Modulation Classification Based on Signal Constellation Diagrams and Deep Learning

Shengliang Peng, *Member, IEEE*, Hanyu Jiang, *Student Member, IEEE*, Huaxia Wang, *Student Member, IEEE*, Hathal Alwageed, *Student Member, IEEE*, Yu Zhou, *Student Member, IEEE*, Marjan Mazrouei Sebdani, and Yu-Dong Yao, *Fellow, IEEE*

***Abstract*— Deep learning (DL) is a new machine learning (ML) methodology that has found successful implementations in many application domains. However, its usage in communications systems has not been well explored. This paper investigates the use of the DL in modulation classification, which is a major task in many communications systems. The DL relies on a massive amount of data and, for research and applications, this can be easily available in communications systems. Furthermore, unlike the ML, the DL has the advantage of not requiring manual feature selections, which significantly reduces the task complexity in modulation classification. In this paper, we use two convolutional neural network (CNN)-based DL models, AlexNet and GoogLeNet. Specifically, we develop several methods to represent modulated signals in data formats with gridlike topologies for the CNN. The impacts of representation on classification performance are also analyzed. In addition, comparisons with traditional cumulant and ML-based algorithms are presented. Experimental results demonstrate the significant performance advantage and application feasibility of the DL-based approach for modulation classification.**

***Index Terms*— Convolutional neural network (CNN), data conversion, deep learning (DL), modulation classification.**

I. Introduction

DEEP learning (DL) is a branch of machine learning (ML) that uses numerous successive layers of nonlinear processing units to model high-level abstractions in data. In recent years, this technique has gained a great popularity due to its state-of-the-art capability for the big data analysis [1]. As one of the most powerful identification/classification tools, it has been applied in various application fields, such as computer vision [2], natural language processing [3], economics [4], and bioinformatics [5]. In the competition of ImageNet large scale visual recognition challenge (ILSVRC), many research teams submitted DL algorithms for large-scale object detection and image classification, and the top-5 accuracy of identifying objects from 1000 categories exceeds 95% [2]. In [3], Google announced the Google neural machine translation system, which utilizes an eight-encoder–eight-decoder DL network and approaches the accuracy achieved by average bilingual human translators on some of the test sets.

Although the DL is widely investigated in many application domains, its usage in communications systems has not been well explored. For modulation classification using the DL, there have been some reported work, including [6]–[12]. The focus of this paper is to develop methods to represent modulated signals in data formats for a convolutional neural network (CNN). It is noticed that, in the ML, a support vector machine (SVM) and a K-nearest neighbor (KNN) have been applied for medium access control protocol identification [13] and modulation classification [14]. This paper studies the use of DL in communications systems, specifically, for modulation classification. There are several advantages of using the DL in communications system applications. First, a large amount of data are required for the DL and, in communications systems, massive data can be easily obtained due to the large number of communications devices and the high transmission data rates. Second, manual feature selections, which can be a significant challenge in communications systems or modulation classification, are not needed in the DL. Third, with the rapid and significant advancement of DL technologies, considerable potentials can be found for many complex applications in communications systems.

The major concern for using the DL in communications systems might be about high computational complexity. Note that the DL usually consists of two phases: training and testing. In the training phase, massive data are fed to DL networks to produce a trained model, and this phase always runs billions of computations and costs dozens of hours. In the testing phase, one example of data is input into the network to infer a result, and this phase can be completed in several milliseconds with limited computational resource. Fortunately, when implementing the DL in real communications systems, only the testing phase is frequently applied. The training phase

Manuscript received July 21, 2017; revised April 21, 2018 and June 14, 2018; accepted June 20, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 61362018, in part by the National Natural Science Foundation of China under Grant 61771121, in part by the China Scholarship Council under Grant 201508350023, and in part by Huaqiao University under Grant 13BS101. *(Corresponding author: Yu-Dong Yao.)*

S. Peng was with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030 USA. He is now with the College of Information Science and Technology, Huaqiao University, Xiamen 361021, China.

H. Jiang, H. Alwageed, Y. Zhou, and Y.-D. Yao are with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030 USA (e-mail: yyao@stevens.edu).

H. Wang is with FutureWei Technologies, Inc., Bridgewater, NJ 08807 USA.

M. M. Sebdani is with Publicis Media, New York, NY 10014 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2018.2850703

is usually conducted beforehand and does not incur too much computational burden.

