converted by short-time Fourier transform [12], Wigner-Ville distribution [13], *et al*. In addition, device identification is to compare the extracted RFF with RFFs in database and then give the identification results. The common individual

Manuscript received December 8, 2020; revised March 17, 2021 and May 3, 2021; accepted May 16, 2021. Date of publication June 7, 2021; date of current version July 16, 2021. This work was supported in part by the Japan Society for the Promotion of Science (JSPS) KAKENHI under Grant JP19H02142, in part by the U.S. National Science Foundation under Grant CCr-1908308 and Grant CCF-1908308, in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) through its Discovery Program, in part by the Project Funded by the National Science and Technology Major Project of the Ministry of Science and Technology of China under Grant TC190A3WZ-2, in part by the National Natural Science Foundation of China under Grant 61901228, in part by the Summit of the Six Top Talents Program of Jiangsu under Grant XYDXX-010, in part by the Program for High-Level Entrepreneurial and Innovative Team under Grant CZ002SC19001, in part by the Project of the Key Laboratory of Universal Wireless Communications (BUPT) of Ministry of Education of China under Grant KFKT-2020106, and in part by the Open Project of the Shaanxi Key Laboratory of Information Communication Network and Security under Grant ICNS201902. *(Corresponding author: Guan Gui.)*

Yu Wang and Guan Gui are with the College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China (e-mail: 1018010407@njupt.edu.cn; guiguang@njupt.edu.cn).

Haris Gacanin is with the Faculty of Electrical Engineering and Information Technology, RWTH Aachen University, 55-52062 Aachen, Germany (e-mail: harisg@ieee.org).

Tomoaki Ohtsuki is with the Department of Information and Computer Science, Keio University, Yokohama 223-8521, Japan (e-mail: ohtsuki@ics.keio.ac.jp).

Octavia A. Dobre is with the Faculty of Engineering and Applied Science, Memorial University, St. John's, NL A1C 5S7, Canada (e-mail: odobre@mun.ca).

H. Vincent Poor is with the Department of Electrical and Computer Engineering, Princeton University, Princeton, NJ 08544 USA (e-mail: poor@princeton.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JSAC.2021.3087243.

Digital Object Identifier 10.1109/JSAC.2021.3087243

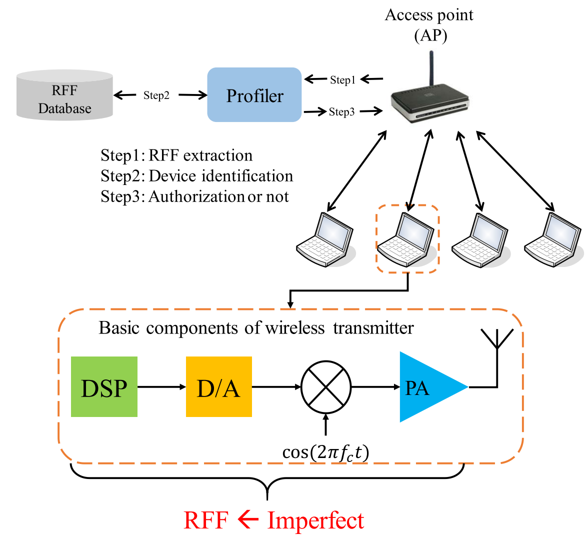


Fig. 1. The typical SEI-aided authentication system, which consists of RFF extraction, device identification and authorization. In addition, the basic components of wireless transmitter are also given, and RFF is originated from the imperfection of these components [4].

identification technologies includes K-nearest neighbor [11] and multiple discriminant classifier [14].

Recently, deep learning (DL) has been introduced into wireless communication technologies for improving performance or reducing complexity [15]–[17]. For instance, Ye *et al.* [18] investigated the power of DL in signal detection, and authors proposed a effective and data-driven detector for orthogonal frequency division multiplex (OFDM) system. Huang *et al.* [19] proposed a novel beamforming prediction network (BPNet) for jointly optimizing power allocation and virtual uplink beamforming design which has been demonstrated that BPNet has fast speed with superior performance. Guo *et al.* [20] proposed a multiple-rate CSI compression and quantization method, which not only enhances reconstruction accuracy by replacing fully-connected layers with convolution layers, but also reduces storage space by sharing some model parameters.

Moreover, DL has been also applied into signal identification [21]–[24] and SEI [25]–[34], and the previously proposed DL-based SEI methods [25], [26] adopted real-valued neural network (RVNN) trained on the two-dimension matrix, consisting of the in-phase and quadrature (IQ) components of complex baseband signal. However, these RVNNs cannot directly process complex baseband signals, and they generally ignored the correlation between IQ, which would result in performance degradation [24]. In the last years, Agadakos *et al.* [32] proposed a deep complex-valued convolution neural network combined with long short term memory (LSTM) for SEI, which achieved outstanding performance, but has a high inference time. Gopalakrishnan *et al.* [33] analyzed the performance of CVNN and RVNN, showing that CVNN has outstanding performance for SEI methods, but it also brings the high computing complexity or large model size.

In this paper, we design an efficient SEI method based on CVCNN and neural network compression technologies for achieving performance improvement and model compression. Unlike recent works [25], [26], [32], [33] with large convolution kernel, small convolution kernel (kernel size is 3) is adopted and pooling operation, following behind every complex-valued convolution layer, is applied for feature dimension reduction. Based on these tricks, we design an efficient basic CVCNN model. To further reduce the model size and computational complexity of the basic CVCNN model for SEI, we introduced a sparse structure selection (Triple-S) algorithm based on sparse regularization [35], [36], quantization mask [37] and proximal gradient method [38]. The proposed method can select an efficient CVCNN structure from the original structure, which is named as SlimCVCNN. Moreover, we consider a knowledge distillation (KD) algorithm to adopt training samples with its corresponding labels and the predicted labels from the basic CVCNN model. This jointly recovers the performance of SlimCVCNN. Simulation results confirmed that the CVCNN-based SEI method is superior to the real-valued convolution neural network (RVCNN)-based SEI method in both identification accuracy and convergence rate, and the proposed SlimCVCNN-based SEI method with KD achieves high identification accuracy, while introducing lower size of the model leading to significant reduction of complexity.

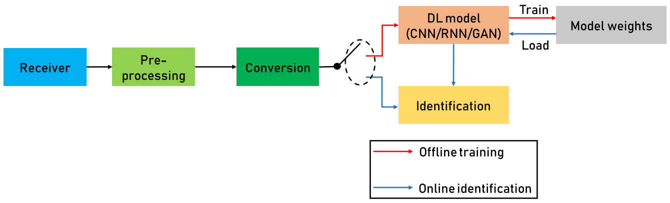


Fig. 2. DL-based SEI methods with offline training and online identification.

II. Related Works

A. DL-Based SEI Methods

Here, we mostly focus on SEI method using the steady-state signal, and it can be categorized into two types: 1) based on artificial features and typical classifiers; 2) based on DL with automatic feature extraction and identification. The latter has been demonstrated to have better performance than the former. Thus, we pay more attention to the DL-based SEI methods.

The structure of DL-based SEI methods is shown in Fig. 2, and the common DL models include convolution neural network (CNN) [25]–[29], recurrent neural network (RNN) [30], generative adversarial network (GAN) [31] and so on. Next, various previously proposed DL-based SEI methods will be introduced.

1) SEI Methods in Time Domain: The kind of SEI methods is based on complex baseband signals in time domain. Merchant *et al.*, [25] firstly proposed a CNN-based SEI method for ZigBee devices using the complex baseband error signal, which has been demonstrated more than 90% identification accuracy at high SNR. Yu *et al.* [26] applied multisampling CNN (MSCNN) to identify 54 CC2530 devices, which achieves higher than 97% identification accuracy under the line-of-slight scenarios with high SNR. In detail, authors adopted multiple downsampling transformations to pre-process the complex baseband signal, and than these multiple-downsampled signals are input into CNN for extracting multiscale features. F. Xie, *et al*., also proposed a one-dimensional signal convolution for extracting features [27]. And it is innovative that they applied discrete wavelet transform-based pooling operation rather than maximum or average pooling operation to reduce the dimensionality of extracted features, which can save most of important features of signals.

2) SEI Methods in Transformation Domain: Different from the above SEI methods, this kind of SEI methods generally needs to transform baseband signals by spectrum analysis, signal decomposition and so on, and then the transformed signals are fed into DL. Ding *et. al.* [28] proposed a novel CNN-base SEI method, where it adopted the compressed bispectrum of the received steady-state signals to train and test the CNN-based SEI, and this method outperformed HHT-based, square integral bispectrum (SIB)-based, and constellation-based SEI methods. In addition, Peng *et al*. [29] proposed a DCTF-based CNN for classifying 54 target ZigBee devices, where the baseband signals are converted into DCTF with the size of 65 × 65, and then built a simple four-layer CNN to identify devices. Simulation results demonstrated that its accuracy exceeds 99% at SNR $\leq$ 30 dB. Moreover, a signal decomposition and LSTM-based SEI method was proposed by He and Wang [30] and it also achieved the advanced performances. In detail, authors adopts three signal decomposition methods of empirical mode decomposition, intrinsic time-scale decomposition, and variational mode decomposition to decompose baseband signals, and then LSTM is to learn the deep features for identifying different specific emitters.

Comparing the above two methods, we can find that although the performance of the latter is slightly higher than the former, especially when SNR is low, signal transformation will introduce additional computing complexity, which is not conducive to the application in Internet-of-things and other fields.

B. The Applications of Complex-Valued Neural Networks and Network Compression

Most of DL-based communication technologies are based on real-valued neural network (RVNN), which cannot directly process complex data, but a novel DL model, named as complex-valued neural network (CVNN), has been proposed, and it has the capability to process complex baseband signal. It contains various layers with complex-valued weights, such as complex-valued convolution layer, complex-valued fully-connected layer, complex-valued batch normalization (CVBN) and complex-valued activation function. In recent years, CVNN has been applied in different communication scenarios, including channel prediction [39], modulation classification [24], noncoherent demodulation [40], mitigation of signal distortion [41] and SEI [32], [33], because of its excellent performance and the capability to process complex-valued data directly. However, the CVNN has higher computing complexity or more model parameters than the RVNN [32], [33], which may be solved by network compression technologies.

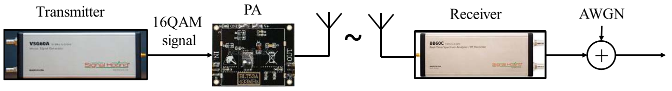


Fig. 3. The steps of data collection.