Modulation classification is a major issue in many communications systems with both civilian and military applications, such as spectrum management, interference identification, electronic warfare, and threat analysis. Previously, this issue has been handled through traditional signal processing [15], [16], the artificial neural network (ANN) [17]–[19], or ML [14] methods. In [15], the fourth-order cumulants are suggested for modulation classification. In [16], two general classes of modulation classification techniques, including the likelihood-based approach and the feature-based approach, are summarized. Nandi and Azzouz [17] design an ANN-based modulation recognition algorithm consisting of three networks, which contain 3, 0, and 0 hidden layers, respectively. In [18], an ANN with one hidden layer is selected. In [19], MAXNET, whose feed forward network has two hidden layers with five neurons in each layer, is utilized for modulation classification. Note that these ANNs have fewer hidden layers and are different from DL networks. The use of KNN for modulation classification is discussed in [14]. Recently, the use of the DL has been investigated for this issue [6]–[8], [10], [11]. In [6], the DL algorithm is suggested to realize modulation recognition-based signal discriminations, including full temporal characteristics, frequency spectrum, and several higher order spectral characteristics. In [7], [9], and [12], adaptation of DL to the complex temporal signal domain is studied. We show the possibility of modulation classification using DL and constellation information in [8]. A nonnegativity constraint is introduced into a DL autoencoder network for modulation classification in [10]. In [11], graphic constellations and deep belief network are proposed to recognize modulation schemes.

This paper addresses the topic of modulation classification using the DL and we have the following contributions. First, we propose methods to convert complex signals into data formats with a gridlike topology, e.g., images, which facilitates the use of prevalent DL network models and frameworks for classification. Second, we analyze the impacts of convention parameters (e.g., the size of selected area of complex plane and image resolution) on the classification performance. Third, we develop and configure DL models based on CNNs for modulation classification, which achieves the performance significantly improved compared with other classification algorithms. Finally, we illustrate computational complexity of proposed approaches and demonstrate the application feasibility of DL in real communications systems.

The rest of this paper is organized as follows. We first give an overview of the architectures and models of DL. We then develop methods converting modulated signals into images for DL. We also present steps of using CNNs for modulation classification. Finally, experimental results are given to demonstrate the performance advantage of this paper.

II. DL Architectures and Models

Building a "deep" neural network that consists of multiple layers and multiple units, each layer for abstracting massive amount of data is the main idea of DL, which is inspired by the architectural depth of the brain and the mathematical model of a biological neuron. Technically, a deep neural network aims at learning different hierarchies of features autonomously from data: extract low-level features from raw inputs and higher level features based on feature representations from the previous level. In other words, it is designed to recognize patterns by itself without providing manual feature representations to machine, which is so-called data-driven feature extraction [5]. In traditional ML approaches, detection and extraction of effective features are time-consuming and need an extensive domain expertise, and they always suffer from the growing complexity of real-world problems. In contrast, deep neural networks can assemble simple features to abstract more complex features autonomously through multiple nonlinear transformations that may contain billions of weight parameters. The procedure of training the deep neural network is gradually optimizing those parameters to approach the optimal hierarchical representation learned from data. In general, the DL is good at handling the tasks of mapping input vectors to output vectors, which could be done by people or experts in their specific fields, given well-designed DL models and a sufficiently large amount of labeled data for training [1].

With the rapid development and advancement in recent years, various architectures of DL have been proposed and applied in different fields with state-of-the-art performance, such as CNNs, recurrent neural networks, and restricted Boltzmann machines [1]. In this paper, the CNN is studied and applied in communications systems. In particular, two popular CNN models, AlexNet [20] and GoogLeNet [21], are investigated for performance evaluation.

A. Convolutional Neural Networks

CNNs, derived from the feedforward neural networks and also known as convolutional networks [22], have been extensively studied for practical applications (i.e., image processing and speech recognition). Reference [1] provides the definition of CNNs as follows.

Convolutional networks are simply neural networks that use convolution in the place of general matrix multiplication in at least one of their layers.

As shown in Fig. 1(a), there are three types of layers in typical CNN architectures: convolutional layer, pooling layer, and fully connected layer. The convolutional layer is the essential part of CNNs that takes most of the calculation. There are two important characteristics of the convolutional layer that improve the entire ML system.