Network compression is a set of technologies to compress model size and reduce computing complexity, whose typical technologies include neuron pruning, quantization, efficient structure design, knowledge distillation (KD) and so on [42]. Work in [43] firstly introduced neuron pruning for designing a lightweight modulation classification method, where authors applied average percentage of zero activation as the metric to distinguishing effective neurons from redundant neurons. The method requires iterative training. A lightweight semantic communication system is designed by weight-sum-based neuron pruning and quantization in [44]. Paper [45] applied efficient structure design of lightweight CSI feedback method, inspired by Shufflenet [46]. The above papers have shown the feasibility and potential of network compression in communication systems or communication technologies. However, there are few papers about network compression for SEI methods and network compression. Thus, in this paper, we are committed to develop CVNN-based SEI method employing a network compression method.

III. Data Collection and SEI Design

A. Data Collection and Pre-Processing

In this paper, PA is adopted as the research object of SEI, because it is the last component of the transmitter and it is the most difficult to modify by software or even baseband control [4], and RFF generated by PA has higher credibility. The detailed steps of data collection about different PAs are given as follows.

The sixteen-quadrature amplitude modulation (16QAM) signal is produced by VSG60A, and its frequency is 433 MHz. Then, the signal is fed into identical PAs, whose model is BLT53A with 0 dBm gain, and we can get the transmitted signal. Next, the transmitted signal passes through the real communication environment and is collected by BB60C [10], i.e., the received complex baseband signal. Finally, we add additive white Gaussian noise (AWGN) to this received signal and normalize its energy. The processed signal is denoted as $\mathbf{S}$, whose SNR is in the range of {10, 15, 20, 25, 30, 35, 40} dB.

B. DL-Based SEI Design

DL-based SEI is generally designed by maximum-a-posteriori (MAP) criterion [47], which can be described as

$$\hat{\mathbf{y}} = \underset{\mathbf{y} \in \mathbf{C_Y}}{\arg\max} f_{\mathrm{DL}}(\mathbf{y}|\mathbf{x}), \tag{1}$$

where $\mathbf{x}$ is the signal sample, and it is just the complex baseband signal sample $\mathbf{S}$ as the input of complex-valued

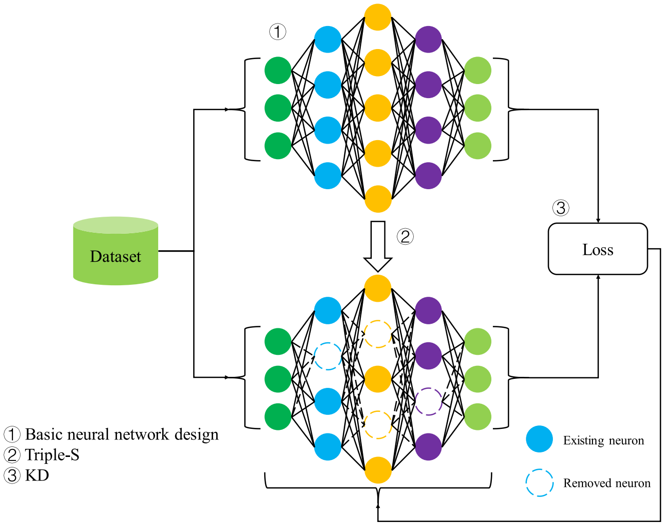


Fig. 4. The design of efficient SEI, which consists of three steps: basic neural network design, sparse structure selection (Triple-S), and knowledge distillation (KD).

neural networks, while it is the real-valued matrix, which consists of the real and imaginary parts of $\mathbf{S}$, i.e., $\mathbf{x} = \begin{Bmatrix} \text{real}(\mathbf{S}) \\ \text{imag}(\mathbf{S}) \end{Bmatrix}$; $\hat{\mathbf{y}}$ is the estimated specific emitter; $\mathbf{y}$ is the real specific emitter; $\mathbf{C_Y}$ is the collection of all specific emitters; $f(\cdot)$ is the mapping function between the input samples and labels. Here, DL is to build the mapping function $f_{\text{DL}}(\cdot)$ by large amounts of samples with the corresponding labels, and $f_{\text{DL}}(\cdot)$ can be considered as the trained DL model.

In this paper, we focus on time-domain SEI method based on steady-stat signals, and its implementation steps are given in Fig. 4. It consists of basic neural network design, sparse structure selection (Triple-S) (it is a neuron pruning technology), and KD. Basic neural network design is to build a redundant neural network via complex-valued convolution. Triple-S algorithm is to select an efficient structure from the basic neural network. Finally, KD algorithm is applied to recover the identification performance loss caused by Triple-S algorithm. These three parts will be introduced in next section.

IV. The Proposed Efficient SEI Methods

A. CNCNN-Based SEI

The first step for CVCNN-based SEI method is to construct the basic neural network. The structure of the original CVCNN is shown in Tab. I, which consists of nine convolution layers and two fully-connected layers.

It is noted that we apply small convolution kernel, whose kernel size is 3, and in order to make up for the lack of feature extraction capability caused by small convolution kernel, we adopt the scheme of increasing the depth of neural network, i.e., increasing the layers. In addition, max pooling operation follows behind every convolution layers for reducing feature dimensionality. These two tips are applied to construct the basic neural network and ensure the performance of SEI.

B. Triple-S Algorithm for Higher Efficiency

Triple-S algorithm is inspired by paper [35] and paper [36], which are sparse regularization-based neuron pruning methods for RVCNN in the field of computer vision, but it is difficult to directly apply these methods into CVCNN, because of its complex-valued weights. Thus, we introduce quantization mask [37] and proximal gradient method [38] to modify the original sparse regularization-based methods, for compressing model size and reducing the computational complexity of the CVCNN-based SEI method, which is shown in Fig. 5.

TABLE I
The Structure of CNN With Two Convolutional Layers and Three Fully-Connected Layers

Structure	The number of layers
Input (samples, labels)	-
CVConv1D (N, 3) + CVReLU + CVBN + Maxpooling1D (2)	×9
Flatten	×1
Dense (1024)+ ReLU + Dropout (0.5)	×1
Dense (7) + Softmax	×1

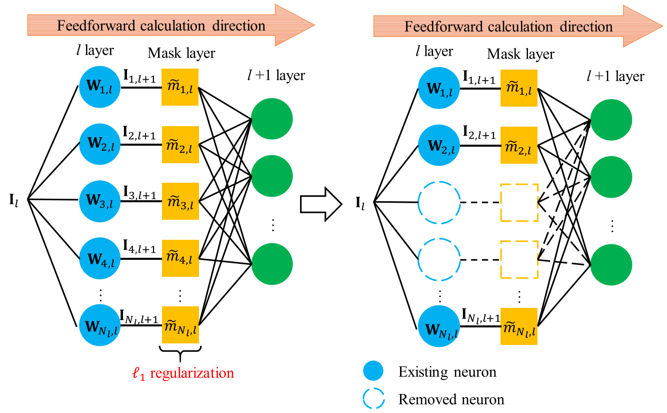


Fig. 5. The principle of Triple-S via mask layer and sparse regularization.

In Triple-S algorithm, ternary mask layers are inserted after all available layers of CVCNN, and every mask in the ternary mask layers has a corresponding neuron of CVCNN. In the training phase, all masks are constrained by regularization, and after training, these neurons, whose corresponding masks are zero, will be removed. CVCNN, after removing redundant neurons, is denoted as SlimCVCNN. In the next parts, the principle of ternary mask layer, loss function with sparse regularization, forward propagation and backward propagation will be introduced.

1) Ternary Mask Layer: Assuming that the total model weight is denoted as $\{\{\mathbf{W}_{i,l}\}_{i=1}^{N_l}\}_{l=1}^{L}$, the calculation process of the i-th neuron in the l-th layer (convolution layer or fully-connected layer) is given as

$$\mathbf{I}_{i,l+1} = f_l(\mathbf{W}_{i,l} * \mathbf{I}_l), \tag{2}$$

where $\mathbf{I}_l$ is the input; $\mathbf{I}_{i,l+1}$ is the output of the i-th neuron in the l-th layer (and it is also the part of the input of the $l+1$-th layer); $f_l(\cdot)$ includes many operations, such as CVBN, pooling and activation function. Ternary mask $\widetilde{m}_{i,l}$ is multiplied by every neuron in every available layer, which can be expressed as

$$\widetilde{\mathbf{I}}_{i,l+1} = \widetilde{m}_{i,l} \cdot \mathbf{I}_{i,l+1}. \tag{3}$$

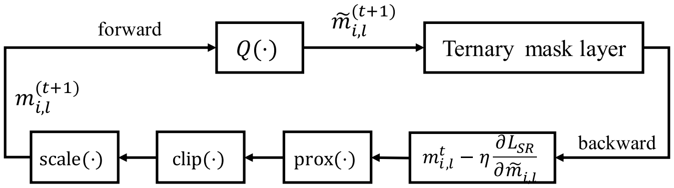

forward
$Q(\cdot)$
$\widetilde{m}_{i,l}^{(t+1)}$
Ternary mask layer
$m_{i,l}^{(t+1)}$
backward
scale$(\cdot)$
clip$(\cdot)$
prox$(\cdot)$
$m_{i,l}^{t} - \eta \frac{\partial L_{SR}}{\partial \widetilde{m}_{i,l}}$

Fig. 6. Forward propagation and backward propagation in the binary mask layer.

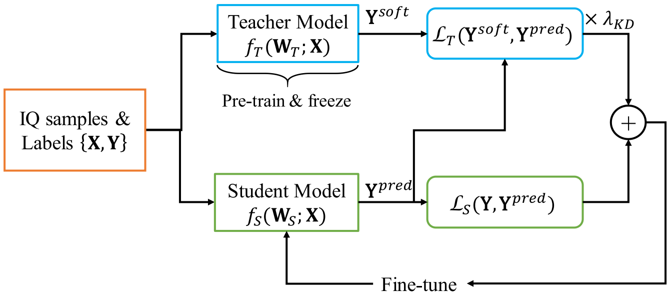

Teacher Model
$f_T(\mathbf{W}_T; \mathbf{X})$
Pre-train & freeze
$\mathbf{Y}^{soft}$
$\mathcal{L}_T(\mathbf{Y}^{soft}, \mathbf{Y}^{pred})$
$\times\, \lambda_{KD}$
IQ samples & Labels $\{\mathbf{X}, \mathbf{Y}\}$
+
Student Model
$f_S(\mathbf{W}_S; \mathbf{X})$
$\mathbf{Y}^{pred}$
$\mathcal{L}_S(\mathbf{Y}, \mathbf{Y}^{pred})$
Fine-tune

Fig. 7. The principle of KD.

It is noted that there are three possible values of $\widetilde{m}_{i,l}$, which are 0 and ± 1. In addition, when $\widetilde{m}_{i,l}$ is equal to 0, the neuron is invalid, and if $\widetilde{m}_{i,l}$ is equal to ± 1, it will have no effect on the output of the corresponding neurons.

2) Loss Function With Sparse Regularization: In order to force most of $\widetilde{m}_{i,l}$ to become zero, sparse regularization is imposed on $\widetilde{\mathbf{M}} = \{\{\widetilde{m}_{i,l}\}_{i=1}^{N_l}\}_{l=1}^{L}$. Given that $\{\mathbf{X}, \mathbf{Y}\} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{N_T}$ is a set, including N_T training samples and their corresponding labels, the loss function with ℓ_1 regularization can be written as

$$\begin{aligned}&\mathcal{L}_{SR}(f(\mathbf{W}, \widetilde{\mathbf{M}}); \{\mathbf{X}, \mathbf{Y}\}) \\ &\quad= -\frac{1}{N_T}\sum_{i=1}^{N_T} \mathbf{y}_i \log(f(\mathbf{W}, \widetilde{\mathbf{M}}; \mathbf{x}_i)) + \lambda_{SR}\mathrm{R}(\widetilde{\mathbf{M}}), \end{aligned} \tag{4}$$

where the first term represents identification performance, and the second term is sparse penalty, where $\mathrm{R}(\cdot)$ is ℓ_1 regularizer and λ_{SR} is a tunable super-parameter to balance these two terms [48]. It is noted that the larger λ_{SR}, the sparser the network structure, but it may result in irreversible performance loss. Thus, it is important for Triple-S algorithm to choose a suitable λ_{SR} for the aim of obtaining the advanced identification performance and a sparse network structure at the same time.

3) Forward Propagation and Backward Propagation: There are two steps of ternary mask layer in the forward propagation, which are shown in Fig. 6. The first step is to quantize the full-precision mask $m_{i,l}$ into ternary mask $\widetilde{m}_{i,l}$, which can be formulated as

$$\widetilde{m}_{i,l} = \mathrm{Q}(m_{i,l}) = \mathrm{sgn}(m_{i,l}) \cdot \mathrm{H}(|m_{i,l}| - \mu_Q), \tag{5}$$

where $\mathrm{H}(x) = \begin{cases} 0, & x \le 0 \\ 1, & x > 0 \end{cases}$ is the step function, and μ_Q is the quantization threshold. Then, according to Eq. (3), forward propagation is performed.

In the backward propagation, the weights of almost all layers, except ternary mask layers, are updated by stochastic gradient descent (SGD). However, SGD cannot optimize the ternary mask layers, because the gradient of $\mathbf{M}$ is almost equal to zero everywhere, which is shown as follows.

$$\begin{aligned}\mathbf{M}^{(t+1)} &= \mathbf{M}^{(t)} - \eta_t \cdot \frac{\partial \mathcal{L}_{SR}}{\partial \mathbf{M}} \\ &= \mathbf{M}^{(t)} - \eta_t \cdot \frac{\partial \mathcal{L}_{SR}}{\partial \widetilde{\mathbf{M}}} \cdot \frac{\partial \mathrm{Q}(\mathbf{M})}{\partial \mathbf{M}},\end{aligned} \tag{6}$$

where $\frac{\partial \mathrm{Q}(\mathbf{M})}{\partial \mathbf{M}}$ is almost equal to zero everywhere. Thus, inspired by quantization neural network (QNN), straight-through gradient descent (STGD) can be adopted to update the ternary weights of ternary mask layers, where STGD is just a common SGD with straight-through estimator (STE) [37], which is to estimate the gradient in the backward propagation. The STE of $\frac{\partial \mathcal{L}_{SR}}{\partial \mathbf{M}}$ can be written as

$$\begin{aligned}\mathrm{STE}(\frac{\partial \mathcal{L}_{SR}}{\partial \mathbf{M}}) &= \frac{\partial \mathcal{L}_{SR}}{\partial \widetilde{\mathbf{M}}} \cdot \frac{\partial \mathrm{Htanh}(\mathbf{M})}{\partial \mathbf{M}} \\ &= \frac{\partial \mathcal{L}_{SR}}{\partial \widetilde{\mathbf{M}}} \cdot 1_{\forall m_{i,l} \in \mathbf{M}, |m_{i,l}| \le 1},\end{aligned} \tag{7}$$

where $\mathrm{Htanh}(\cdot)$ can written as

$$\mathrm{Htanh}(x) = \begin{cases} 1, & |x| > 1 \\ x, & |x| \le 1 \end{cases} \tag{8}$$

which is applied to replace $\mathrm{Q}(\cdot)$ in the backward propagation. It means that $\frac{\partial \mathcal{L}_{SR}}{\partial \mathbf{M}}$ will be replaced with $\frac{\partial \mathcal{L}_{SR}}{\partial \widetilde{\mathbf{M}}}$ in the backward propagation, if $\forall m_{i,l} \in \mathbf{M}, |m_{i,l}|$ is less than or equal to 1.

However, the above STGD is not very suitable for minimizing the loss function Eq. (4) with ℓ_1 regularizer, which can be optimized by iterative shrinkage-thresholding algorithm (ISTA) [48], [49]. First, $\mathrm{Q}(\cdot)$ in Eq. (4) is approximate by $\mathrm{Htanh}(\cdot)$, which is written as

$$\begin{aligned}&\mathcal{L}_{SR}(f(\mathbf{W}, \widetilde{\mathbf{M}}); \{\mathbf{X}, \mathbf{Y}\}) \\ &\quad:= -\frac{1}{N_T}\sum_{i=1}^{N_T} \{\mathbf{y}_i \log(f(\mathbf{W}, \mathbf{M}; \mathbf{x}_i)) + \lambda_{SR}\mathrm{R}(\mathbf{M})\}, \\ &\quad s.\ t.\ \forall m_{i,l} \in \mathbf{M}, |m_{i,l}| \le 1\end{aligned} \tag{9}$$

The basic idea of ISTA is to project the parameter at every step of gradient descent to a potentially more sparse one subject to a proxy problem [38]. Thus, at the $t+1$-th step, the problem can be written as

$$\mathbf{M}^{(t+1)} = \min_{\mathbf{M}}\{\frac{1}{\eta_t}||\mathbf{M} - \mathbf{M}^{(t)} + \eta_t \frac{\partial \mathcal{L}_{SR}}{\partial \mathbf{M}}||^2 + \lambda_{SR} R(\mathbf{M})\}. \tag{10}$$

And Eq. (10) has closed form solution, which is written as follows.

$$\begin{aligned}\mathbf{M}^{(t+1)} &= \mathrm{prox}(\mathbf{M}^{(t)} - \eta_t \cdot \mathrm{STE}(\frac{\partial \mathcal{L}_{SR}}{\partial \mathbf{M}}); \eta_t \lambda_{SR}) \\ &= \mathrm{prox}(\mathbf{M}^{(t)} - \eta_t \cdot \frac{\partial \mathcal{L}_{SR}}{\partial \widetilde{\mathbf{M}}}; \eta_t \lambda_{SR}),\end{aligned} \tag{11}$$

Algorithm 1 Triple-S Algorithm for SlimCVCNN-Based SEI

```
Input: Complex baseband signals and labels $\{\mathbf{X}, \mathbf{Y}\}$;
Output: SlimCVCNN;
Construct CVCNN ($N = 64$) by Tab. I;
Insert ternary mask layers for every available layer in CVCNN ($N = 64$);
Set hyperparameters:
  • Set learning rate;
  • Set maximum epochs;
  • Set suitable $\lambda_{SR}$;
  • Set suitable $\mu_Q$;
  • Use ones initialization for ternary mask layers;
Training on $\{\mathbf{X}, \mathbf{Y}\}$:
for $t = 1$ to $T$ do:
Forward propagation:
    for $l = 1$ to $L$ do:
        for $i = 1$ to $N_l$ do:
            $\widetilde{m}_{i,l}^{(t)} \leftarrow \mathrm{Q}(m_{i,l}^{(t)})$;
            $\mathbf{I}_{l+1} \leftarrow \widetilde{m}_{i,l}^{(t)} \cdot f_l(\mathbf{W}_{i,l}^{(t)} * \mathbf{I}_l)$;
        end for
    end for
Backward propagation:
    Update model weights $\mathbf{W}$ by SGD:
        $\mathbf{W}^{(t+1)} \leftarrow \mathbf{W}^{(t)} - \eta_t \cdot \frac{\partial \mathcal{L}_{SR}}{\partial \mathbf{W}}$;
    Update masks $\mathbf{M}$ by PSTGD:
        $\mathbf{M}_{prox}^{(t+1)} \leftarrow \mathrm{prox}(\mathbf{M}^{(t)} - \eta_t \cdot \frac{\partial \mathcal{L}_{SR}}{\partial \widetilde{\mathbf{M}}}; \eta_t \lambda)$;
        $\mathbf{M}_{clip}^{(t+1)} \leftarrow \mathrm{clip}(\mathbf{M}_{prox}^{(t+1)}, -1, 1)$;
        $\mathbf{M}^{(t+1)} \leftarrow \mathrm{scale}(\mathbf{M}_{clip}^{(t+1)})$;
end for
Removing redundant neurons:
for $l = 1$ to $L$ do:
    for $i = 1$ to $N_l$ do:
        if $m_{i,l}^{B} == 0$ then:
            Removing the $i$-th neuron in the $l$-th layer of CVCNN ($N = 64$)
        end if
    end for
end for
return SlimCVCNN
```

where $\mathrm{prox}(x; \eta_t \lambda_{SR}) = \mathrm{sgn}(x) \cdot \max(|x| - \eta_t \lambda, 0)$; η_t is the learning rate. This optimization method is denoted as proximal STGD (PSTGD).