1) *Sparse Connectivity*: Unlike the traditional neural network in which every input unit interacts with every output unit, CNNs can achieve sparse connectivity between input and output units by making the kernel size smaller than the input size, which improves statistical efficiency and reduces memory requirements.
2) *Parameter Sharing*: In CNNs, the same parameter could be used multiple times by different neurons. Thus, the storage requirements of the model can be significantly reduced.

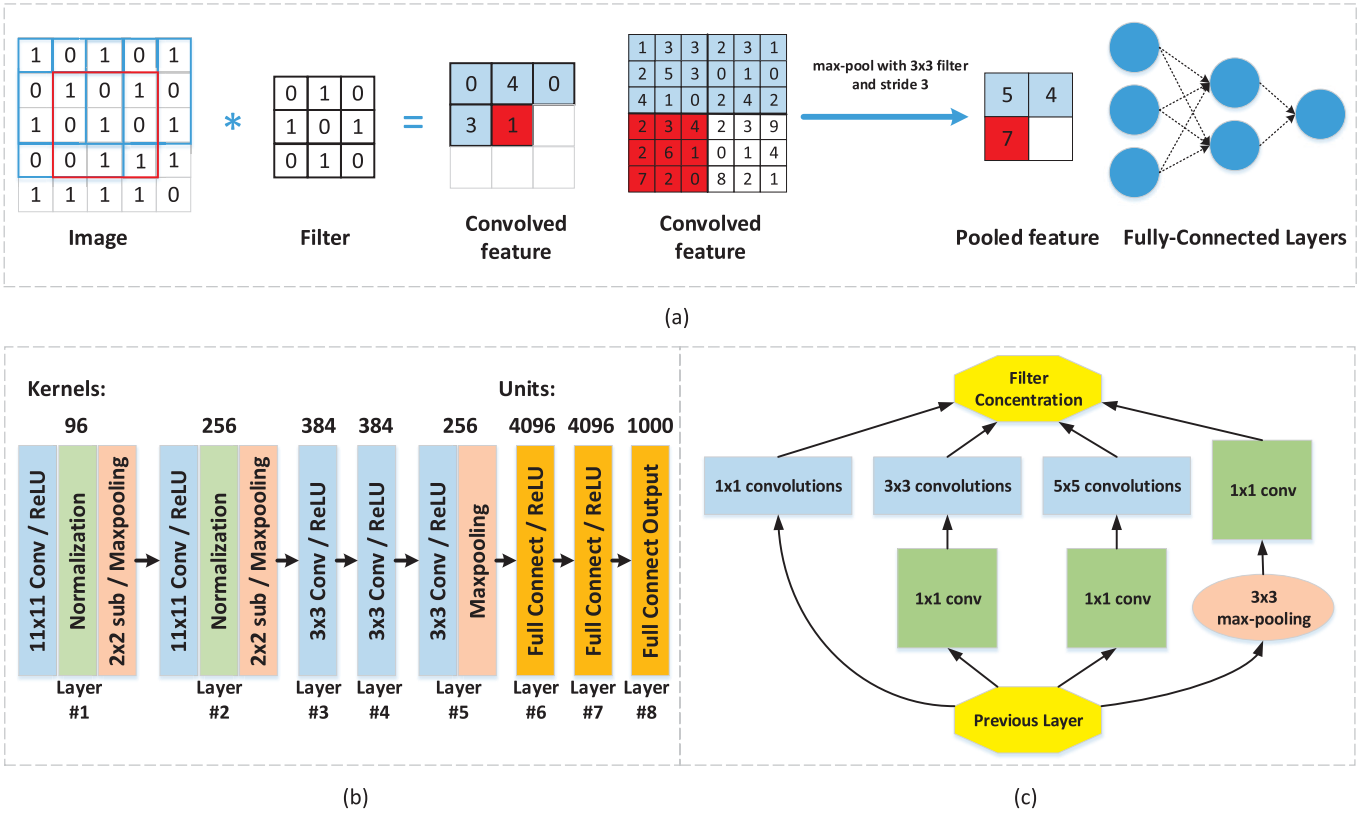


Fig. 1. CNN and two typical CNN-based DL models. (a) Three types of layers in CNN. (b) AlexNet. (c) Inception module of GoogleNet.

The pooling layer replaces the output at certain location with a relevant statistic of the nearby outputs. One of the popular pooling operations is max pooling [23], which returns the maximum output value within a rectangle grid. After pooling operation, the output representation becomes invariant to the small translations of inputs, which significantly improves the network statistic efficiency. The fully connected layer in CNNs is exactly the same as in traditional neural networks where neurons have full connections to all activations in the previous layer.

B. AlexNet

As the landmark model for ImageNet classification and the winner of ILSVRC'12 [20], AlexNet made a huge contribution in popularizing the CNN in computer vision with a large and deep architecture. AlexNet consists of five convolutional layers and three fully connected layers with a 1000-way softmax layer.

In order to improve the performance and reduce training time as well, AlexNet merged several novel features into the network. first of all, instead of using the traditional neuron activation function, $f(x) = \tanh(x)$, a neuron with nonsaturating nonlinearity was introduced as rectified linear units (ReLUs): $f(x) = \max(0, x)$, with a faster training procedure. As shown in Fig. 1(b), ReLUs are attached to every convolutional and fully connected layers. Another pioneering contribution of this paper involves effective techniques for reducing overfitting effects that traditional ML methods suffered, which also enabled the whole architecture of AlexNet to be feasible in practice. Due to a large amount of parameters and neurons involved (about 60 000 000 parameters and 650 000 neurons) but limited amount of labeled images for most applications, data augmentation through extracting random patches and altering the intensities of RGB channels were adopted to enlarge the original image data set. In addition, a regularization method, "dropout" [20], was also employed to set the outputs of hidden neurons to zero with a probability of 0.5 to learn more general features. The number of training iterations for convergence was roughly doubled when adding random "dropout" mechanism onto the first two fully connected layers. However, it is still more efficient than ensemble prediction approaches that require multiple pretrained models to improve prediction accuracy.

C. GoogLeNet

GoogLeNet, a 22-layer CNN and the winner of ILSVRC'14 [21], is an efficient deep neural network model designed for computer vision. When other CNN-based models are trapped in serious negative effects brought by size growth, GoogLeNet is able to significantly increase the depth and width of the network for improving performance while keeping the demand of computational workload at a reasonable level. The key reason for such a success is the introduction of a bunch of *inception* modules that are usually stacked on the higher layers. The structure of an inception module is shown in Fig. 1(c), where 1×1 convolutions are added before expensive 3×3 and 5×5 convolution as the dimension reduction modules. This approach significantly