Moreover, to satisfy the constraint in Eq. 10, $\mathbf{M}$ is clipped by Eq. (12), after being updated by Eq. (11).

$$\mathrm{clip}(m_{i,l}, -1, 1) = \max(-1, \min(1, m_{i,l})), \quad \forall m_{i,l} \in \mathbf{M}. \tag{12}$$

Finally, scaling trick is applied to prevent all of $m_{i,l}$ from being too small and further prevent all of $\widetilde{m}_{i,l}$ from being zero in one layer. Scaling can be described as

$$\mathrm{scale}(m_{i,l}) = \frac{m_{i,l}}{\max(|m_{j,l}|)}, \quad \forall m_{i,l} \in \mathbf{M}. \tag{13}$$

C. Knowledge Distillation (KD) for Better Performance

KD algorithm is a kind of transfer learning, which is to transfer the knowledge from the large model to the small model [50], where the large model and small model are denoted as teacher model $f_T(\cdot)$ and student model $f_S(\cdot)$, respectively. In detail, KD is to design and minimize the loss function with the soft labels from teacher model and the one-hot labels from training dataset to improve performance. The loss function of KD is given as

$$\begin{aligned} &\mathcal{L}_{KD}(f_T(\mathbf{W}_T), f_S(\mathbf{W}_S); \{\mathbf{X}, \mathbf{Y}\}) \\ &\quad = \lambda_{KD} \sum_{\mathbf{x}_i \in \mathbf{X}} \mathcal{L}_T(f_T(\mathbf{x}_i; \mathbf{W}_T), f_S(\mathbf{x}_i; \mathbf{W}_S)) \\ &\qquad + \sum_{\substack{\mathbf{x}_i \in \mathbf{X} \\ \mathbf{y}_i \in \mathbf{Y}}} \mathcal{L}_S(\mathbf{y}_i, f_S(\mathbf{x}_i; \mathbf{W}_S)), \end{aligned} \tag{14}$$

where λ_{KD} ($\lambda_{KD} \geq 0$) is a tunable hyperparameter to balance two loss terms; $\mathcal{L}_T(\cdot)$ and $\mathcal{L}_S(\cdot)$ are two loss functions, and the former one is to measure the difference between teacher model and student model, while the latter one is to penalty the difference between the predicted probability distribution of the student model and the corresponding label; $\mathbf{y}_i^{pred} = f_S(\mathbf{x}_i; \mathbf{W}_S)$ is the predicted probability distribution of the student model $f_S(\mathbf{W}_S)$, and $\mathbf{W}_S$ is the student model weight; $\mathbf{y}_i^{soft} = f_T(\mathbf{x}_i; \mathbf{W}_T)$ is the soft label, which is the predicted probability distribution of the teacher model $f_T(\mathbf{W}_T)$. It is noted that $\mathbf{W}_T$ is the pre-trained weight of the teacher model, and it is frozen and untrainable in the process of KD.

Here, $\mathcal{L}_S(\cdot)$ is cross entropy (CE), and there are four possible loss functions to be putted for $\mathcal{L}_T(\cdot)$, which are mean square error (MSE), CE, Kullback-Leibler divergence (KL), and Jensen-Shannon divergence (JS). Four loss functions can be expressed as

$$\mathcal{L}_{MSE}(\mathbf{Y}^{soft}, \mathbf{Y}^{pred}) = \sum_{\substack{\mathbf{y}_i^{soft} \in \mathbf{Y}^{soft} \\ \mathbf{y}_i^{pred} \in \mathbf{Y}^{pred}}} ||\mathbf{y}_i^{soft} - \mathbf{y}_i^{pred}||_2^2, \tag{15}$$

$$\mathcal{L}_{CE}(\mathbf{Y}^{soft}, \mathbf{Y}^{pred}) = - \sum_{\substack{\mathbf{y}_i^{soft} \in \mathbf{Y}^{soft} \\ \mathbf{y}_i^{pred} \in \mathbf{Y}^{pred}}} \mathbf{y}_i^{soft} \log(\mathbf{y}_i^{pred}), \tag{16}$$

$$\mathcal{L}_{KL}(\mathbf{Y}^{soft}, \mathbf{Y}^{pred}) = \sum_{\substack{\mathbf{y}_i^{soft} \in \mathbf{Y}^{soft} \\ \mathbf{y}_i^{pred} \in \mathbf{Y}^{pred}}} \mathbf{y}_i^{soft} \log(\frac{\mathbf{y}_i^{soft}}{\mathbf{y}_i^{pred}}), \tag{17}$$

$$\begin{aligned} \mathcal{L}_{JS}(\mathbf{Y}^{soft}, \mathbf{Y}^{pred}) &= \frac{1}{2} \mathcal{L}_{KL}(\mathbf{Y}^{soft}, \frac{\mathbf{Y}^{pred} + \mathbf{Y}^{soft}}{2}) \\ &\quad + \frac{1}{2} \mathcal{L}_{KL}(\mathbf{Y}^{pred}, \frac{\mathbf{Y}^{pred} + \mathbf{Y}^{soft}}{2}), \end{aligned} \tag{18}$$

It is noted that the teacher model is pre-trained and the teacher model weight is frozen in the process of KD. The detailed algorithm of KD is shown in Alg. (2).

Algorithm 2 KD Algorithm for SlimCVCNN-Based SEI

```
Input: Complex baseband signals and labels $\{\mathbf{X}, \mathbf{Y}\}$;
Output: SlimCVCNN;
    Achieving the student model:
    Get the structure $f_{Slim}(\cdot)$ and weight $\mathbf{W}_{Slim}$ of SlimCVCNN from Alg. (1);
    Set SlimCVCNN as the student model ($f_S(\mathbf{W}_S) = f_{Slim}(\mathbf{W}_{Slim})$);
    Achieving the teacher model:
    Construct CVCNN ($N = 128$) $f_T(\mathbf{W}_T)$ as the teacher model, and train it on $\{\mathbf{X}, \mathbf{Y}\}$;
    KD:
    Choose a proper loss function $\mathcal{L}_T$ between the predicted labels of the teacher model and the predicted labels of the student labels;
    Choose CE as the loss function between the real labels and the predicted labels of the student labels;
    Set a proper parameter $\lambda_{KD}$;
    Freeze the teacher model weight ($\mathbf{W}_T$ is unchangeable in the process of KD);
    Update $\mathbf{W}_S$ for minimizing Eq. (14);
    return SlimCVCNN
```

TABLE II

Simulation Hyperparameters

Hyperparameter	Values
Training samples of seven PAs ("PA1"∼"PA7")	$\{3.2, 5.2, 3.8, 3.2, 3.2, 3.8, 3.8\}\times10^4$
Ratio of training and validation	7:3
Test samples per PA	30, 000
The length of samples K	1, 000
SNRs	{10, 15, 20, 25, 30, 35, 40}dB
Maximum training epoch	200 (Sec. A)/400 (Sec. B)/20 (Sec. C)
Learning rate η	0.01
Sparse penalty factor λ_{SR}	$\{1\times10^{-2}, 1\times10^{-3}, 1\times10^{-4}\}$
Quantization threshold μ_Q	0.5

V. Simulation Results

A. Implementation Platform and Simulation Hyperparameters

The simulation is based on Keras library and GTX1080Ti, and the pre-processing of dataset is performed by Matlab. Other simulation hyperparameters are listed in Tab. II.

B. RVCNN Vs. CVCNN

1) Identification Performance Vs. SNR: The identification performance is shown in Fig. 8. Our proposed RVCNN has better identification performance than the other real-valued CNN, including CNN1 [26], MSCNN1 [26] and CNN2 [25], and its performance is also better than the TD feature-based traditional method. More importantly, CVCNN, which has the similar structure of RVCNN and just replaces real-valued convolution layer with complex-valued convolution layer, outperforms RVCNN. The maximum performance gap between them can reach up to almost 5.8% at SNR = 25 dB.

Here, two CVCNNs with different parameters are provided for the fair comparison, because under the condition of the same neurons, CVCNN ($N = 128$) is more complex than RVCNN ($N = 128$), while CVCNN ($N = 64$) has the similar computing complexity and parameters with RVCNN ($N = 128$). The detailed complexity analysis is shown in the next section. In terms of the identification performance, the accuracy of CVCNN ($N = 64$) is slightly lower than that of CVCNN ($N = 128$) at SNR = 10 dB or SNR = 15 dB, and the maximum accuracy gap between them is at SNR = 10 dB, which is about 1%. When SNR is higher than 25 dB, the accuracy of two CVCNNs can both exceed 99% in this dataset.

2) Complexity Analysis: Assuming that there are C_l ($l \in [1, L_C]$) neurons in the l-th convolution layer, whose corresponding kernel size and output dimensionality are D_l^k and N_l^{out}; F_l ($l \in [1, L_F]$) is the number of neurons in the fully-connected layers, the computing complexity of one real-valued convolution layer C^{RV}, one complex-valued convolution layer C^{CV} and fully-connected layer C^{FC} can be written as

$$C^{RV} \sim O(D_l^k \cdot N_l^{out} \cdot C_{l-1} \cdot C_l), \tag{19}$$

$$C^{CV} \sim O(4 \cdot D_l^k \cdot N_l^{out} \cdot C_{l-1} \cdot C_l), \tag{20}$$

$$C^{FC} \sim O(F_{l-1} \cdot F_l). \tag{21}$$

Due to the low computing complexity of BN layer, pooling layer and activation function, the computing complexity is simplified to the sum of the computing complexity of all convolution layers and all join layers, and we adopt the multiply-accumulate operation (MACC) to represent them, which are listed as follows.

$$MACC^{RVCNN} = \sum_{l=1}^{L_C} D_l^k N_l^{out} C_{l-1} C_l + \sum_{l=1}^{L_F} F_{l-1} F_l, \tag{22}$$

$$MACC^{CVCNN} = \sum_{l=1}^{L_C} 4 D_l^k N_l^{out} C_{l-1} C_l + \sum_{l=1}^{L_F} F_{l-1} F_l, \tag{23}$$

where C^0 is the number of sample channels, which is one in the CVCNN or two in the RVCNN; F^0 is the output dimensionality of the last convolution layer. Thus, the MACCs of six neuron networks are shown in Tab. III, where parameters are automatically counted by Keras.

If ignoring BN, pooling operation and so on, which have low computing complexity, CVCNN ($N = 64$) will have the same MACCs with RVCNN ($N = 128$), and CVCNN ($N = 128$) can achieve four times computing complexity of CVCNN ($N = 64$) or RVCNN ($N = 128$). In addition, as the result of the complex-valued input of CVCNN, which just has half of the input channels of RVCNN, CVCNN ($N = 64$) has fewer parameters than RVCNN ($N = 128$).

3) Loss Vs. Epoch: Except the superior identification performances of our proposed RVCNN and CVCNNs, Fig. 9 has demonstrated the advantages of convergence speed, where the training loss and accuracy curves in 200 epochs are given.

TABLE III

PARAMETERS AND MACCs OF RVCNN, CVCNN, AND TWO CNNs PROPOSED IN [25], [26]

Method	Parameters	MACC
RVCNN (N=128)	539,015	49,714,176
CVCNN (N=128)	1,071,367	197,037,056
CVCNN (N=64)	343,175	49,714,176
CNN1 [26]	1,182,343	134,484,864
MSCNN1 [26]	3,544,967	236,281,728
CNN2 [25]	330,423	7,810,160

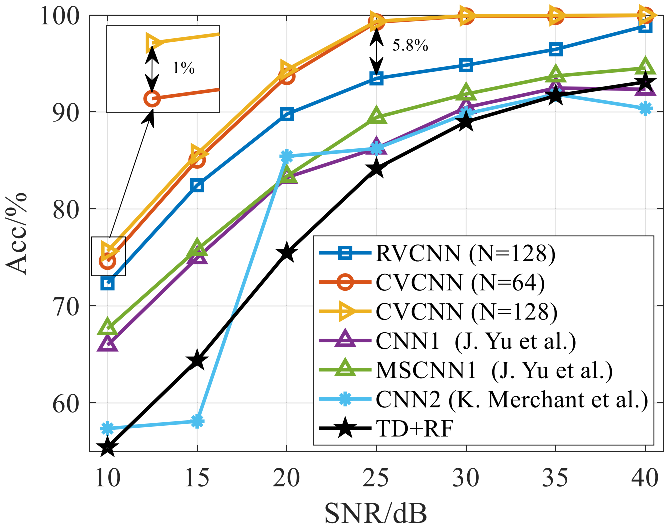


Fig. 8. The identification performances of various SEI methods, including two kinds of CVCNN, RVCNN, CNN1 [26], MSCNN1 [26], CNN2 [25] and a traditional method based on TD features [51] and RF classifier.