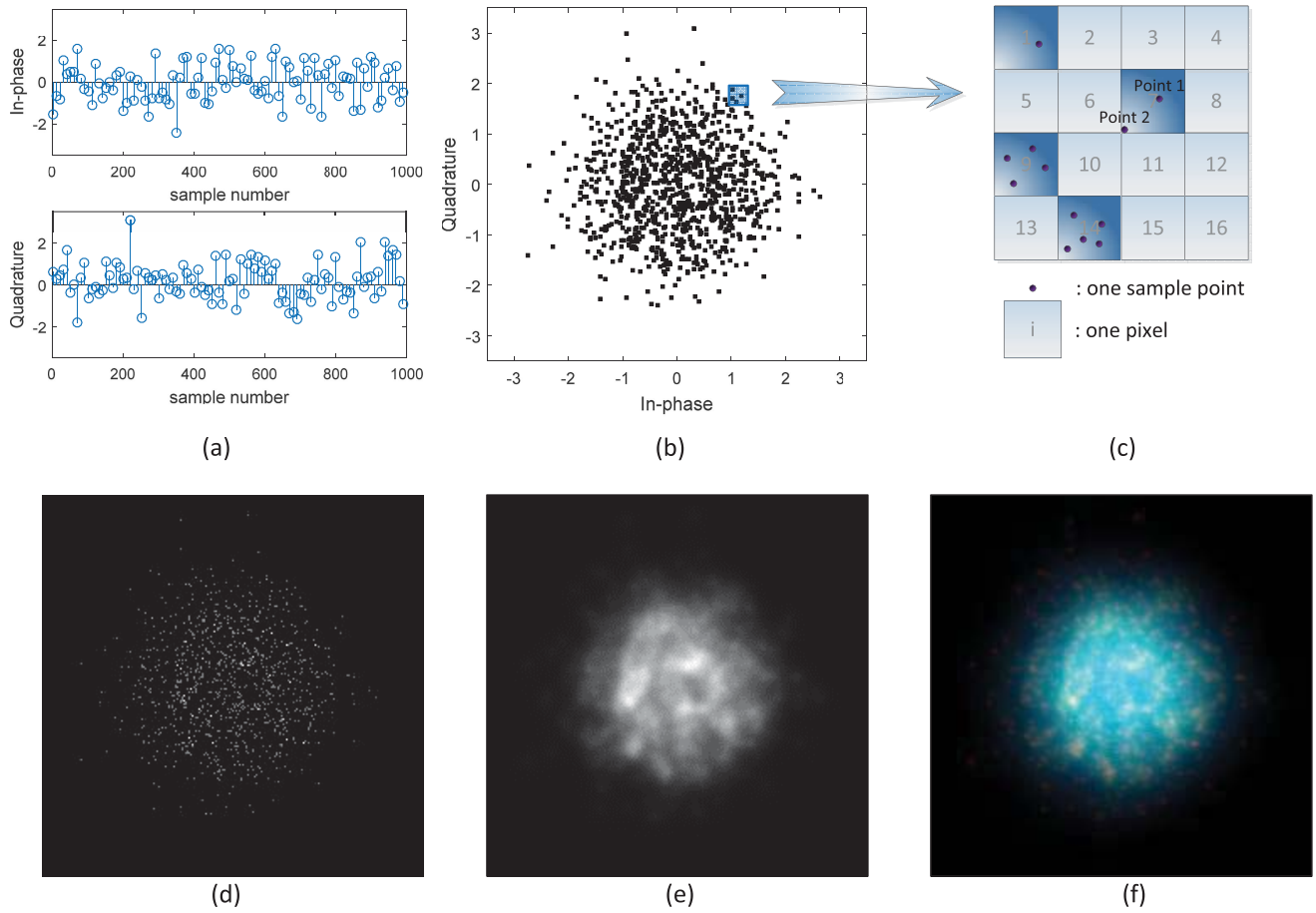


Fig. 2. Complex samples and different types of images for 8PSK signal at SNR = 2 dB. (a) Complex signal samples. (b) Constellation diagram. (c) Sample points and pixels. (d) Gray image. (e) Enhanced gray image. (f) Three-channel image.

reduces the total number of parameters in the network (GoogLeNet with 7 000 000 parameters versus AlexNet with 60 000 000 parameters). The initial purpose of designing this inception module is to approximate a local sparse structure of CNN by using readily available dense components (matrices) which can make efficient use of computational resources in current computing infrastructures. Moreover, in GoogLeNet, considering backpropagate gradients, extra auxiliary classifiers are added onto intermediate layers, which is expected to reduce the effects of vanishing gradient [21] while providing regularization.

III. Modulation Data Conversion for DL

This paper studies the use of DL in communications systems, specifically, the task of modulation classification. Similar to [14] and [15], it is assumed that the communications system operates in a coherent and synchronous scenario with single-tone signaling. It is also assumed that carrier, timing, and waveform recovery have been accomplished. We then obtain a baseband sequence of complex samples with additive white Gaussian noise, as shown in Fig. 2(a). Given a batch of samples, our task is to determine the modulation format of the received signal. Eight possible modulation formats/categories are considered, including binary phase-shift keying (BPSK), four amplitude-shift keying (4ASK), quadrature phase-shift keying (QPSK), offset QPSK (OQPSK), eight phase-shift keying (8PSK), 16 quadrature amplitude modulation (16QAM), 32 quadrature amplitude modulation (32QAM), and 64 quadrature amplitude modulation (64QAM).

As mentioned above, most existing CNN models, including AlexNet and GoogLeNet, are developed for image recognition/classification. However, in modulation classification, the data to be processed are not images but complex data samples. In order to utilize the existing DL models, we can convert complex data samples into images. Since a data conversion process usually introduces information loss, our research and design aim to preserve the original information as much as possible.

A. Constellation Diagram

A constellation diagram has been widely used as a 2-D representation of a modulated signal by mapping signal samples into scattering points on a complex plane. Note that the complex plane extends infinitely, while the area an image can depict is limited. We have to select part of the complex plane to generate a constellation diagram image. If the selected area is too small, some signal samples may be excluded from the image and abandoned due to severe noise levels. On the

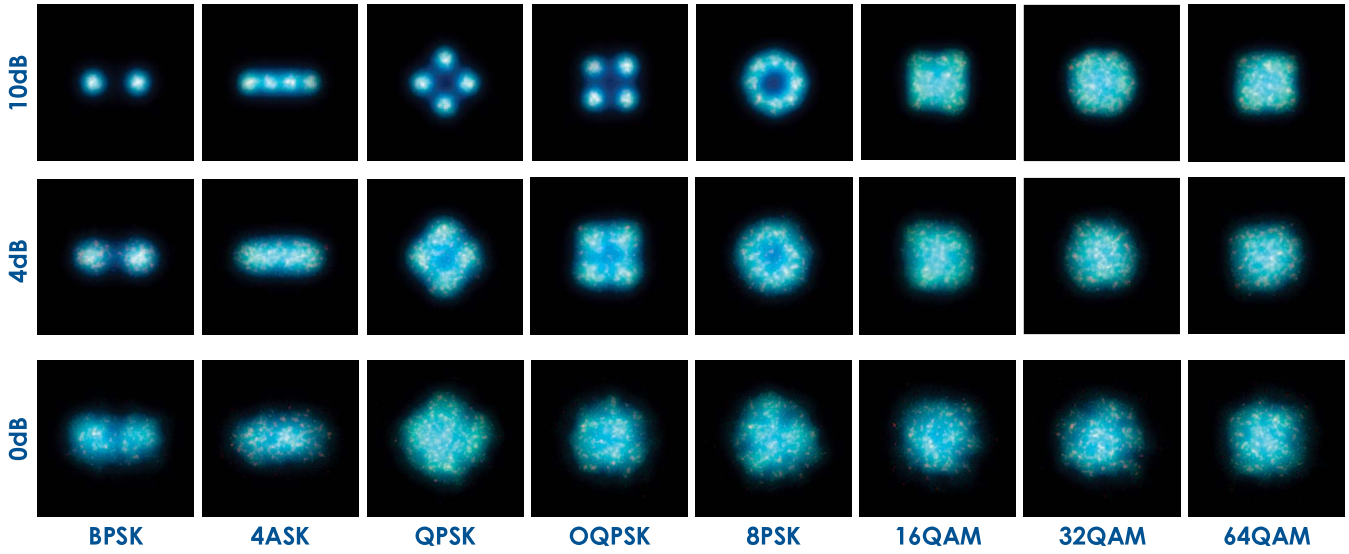


Fig. 3. Three-channel images for eight modulation categories with different SNRs.

contrary, if the area is too large, signal samples may congregate in a small region and probably overlap. Unless otherwise specified, this paper selects a 7×7 complex plane, assuming a typical signal-to-noise ratio (SNR) range from 0 to 10 dB. Fig. 2(b) gives an example of the constellation diagram image, which is generated from 1000 samples of 8PSK modulated signal with the SNR at 2 dB.

B. Gray Image

If the pixel density is sufficiently large, a constellation diagram image can be regarded as a full representation of signal samples. In this case, each sample can be represented by one or more pixels. However, due to the limited pixel density, there might be multiple samples within one pixel, as shown in Fig. 2(c). Note that a constellation diagram is a binary image, in which pixels with one or more sample points are treated/represented similarly. To consider the impact of multiple sample points, we represent the number of samples within each pixel as an intensity value (e.g., the intensity values of Pixels 1, 2, and 14 are 1, 0, and 5). The binary constellation diagram can then be converted into a gray image, as depicted in Fig. 2(d).

C. Enhanced Gray Image

In a gray image, the number of samples in each pixel is considered. However, there are still two limitations. First, the location of each sample within a pixel is not considered. Second, the impact of each sample in a pixel on its neighboring pixels is ignored. Therefore, we introduce an enhanced gray image, which takes into consideration the distances between sample points and centroids of pixels. Furthermore, it adopts an exponential decay model, $B_{i,j} = P \cdot e^{-\lambda \cdot d_{i,j}}$, where $B_{i,j}$ represents the impact of a sample point #i on Pixel #j, P is the power of each sample point, $d_{i,j}$ is the distance between sample point #i and the centroid of Pixel j, and λ is an exponential decay rate. We will then combine the impacts of all data samples on each pixel. These results can be utilized as the intensity values to generate an enhanced gray image, as shown in Fig. 2(e).