At SNR = 40 dB, the convergence of two CVCNNs is slightly faster than that of RVCNN, and two CVCNNs have the similar convergence speed. Unfortunately, CNN1 [26], MSCNN1 [26] and CNN2 [25] have slower convergence speed than RVCNN and two CVCNNs. In detail, the training epochs for convergence in RVCNN and two CVCNNs are fewer than 50, while that of CNN1, MSCNN1 and CNN2 are generally more than 100.

C. SlimCVCNN

Compared with CVCNN ($N = 128$), CVCNN ($N = 64$) just has 1% identification performance loss at most, but its computing complexity and parameters are only $1/4$ and $1/3$ of the former. Thus, Triple-S algorithm is performed on CVCNN ($N = 64$) for SlimCVCNN with more efficient network structure and the similar identification performance.

Before introducing the detailed results, two metrics are introduced to measure the compression performance of SlimCVCNN: 1) the parameter compression ratio R_{param}; 2) the computing compression ratio R_{MACC}. They are defined as follows.

$$R_{Param} = \frac{Param_{Original} - Param_{Slim}}{Param_{Original}} \times 100\%, \tag{24}$$

$$R_{MACC} = \frac{MACC_{Original} - MACC_{Slim}}{MACC_{Original}} \times 100\%, \tag{25}$$

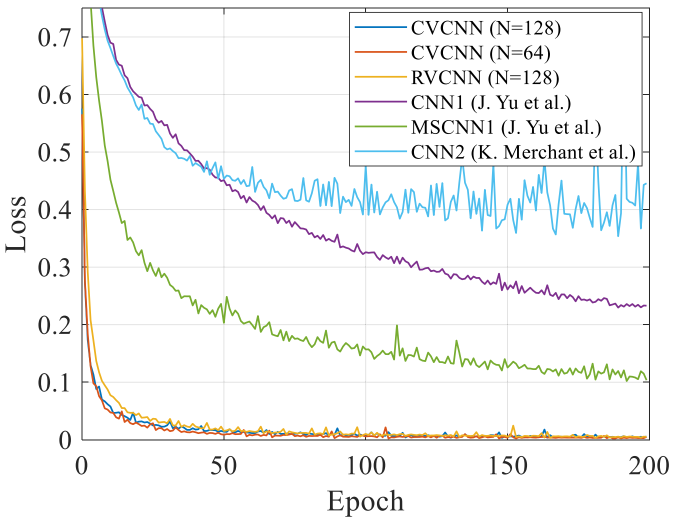


Fig. 9. The training loss of different CNN-based SEI methods at SNR = 40 dB.

where $Param_{Original}$, $MACC_{Original}$, $Param_{Slim}$, and $MACC_{Slim}$ represent the parameters and the MACCs of the original CVCNN ($N = 64$) and SlimCVCNN.

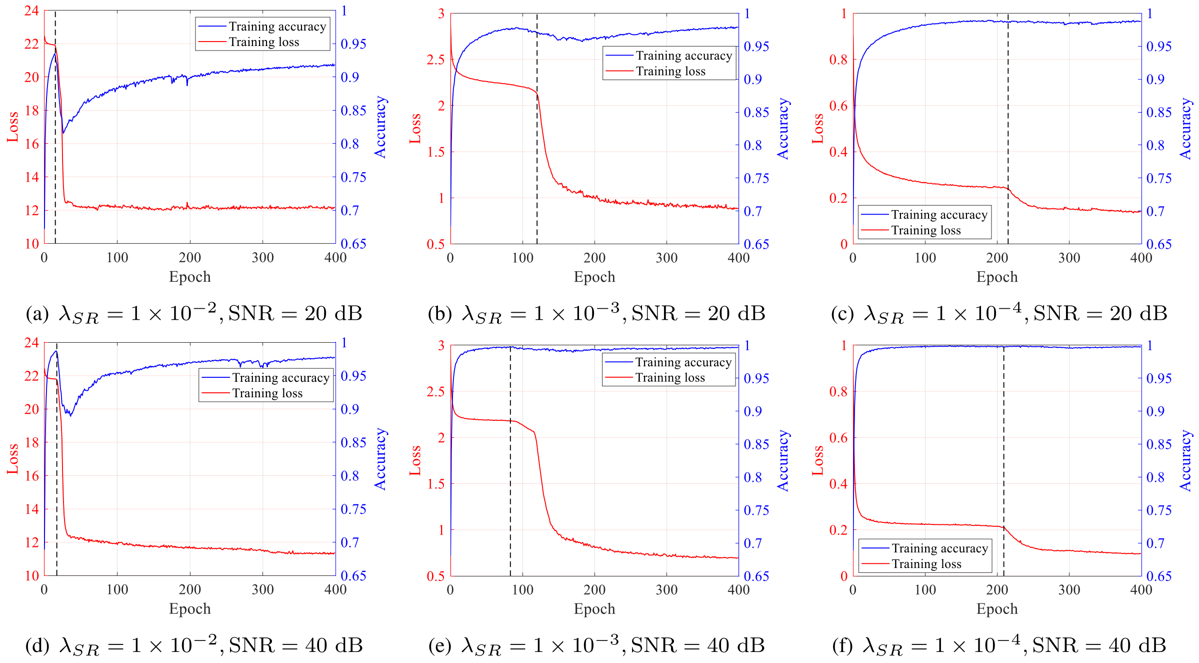
1) The Influence of λ_{SR}: In SlimCVCNN, one of the most important hyperparameters is λ_{SR}, which can balance identification performance and complexity. Here, we choose three possible values $\{1 \times 10^{-2}, 1 \times 10^{-3}, 1 \times 10^{-4}\}$, and the influences of λ_{SR} are shown in Fig. 10 and Tab. IV.

It can be observed that with the increasing of λ_{SR}, R_{param} and R_{MACC} both increase in Tab. IV. However, when λ_{SR} is equal to 1×10^{-2}, the identification performance declines seriously, but $\lambda_{SR} = 1 \times 10^{-3}$ or $\lambda_{SR} = 1 \times 10^{-4}$ has little impact on identification performance of SlimCVCNN. In detail, ΔAcc_{64} and ΔAcc_{128} are defined to reveal the performance gaps between CVCNN ($N = 64$) and SlimCVCNN and between CVCNN ($N = 128$) and SlimCVCNN, respectively. ΔAcc_{64} and ΔAcc_{128} are almost 50% or 70% under the condition of $\lambda_{SR} = 1 \times 10^{-2}$, but they will be far less than 1%, if λ_{SR} is equal to 1×10^{-3} or 1×10^{-4}.

In addition, it can be observed from Fig. 10 that the training process of SlimCVCNN can be divided into two phases by two turning points. The first phase is mainly to reduce training loss by improving identification performance, and the second phase focuses on the improvement of network sparsity to further reduce training loss. If λ_{SR} is too large, there will be a serious degradation in identification performance at the second phase, which can be demonstrated in Fig. 10a and Fig. 10d.

Moreover, λ_{SR} can also affect the convergence rate. It can be demonstrated by Fig. 10 that the number of epochs in the first phase is less than 20, about 100, and more than 200, when λ_{SR} is equal to 1×10^{-2}, 1×10^{-3} and 1×10^{-4}, respectively. It means that with the degradation of λ_{SR}, the duration of the first phase of training process will be longer, and the second phase will come latter.

2) The Influence of Scaling Operation: Scaling operation is an important trick, and its influences are shown in Fig. 11. After less than 30 epochs, the training accuracy drops seriously, and the training accuracy and the training



(a) $\lambda_{SR} = 1 \times 10^{-2}$, SNR = 20 dB (b) $\lambda_{SR} = 1 \times 10^{-3}$, SNR = 20 dB (c) $\lambda_{SR} = 1 \times 10^{-4}$, SNR = 20 dB

(d) $\lambda_{SR} = 1 \times 10^{-2}$, SNR = 40 dB (e) $\lambda_{SR} = 1 \times 10^{-3}$, SNR = 40 dB (f) $\lambda_{SR} = 1 \times 10^{-4}$, SNR = 40 dB

Fig. 10. Evaluation of the training accuracy and loss curves of CVCNN ($N = 64$) with Triple-S algorithm under different λ_{SR}.

TABLE IV
THE STRUCTURES, PARAMETERS, MACCS AND ACCURACY OF SLIMCVCNN AT DIFFERENT SPARSE PENALTY FACTOR, WHEN SNR = 40 DB

λ_{SR}/SNR (dB)	Structure	Parameters (R_{Param})	MACC (R_{MACC})	*Acc* (ΔAcc_{128}/ΔAcc_{64})
1×10^{-2}/20	[41, 56, 8, 62, 11, 64, 14, 4, 6, 114]	41,398 (87.93%↓)	17,294,310 (65.21%↓)	44.85% (49.51%↓/48.82%↓)
1×10^{-3}/20	[31, 64, 57, 52, 44, 53, 37, 23, 14, 47]	105,282 (80.63%↓)	30,673,477 (38.30%↓)	94.73% (0.37%↑/1.06%↑)
1×10^{-4}/20	[37, 64, 64, 64, 62, 62, 54, 47, 25, 41]	161,952 (52.80%↓)	38,326,629 (22.91%↓)	93.73% (0.63%↓/0.06%↑)
1×10^{-2}/40	[10, 64, 9, 53, 20, 28, 16, 12, 47, 45]	35,430 (89.67%↓)	7,522,077 (84.87%↓)	28.71% (71.27%↓/71.25%↓)
1×10^{-3}/40	[20, 64, 33, 30, 30, 29, 27, 20, 7, 54]	50,699 (85.23%↓)	16,926,714 (65.95%↓)	99.84% (0.14%↓/0.12%↓)
1×10^{-4}/40	[38, 64, 64, 50, 51, 51, 39, 27, 23, 43]	119,550 (78.17%↓)	35,471,879(28.64%↓)	99.98% (0.00%/0.02%↑)

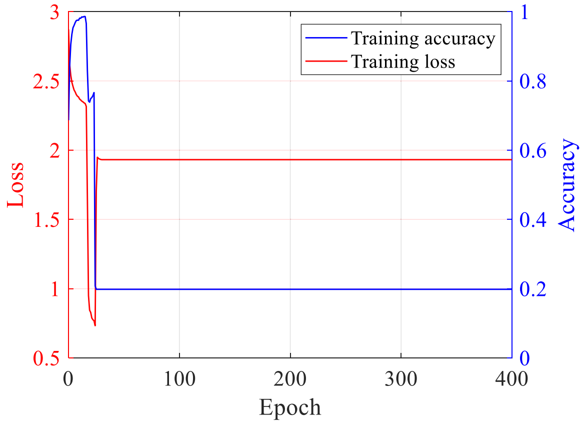


Fig. 11. The accuracy and loss curves of Triple-S algorithm without scaling operation and $\lambda_{SR} = 1 \times 10^{-3}$ at SNR = 40 dB.

loss remain static. If there is no scaling operation, all masks in one layer may become zero, and this phenomenon will happen. However, scaling operation can ensure that at least one ternary mask is not zero, and make sure the training continue.