D. Three-Channel Image

CNN-based models are usually designed to deal with color images, in which there are three data processing channels (red, green, and blue). According to the enhanced gray image approach, a single data processing channel is used. Therefore, we introduce an idea of three-channel image, which comprises three enhanced gray images derived from the same set of data samples with three different exponential decay rates, as shown in Fig. 2(f).

In the following, we will feed modified AlexNet and GoogLeNet models with three-channel images to implement modulation classification. Each image is generated based on 1000 samples of a modulated signal. Examples of three-channel images for eight modulation formats with different SNRs are shown in Fig. 3.

IV. Modulation Classification Using DL

In order to build CNN-based models for modulation classification, Caffe [24], a widely used DL framework, is utilized. Due to its strong modularity with Python/MATLAB support, we are able to design and evaluate our models efficiently. Two built-in models, *Berkeley Vision and Learning Center (BVLC) Reference CaffeNet* (a minor variation of AlexNet) and *BVLC GoogLeNet*, provided within Caffe toolkit are investigated.

A. AlexNet Configurations

We slightly modify two layers of the *BLVC reference CaffeNet* while keeping the rest six layers unchanged. The number of output in layer 8 is changed from the default of 1000 to 8 which is matched to the number of modulation categories investigated in our case. The number of neurons in

TABLE I
CLASSIFICATION RESULTS OF FOUR IMAGE TYPES FOR 8PSK WITH SNR = 2 dB AND ALEXNET

IMAGE TYPES	BPSK	4ASK	QPSK	OQPSK	8PSK	16QAM	32QAM	64QAM	ACCURACY
Constellation Diagram	0	0	5	5	828	22	140	0	82.8%
Gray Image	0	0	6	8	888	16	80	2	88.8%
Enhanced Gray Image	0	0	2	3	965	1	29	0	96.5%
3-Channel Image	0	0	0	1	971	5	23	0	97.1%

layer 7 is shrinked to 512, since the original 4096 neurons always lead to difficulty in convergence during our model training. In addition to the changes of model architecture, several parameters of solver configuration are also adjusted for a better classification performance as well as a higher training speed, such as learning rate (0.01 → 0.001), *gamma* (0.1 → 0.5), and *stepsize* (100 000 → 10 000). Unless otherwise specified, constellation images with 227×227 pixels are preferred. Image resolution cannot be lower than 227×227 due to the structure of AlexNet. If it is higher than 227×227, random cropping [24] is enabled. Moreover, considering the tradeoff between available computing resources on graphics processing unit (GPU) and training efficiency, we use a *mini-batch* gradient descent with a batch size of 64 in the input layer.

B. GoogLeNet Configurations

Several places have been changed from the default configuration of *BVLC GoogLeNet*. The number of outputs for all the three classifiers, including two auxiliary classifiers located at the top of inception modules [21], are changed from 1000 to 8 modulation categories. The default image resolution is 224×224. Random cropping is enabled if resolution is higher than 224×224. According to our experiments of tuning parameters, changing the learning rate (0.01 → 0.001) and *gamma* (0.96 → 0.8) yields a better performance and efficiency.

C. Implementation

Both AlexNet and GoogLeNet are trained as follows: 1) baseband complex samples of modulated signals are obtained either from real receivers or computer simulations; 2) each block of 1000 samples is utilized to generate a constellation image; 3) each image is labeled according to the modulation category of samples; 4) 100 000 and 1000 labeled images per modulation category are collected to form training and validation data sets, respectively; 5) both data sets are fed to DL networks for training with Caffe; and 6) after a maximum of 200 000 training iterations, some trained models with suffix *".caffemodel"* can be obtained. In this paper, training is conducted on our computing server with a Nvidia Tesla K40m GPU, and the training time of AlexNet and GoogLeNet is about 12.5 and 24.3 h, respectively.

The task of exploiting trained DL models to identify modulation categories is regarded as testing or inference. In this task, the received signal is first sampled. Then, 1000 samples of received signal are collected to generate a constellation image. After that, the constellation image and one trained model are fed to DL networks for calculation. Calculation result is a probability vector that indicates all possible modulation categories of received signal.