3) PSTGD Vs. STGD: Here, PSTGD is compared with STGD in the process of updating the ternary mask layers, and the related simulation results are shown in Fig. 12 and Tab. VI. STGD has slower convergence speed than PSTGD, which has been demonstrated by Fig. 12, and PSTGD just needs to spend about 100 epochs to achieve the second turning point, but STGD requires more than 200 epochs.

In addition, the final training loss of SlimCVCNN optimized by PSTGD is much less than that optimized by STGD. It means that the complexity of final model optimized by PSTGD is far lower than that optimized by STGD, which it is also demonstrated by Tab. VI. In detail, the number of neurons in each layer of SlimCVCNN based on PSTGD are less than that based on STGD, and the parameter and MACC of the former are both far less than that of the latter.

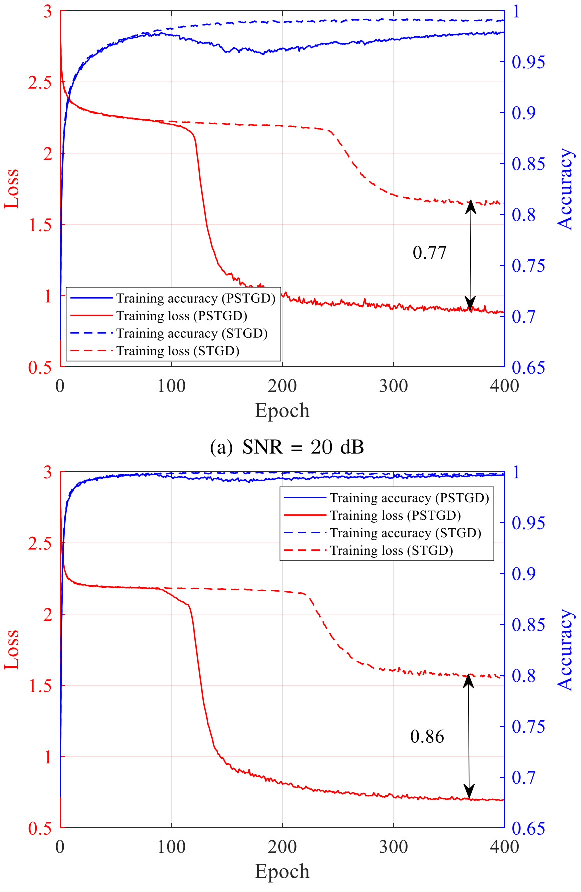
4) Identification Performance and Model Complexity: Here, SlimCVCNN is performed under the condition of $\lambda_{SR} = 1 \times 10^{-3}$, and its identification performance and model complexity are shown in Tab. VI. The structures of SlimCVCNN are slimmer than that of the original CVCNN ($N = 64$), which just have $10\% \sim 30\%$ parameters of the original CVCNN ($N = 64$), and there are also $7\% \sim 66\%$ declines about the MACCs of SlimCVCNN at different SNRs.

Moreover, the identification performance of SlimCVCNN is similar with that of CVCNN ($N = 64$) and CVCNN

TABLE V

THE STRUCTURES, PARAMETERS, MACCS AND ACCURACY OF SLIMCVCNN WITH PSTGD AND STGD

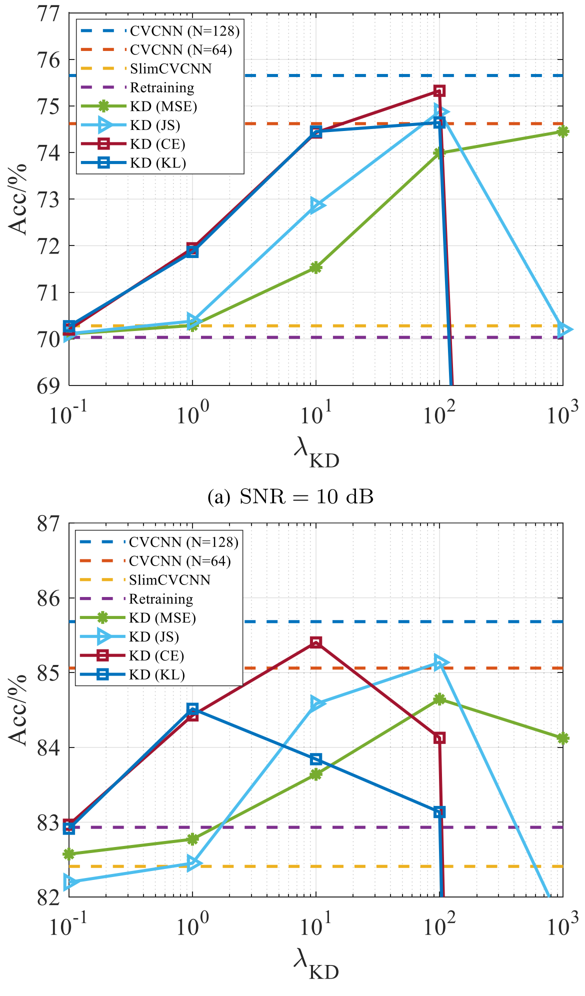
Optimizer/SNR (dB)	Structure	Parameters (R_{Param})	MACC (R_{MACC})	Acc ($\Delta Acc_{128}/\Delta Acc_{64}$)
PSTGD/20	[31, 64, 57, 52, 44, 53, 37, 23, 14, 47]	105,282 (80.63%↓)	30,673,477 (38.30%↓)	94.73% (0.37%↑/1.06%↑)
STGD/20	[56, 64, 64, 64, 64, 64, 60, 58, 39, 125]	195,980 (42.89%↓)	46,254,713 (6.96%↓)	87.83% (6.53%↓/5.84%↓)
PSTGD/40	[20, 64, 33, 30, 30, 29, 27, 20, 7, 54]	50,699 (85.23%↓)	16,926,714 (65.95%↓)	99.84% (0.14%↓/0.12%↓)
STGD/40	[52, 64, 63, 58, 55, 60, 60, 51, 42, 82]	172,851 (49.63%↓)	42,759,574 (13.99%↓)	99.70% (0.28%↓/0.26%↓)



(a) SNR = 20 dB

(b) SNR = 40 dB

Fig. 12. The accuracy and loss curves of SlimCVCNN optimized by PSTGD and STGD, respectively.



(a) SNR = 10 dB

(b) SNR = 15 dB

Fig. 13. The identification performance of KD with various loss functions under different λ_{KD}.

($N = 128$), particularly when SNR is high. In detail, when SNR is larger than 20 dB, the maximum identification performance loss between SlimCVCNN and CVCNN ($N = 64$) or CVCNN ($N = 128$) is only 0.14%, and SlimCVCNN even has performance improvements at some SNRs, such as 20 dB and 25 dB. However, when SNR is lower than 20 dB, the identification performance loss of SlimCVCNN can reach up to 2% $\sim$ 5%, but the performance loss is also very slight, which can be made up by retraining or KD, and there is no need to adjust λ_{SR}. Thus, the following section will introduce different performance compensation methods at low SNRs.

D. KD Vs. Retraining

Due to the identification performance degradation at low SNRs, retraining or KD is considered applied to recover performance within few epochs, and the specified simulation results about retraining and KD are given in Fig. 13. Here, the number of epochs for retraining and KD is set as 20, and CVCNN ($N = 128$) is as teacher model to improve the performance of SlimCVCNN in the process of KD.

It can be observed from Fig. 13a and Fig. 13b that retraining can slightly improve the identification performance of SlimCVCNN at SNR $=$ 15 dB, while it may reduce the identification performance at SNR $=$ 10 dB. However, when choosing suitable loss function and appropriate λ_{KD}, the identification performance of SlimCVCNN after KD will exceed that of CVCNN ($N = 64$) and be close to that of CVCNN ($N = 128$). Here, in order to select the best combination

TABLE VI

THE STRUCTURES, PARAMETERS, MACCs AND ACCURACY OF CVCNN ($N = 64$) WITH TRIPLE-S AT DIFFERENT SNRs

SNR(dB)	Structure	Parameters (R_{Param})	MACC (R_{MACC})	Acc^i ($\Delta Acc^i_{128}/\Delta Acc^i_{64}$)
10	[56, 64, 63, 64, 64, 64, 62, 50, 21, 69]	178,722 (69.06%↓)	45,906,957 (7.65%↓)	70.28% (4.59%↓/4.34%↓)
15	[34, 64, 45, 49, 62, 61, 53, 46, 22, 43]	133,178 (76.10%↓)	29,904,033 (39.84%↓)	82.41% (2.65%↓/2.75%↓)
20	[31, 64, 57, 52, 44, 53, 37, 23, 14, 47]	105,282 (80.63%↓)	30,673,477 (38.30%↓)	94.73% (0.37%↑/1.06%↑)
25	[22, 64, 47, 35, 31, 36, 26, 24, 9, 33]	65,374 (87.02%↓)	21,655,389 (56.44%↓)	99.54% (0.18%↑/0.27%↑)
30	[27, 64, 37, 33, 33, 35, 29, 20, 8, 61]	61,594 (87.40%↓)	21,105,959 (57.66%↓)	99.84% (0.08%↓/0.04%↓)
35	[22, 64, 34, 36, 32, 35, 26, 15, 5, 48]	55,547 (88.61%↓)	18,549,804 (62.83%↓)	99.89% (0.06%↓/0.01%↑)
40	[20, 64, 33, 30, 30, 29, 27, 20, 7, 54]	50,699 (89.23%↓)	16,926,714 (66.08%↓)	99.84% (0.14%↓/0.12%↓)

of loss function and λ_{KD}, we compare the identification performance of SlimCVCNN after KD under the condition of four loss functions and different λ_{KD}. It is obvious that when λ_{KD} is set as 10^2 at SNR$=10$ dB and 10^1 at SNR $= 15$ dB, and CE is chosen as loss function, SlimCVCNN after KD can achieve the best performance.

In detail, the accuracy of SlimCVCNN after KD can reach up to 75.33% and 85.40% at SNR $= 10$ dB and 15 dB, respectively. When compared with CVCNN ($N = 64$), SlimCVCNN after KD has 0.71% and 0.34% performance improvement at SNR $= 10$ dB and 15 dB, respectively. Moreover, there are only 0.33% and 0.28% performance gaps between CVCNN ($N = 128$) and SlimCVCNN at SNR $= 10$ dB and 15 dB, respectively.

Based on the above simulation results, KD can improve the identification performances of SlimCVCNN, but retraining cannot achieve such identification performance. It means that it is hard for SlimCVCNN to learn knowledge by retraining from training samples with real labels, and soft labels in KD contain useful knowledge, which can bring the performance improvement.The above simulation results have demonstrated the effectiveness of KD.

VI. CONCLUSION

An efficient CVCNN-based SEI method is proposed, which can achieve nearly perfect identification accuracy at high SNRs and confirms superiority to the previously proposed CNN-based SEI method in both identification performance and convergence. The proposed method has lower computing complexity and smaller model size. In detail, the original CVCNN has redundant neurons, and Triple-S algorithm is proposed to select a low complexity structure (i.e. SlimCVCNN) from the original CVCNN, and it also can retain most of performance of CVCNN. And KD algorithm is introduced to recover performance loss in Triple-S algorithm at low SNRs.

REFERENCES

[1] O. Dobre, "Signal identification for emerging intelligent radios: Classical problems and new challenges," *IEEE Instrum. Meas. Mag.*, vol. 18, no. 2, pp. 11–18, Apr. 2015.

[2] I. O. Kennedy, P. Scanlon, F. J. Mullany, M. M. Buddhikot, K. E. Nolan, and T. W. Rondeau, "Radio transmitter fingerprinting: A steady state frequency domain approach," in *Proc. IEEE 68th Veh. Technol. Conf.*, Calgary, BC, Canada, Sep. 2008, pp. 1–5.

[3] Q. Xu, R. Zheng, W. Saad, and Z. Han, "Device fingerprinting in wireless networks: Challenges and opportunities," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 94–104, 1st Quart., 2016.

[4] A. C. Polak, S. Dolatshahi, and D. L. Goeckel, "Identifying wireless users via transmitter imperfections," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 7, pp. 1469–1479, Aug. 2011.

[5] J. Sun, W. Shi, Z. Yang, J. Yang, and G. Gui, "Behavioral modeling and linearization of wideband RF power amplifiers using BiLSTM networks for 5G wireless systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 11, pp. 10348–10356, Nov. 2019.

[6] O. León, J. Hernández-Serrano, and M. Soriano, "Securing cognitive radio networks," *Int. J. Commun. Syst.*, vol. 23, no. 5, pp. 633–652, Feb. 2010.

[7] D. A. Knox and T. Kunz, "RF fingerprints for secure authentication in single-hop WSN," in *Proc. IEEE Int. Conf. Wireless Mobile Comput., Netw. Commun.*, Avignon, France, Oct. 2008, pp. 567–573.

[8] J. Zhang, G. Li, A. Marshall, A. Hu, and L. Hanzo, "A new frontier for IoT security emerging from three decades of key generation relying on wireless channels," *IEEE Access*, vol. 8, pp. 138406–138446, Jul. 2020.

[9] Q. Tian *et al.*, "New security mechanisms of high-reliability IoT communication based on radio frequency fingerprint," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 7980–7987, Oct. 2019.

[10] Y. Lin, J. Jia, S. Wang, B. Ge, and S. Mao, "Wireless device identification based on radio frequency fingerprint features," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Dublin, Ireland, Jun. 2020, pp. 1–6.

[11] Y. Zhang, Y. Lin, Z. Dou, M. Wang, and W. Li, "Monitoring and identification of WiFi devices for Internet of Things security," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Waikoloa, HI, USA, Dec. 2019, pp. 1–5.

[12] S. Chen, F. Xie, Y. Chen, H. Song, and H. Wen, "Identification of wireless transceiver devices using radio frequency (RF) fingerprinting based on STFT analysis to enhance authentication security," in *Proc. IEEE 5th Int. Symp. Electromagn. Compat. (EMC-Beijing)*, Beijing, China, Oct. 2017, pp. 1–5.

[13] Y. Tu, Z. Zhang, Y. Li, C. Wang, and Y. Xiao, "Research on the Internet of Things device recognition based on RF-fingerprinting," *IEEE Access*, vol. 7, pp. 37426–37431, Mar. 2019.

[14] D. R. Reising, M. A. Temple, and J. A. Jackson, "Authorized and rogue device discrimination using dimensionally reduced RF-DNA fingerprints," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 6, pp. 1180–1192, Jun. 2015.

[15] L. Liang, H. Ye, G. Yu, and G. Y. Li, "Deep-learning-based wireless resource allocation with application to vehicular networks," *Proc. IEEE*, vol. 108, no. 2, pp. 341–356, Feb. 2020.

[16] J. Guo, J. Wang, C.-K. Wen, S. Jin, and G. Y. Li, "Compression and acceleration of neural networks for communications," *IEEE Wireless Commun.*, vol. 27, no. 4, pp. 110–117, Aug. 2020.

[17] G. Gui, M. Liu, F. Tang, N. Kato, and F. Adachi, "6G: Opening new horizons for integration of comfort, security and intelligence," *IEEE Wireless Commun.*, vol. 27, no. 5, pp. 126–132, Oct. 2020.

[18] H. Ye, G. Y. Li, and B.-H. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 1, pp. 114–117, Feb. 2018.

[19] H. Huang, Y. Peng, J. Yang, W. Xia, and G. Gui, "Fast beamforming design via deep learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 1065–1069, Jan. 2020.

[20] J. Guo, C.-K. Wen, S. Jin, and G. Y. Li, "Convolutional neural network-based multiple-rate compressive sensing for massive MIMO CSI feedback: Design, simulation, and analysis," *IEEE Trans. Wireless Commun.*, vol. 19, no. 4, pp. 2827–2840, Apr. 2020.

[21] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cognit. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, Dec. 2017.

[22] F. Meng, P. Chen, L. Wu, and X. Wang, "Automatic modulation classification: A deep learning enabled approach," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 10760–10772, Nov. 2018.

[23] Y. Wang, J. Yang, M. Liu, and G. Gui, "LightAMC: Lightweight automatic modulation classification via deep learning and compressive sensing," *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 3491–3495, Mar. 2020.

[24] Y. Tu, Y. Lin, C. Hou, and S. Mao, "Complex-valued networks for automatic modulation classification," *IEEE Trans. Veh. Technol.*, vol. 69, no. 9, pp. 10085–10089, Sep. 2020.

[25] K. Merchant, S. Revay, G. Stantchev, and B. Nousain, "Deep learning for RF device fingerprinting in cognitive communication networks," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 160–167, Feb. 2018.

[26] J. Yu, A. Hu, G. Li, and L. Peng, "A robust RF fingerprinting approach using multisampling convolutional neural network," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6786–6799, Aug. 2019.

[27] F. Xie, H. Wen, J. Wu, S. Chen, W. Hou, and Y. Jiang, "Convolution based feature extraction for edge computing access authentication," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 4, pp. 2336–2346, Oct. 2020.

[28] L. Ding, S. Wang, F. Wang, and W. Zhang, "Specific emitter identification via convolutional neural networks," *IEEE Commun. Lett.*, vol. 22, no. 12, pp. 2591–2594, Dec. 2018.

[29] L. Peng, J. Zhang, M. Liu, and A. Hu, "Deep learning based RF fingerprint identification using differential constellation trace figure," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 1091–1095, Jan. 2020.

[30] B. He and F. Wang, "Cooperative specific emitter identification via multiple distorted receivers," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3791–3806, Jun. 2020.

[31] J. Gong, X. Xu, and Y. Lei, "Unsupervised specific emitter identification method using radio-frequency fingerprint embedded InfoGAN," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 2898–2913, Mar. 2020.

[32] I. Agadakos, N. Agadakos, J. Polakis, and M. R. Amer, "Chameleons' oblivion: Complex-valued deep neural networks for protocol-agnostic RF device fingerprinting," in *Proc. IEEE Eur. Symp. Secur. Privacy (EuroS&P)*, Genoa, Italy, Sep. 2020, pp. 322–338.

[33] S. Gopalakrishnan, M. Cekic, and U. Madhow, "Robust wireless fingerprinting via complex-valued neural networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Waikoloa, HI, USA, Dec. 2019, pp. 1–6.

[34] J. Stankowicz, J. Robinson, J. M. Carmack, and S. Kuzdeba, "Complex neural networks for radio frequency fingerprinting," in *Proc. IEEE Western New York Image Signal Process. Workshop (WNYISPW)*, Rochester, NY, USA, Oct. 2019, pp. 1–5.

[35] Y. He, X. Zhang, and J. Sun, "Channel pruning for accelerating very deep neural networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 1389–1397.

[36] Z. Huang and N. Wang, "Data-driven sparse structure selection for deep neural networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Munich, Germany, 2018, pp. 304–320.

[37] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks: Training deep neural networks with weights and activations constrained to $+1$ or -1," 2016, *arXiv:1602.02830*. [Online]. Available: http://arxiv.org/abs/1602.02830

[38] J. Ye, X. Lu, Z. Lin, and J. Z. Wang, "Rethinking the smaller-norm-less-informative assumption in channel pruning of convolution layers," 2018, *arXiv:1802.00124*. [Online]. Available: http://arxiv.org/abs/1802.00124

[39] Y. Yang, F. Gao, G. Y. Li, and M. Jian, "Deep learning-based downlink channel prediction for FDD massive MIMO system," *IEEE Commun. Lett.*, vol. 23, no. 11, pp. 1994–1998, Nov. 2019.

[40] P. E. Gorday, N. Erdol, and H. Zhuang, "Complex-valued neural networks for noncoherent demodulation," *IEEE Open J. Commun. Soc.*, vol. 1, pp. 217–225, Jan. 2020.

[41] P. J. Freire *et al.*, "Complex-valued neural network design for mitigation of signal distortions in optical links," *J. Lightw. Technol.*, vol. 39, no. 6, pp. 1696–1705, Mar. 15, 2021, doi: 10.1109/JLT.2020.3042414.

[42] J. Guo, J. Wang, C.-K. Wen, S. Jin, and G. Y. Li, "Compression and acceleration of neural networks for communications," *IEEE Wireless Commun.*, vol. 27, no. 4, pp. 110–117, Aug. 2020.

[43] Y. Lin, Y. Tu, and Z. Dou, "An improved neural network pruning technology for automatic modulation classification in edge devices," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5703–5706, May 2020.

[44] H. Xie and Z. Qin, "A lite distributed semantic communication system for Internet of Things," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 142–153, Jan. 2021.

[45] Z. Cao, W.-T. Shih, J. Guo, C.-K. Wen, and S. Jin, "Lightweight convolutional neural networks for CSI feedback in massive MIMO," 2020, *arXiv:2005.00438*. [Online]. Available: http://arxiv.org/abs/2005.00438

[46] X. Zhang, X. Zhou, M. Lin, and J. Sun, "ShuffleNet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 6848–6856.

[47] S. Hu, Y. Pei, P. P. Liang, and Y.-C. Liang, "Deep neural network for robust modulation classification under uncertain noise conditions," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 564–577, Jan. 2020.

[48] M. Borgerding, P. Schniter, and S. Rangan, "AMP-inspired deep networks for sparse linear inverse problems," *IEEE Trans. Signal Process.*, vol. 65, no. 16, pp. 4293–4308, Aug. 2017.

[49] Z. Zhang, Y. Xu, J. Yang, X. Li, and D. Zhang, "A survey of sparse representation: Algorithms and applications," *IEEE Access*, vol. 3, pp. 490–530, May 2015.

[50] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*. [Online]. Available: http://arxiv.org/abs/1503.02531

[51] W. E. Cobb, E. D. Laspe, R. O. Baldwin, M. A. Temple, and Y. C. Kim, "Intrinsic physical-layer authentication of integrated circuits," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 1, pp. 14–24, Feb. 2012.

Yu Wang (Graduate Student Member, IEEE) received the B.S. degree in communication engineering from the Nanjing University of Posts and Telecommunications (NJUPT), Nanjing, China, in 2018, where he is currently pursuing the Ph.D. degree. He has published more than 20 IEEE journal/conference papers. His research interests include deep learning, distributed learning, cognitive radio, signal identification, and physical layer security. He received several best paper awards, such as ICEICT 2019, CSPS 2019, and CSPS 2018.

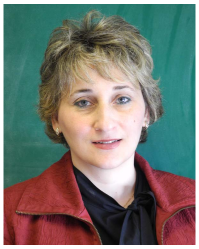
Guan Gui (Senior Member, IEEE) received the Ph.D. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2012.

From 2009 to 2014, he joined Tohoku University as a Research Assistant and a Post-Doctoral Research Fellow, respectively. From 2014 to 2015, he was an Assistant Professor with Akita Prefectural University. Since 2015, he has been a Professor with the Nanjing University of Posts and Telecommunications, Nanjing, China. He has published more than 200 IEEE journal/conference papers and won several best paper awards, such as ICC 2017, ICC 2014, and VTC 2014-Spring. His recent research interests include artificial intelligence, deep learning, non-orthogonal multiple access, wireless power transfer, and physical layer security. He received the IEEE Communications Society Heinrich Hertz Award in 2021, the Elsevier Highly Cited Chinese Researchers in 2020, the Member and Global Activities Contributions Award in 2018, the Top Editor Award of IEEE Transactions on Vehicular Technology in 2019, the Outstanding Journal Service Award of *KSII Transactions on Internet and Information System* in 2020, and the Exemplary Reviewer Award of IEEE Communications Letters in 2017. He was selected as the Jiangsu Specially-Appointed Professor in 2016, the Jiangsu High-level Innovation and Entrepreneurial Talent in 2016, the Jiangsu Six Top Talent in 2018, and the Nanjing Youth Award in 2018. He is serving or served on the editorial boards of several journals, including IEEE Transactions on Vehicular Technology, *IEICE Transactions on Communications*, *Physical Communication*, *Wireless Networks*, IEEE Access, *Journal of Circuits, Systems and Computers*, *Security and Communication Networks*, *IEICE Communications Express*, *KSII Transactions on Internet and Information Systems*, and *Journal of Communication*. In addition, he served as the IEEE VTS Ad Hoc Committee Member in AI Wireless, the Executive Chair for IEEE VTC 2021-Fall, the Vice Chair for IEEE WCNC 2021, the TPC Chair for PHM 2021, the General Co-Chair for Mobimedia 2020, the TPC Chair for WiMob 2020, the Track Chair for IEEE VTC 2020-Spring, ISNCC 2020, and ICCC 2020, and the Award Chair for IEEE PIMRC 2019. He served as a TPC member for many IEEE international conferences, including GLOBECOM, ICC, WCNC, PIRMC, VTC, and SPAWC.

Haris Gacanin (Fellow, IEEE) received the Dipl.-Ing. degree in electrical engineering from the University of Sarajevo in 2000, and the M.Sc. and Ph.D. degrees from Tohoku University, Japan, in 2005 and 2008, respectively.

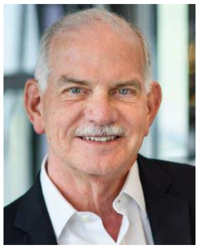
From 2008 to 2010, he was with Tohoku University first as a Japan Society for the Promotion of Science (JSPS) Post-Doctoral Fellow and later as an Assistant Professor. In 2010, he joined Alcatel-Lucent Bell (now Nokia Bell) as a Physical-layer Expert and later moved to Nokia Bell Labs as the Department Head. Since April 2020, he has been with RWTH Aachen University. He is currently the Head of the Chair for Distributed Signal Processing and the Co-Director of the Institute for Communication Technologies and Embedded Systems. He has more than 200 scientific publications (journals, conferences, and patents) and invited/tutorial talks. His professional interests are related to broad areas of digital signal processing and artificial intelligence with applications in wireless communications. He acted as the general chair and a technical program committee member of various IEEE conferences. He was a recipient of several Nokia innovation awards, including the IEICE Communications Society Best Paper Award in 2021, the IEICE Communication System Study Group Best Paper Award (joint 2014, 2015, and 2017), the 2013 Alcatel-Lucent Award of Excellence, the 2012 KDDI Foundation Research Award, the 2009 KDDI Foundation Research Grant Award, the 2008 JSPS Postdoctoral Fellowships for Foreign Researchers, the 2005 Active Research Award in Radio Communications, the 2005 Vehicular Technology Conference (VTC 2005-Fall) Student Paper Award from IEEE VTS Japan Chapter, and the 2004 Institute of IEICE Society Young Researcher Award. He served as an Editor for *IEICE Transactions on Communications* and *IET Communications*. He served as an Associate Editor for *IEEE Communications Magazine*. He was a Distinguished Lecturer of the IEEE Vehicular Technology Society.

Tomoaki Ohtsuki (Senior Member, IEEE) received the B.E., M.E., and Ph.D. degrees in electrical engineering from Keio University, Yokohama, Japan, in 1990, 1992, and 1994, respectively. From 1994 to 1995, he was a Post-Doctoral Fellow and a Visiting Researcher in electrical engineering at Keio University. From 1993 to 1995, he was a Special Researcher of Fellowships of the Japan Society for the Promotion of Science for Japanese Junior Scientists. From 1995 to 2005, he was with the Science University of Tokyo. From 1998 to 1999, he was with the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley. In 2005, he joined Keio University. He is currently a Professor with Keio University. He is engaged in research on wireless communications, optical communications, signal processing, and information theory. He has published more than 205 journal articles and 415 international conference papers. He is a fellow of the IEICE and a member of the Engineering Academy of Japan. He was a recipient of the 1997 Inoue Research Award for Young Scientist, the 1997 Hiroshi Ando Memorial Young Engineering Award, the Ericsson Young Scientist Award 2000, the 2002 Funai Information and Science Award for Young Scientist, IEEE the 1st Asia-Pacific Young Researcher Award 2001, the 5th International Communication Foundation (ICF) Research Award, the 2011 IEEE SPCE Outstanding Service Award, the 27th TELECOM System Technology Award, the ETRI Journal's 2012 Best Reviewer Award, and the 9th International Conference on Communications and Networking in China 2014 (CHINACOM '14) Best Paper Award. He served as the Chair for the IEEE Communications Society, Signal Processing for Communications and Electronics Technical Committee. He has served as the General-Co Chair, the Symposium Co-Chair, and the TPC Co-Chair for many conferences, including IEEE GLOBECOM 2008, SPC, IEEE ICC 2011, CTS, IEEE GLOBECOM 2012, SPC, IEEE ICC 2020, SPC, IEEE APWCS, IEEE SPAWC, and IEEE VTC. He gave tutorials and keynote speeches at many international conferences, including IEEE VTC, IEEE PIMRC, and IEEE WCNC. He was a Vice President and a President of the Communications Society of the IEICE. He served as a Technical Editor for the *IEEE Wireless Communications Magazine* and an Editor for *Physical Communications* (Elsevier). He is currently serving as an Area Editor for the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY and an Editor for the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS. He is a Distinguished Lecturer of the IEEE.



Octavia A. Dobre (Fellow, IEEE) received the Dipl.Ing. and Ph.D. degrees from the Polytechnic Institute of Bucharest, Romania, in 1991 and 2000, respectively.

From 2002 to 2005, she was with the New Jersey Institute of Technology, USA. In 2005, she joined Memorial University, Canada, where she is currently a Professor and the Research Chair. She was a Visiting Professor with the Massachusetts Institute of Technology, USA, and the Universit de Bretagne Occidentale, France. Her research interests encompass various wireless technologies, such as non-orthogonal multiple access and full duplex, optical and underwater communications, and machine learning for communications. She has (co)authored over 350 refereed articles in these areas. She is a fellow of the Engineering Institute of Canada. She obtained the best paper awards at various conferences, including IEEE ICC, IEEE GLOBECOM, IEEE WCNC, and IEEE PIMRC. She served as the general chair, the technical program co-chair, the tutorial co-chair, and the technical co-chair for symposia at numerous conferences. She serves as the Editor-in-Chief (EiC) for the IEEE Open Journal of the Communications Society. She was the EiC of the IEEE COMMUNICATIONS LETTERS, a senior editor, an editor, and a guest editor for various prestigious journals and magazines. She was a Fulbright Scholar, a Royal Society Scholar, and a Distinguished Lecturer of the IEEE Communications Society.



H. Vincent Poor (Life Fellow, IEEE) received the Ph.D. degree in EECS from Princeton University in 1977, and the D.Eng. degree (Hons.) from the University of Waterloo in 2019.

From 1977 to 1990, he was with the Faculty of the University of Illinois at Urbana-Champaign. Since 1990, he has been with the Faculty at Princeton, where he is currently the Michael Henry Strater University Professor. From 2006 to 2016, he served as the Dean of the Princeton's School of Engineering and Applied Science. He has held visiting appointments at several other universities, including most recently at Berkeley and Cambridge. His research interests are in the areas of information theory, machine learning and network science, and their applications in wireless networks, energy systems, and related fields. Among his publications in these areas is the forthcoming book *Machine Learning and Wireless Communications* (Cambridge University Press, 2021). He is a member of the National Academy of Engineering and the National Academy of Sciences. He is a Foreign Member of the Chinese Academy of Sciences, the Royal Society, and other national and international academies. Recent recognition of his work includes the 2017 IEEE Alexander Graham Bell Medal.