V. EXPERIMENTS

A. Effect of Data Conversion Methods

In Section III, we have discussed methods to convert complex signal samples into images without noticeable information loss. Here, we present some experimental results to show the impact of data conversion using different methods. In our experiments, the same set of complex samples is used to generate four types of images, including a constellation diagram, a gray image, an enhanced gray image, and a three-channel image. For each type, images are fed to AlexNet for training, which results in a trained network. For each network, 1000 tests of identifying 8PSK modulated signals with SNR = 2 dB are implemented to evaluate its classification performance. Table I records the test results of four image types. As shown in the table, the classification accuracy improves from 82.8% to 97.1% if the three-channel image is utilized instead of the constellation diagram. Moreover, it is seen that 8PSK is more likely confused with 32QAM in our experiments.

B. Impact of Size of Selected Area

Since the selected area of complex plane to generate constellation images should be neither too small nor too large, we discuss its impact on the classification accuracy in this section. Different sizes, including 3×3, 5×5, 7×7, 9×9, and 11×11, are considered. Three-channel images and AlexNet are adopted for training and testing. 8000 tests (1000 tests per modulation category) are conducted to evaluate classification accuracy. Fig. 4 plots classification accuracy versus the size of selected area under different scenarios of SNR = 0, 4, and 10 dB. According to this figure, as the size of selected area increases, the classification accuracy first increases and then slightly decreases, which is consistent with our assumptions. Note that classification accuracy reaches peak when size is 7×7, and thus, it is the best choice for our scenarios. Moreover, as shown in Fig. 4, the size of selected area causes a larger impact if the SNR is lower. When SNR = 10 dB, classification accuracy is hardly affected as size changes.

C. Impact of Image Resolution

Besides the size of selected area, resolution is another key parameter of constellation images. As introduced in

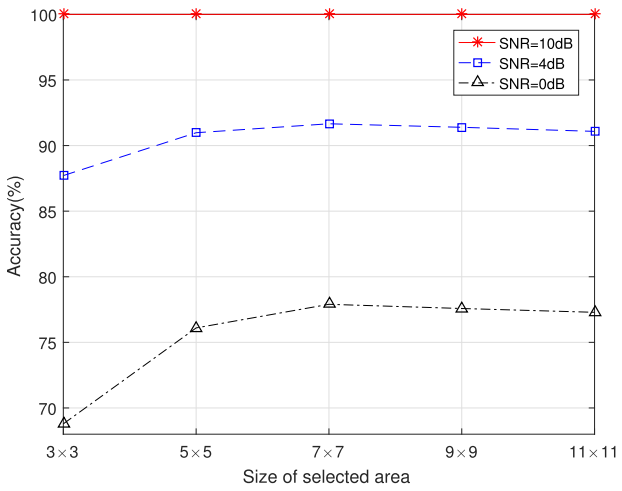


Fig. 4. Classification accuracy versus size of selected area under different SNRs with AlexNet.

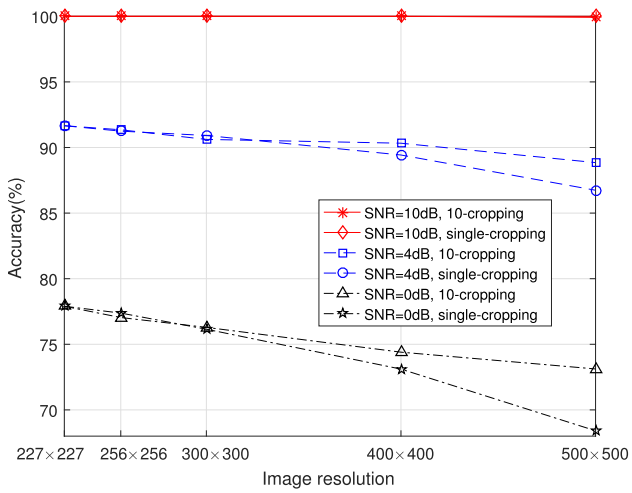


Fig. 5. Classification accuracy versus image resolution under different SNRs with AlexNet.

Section IV, the minimal image resolution for AlexNet is 227×227. If the resolution is higher than 227×227, in the training phase, random cropping that randomly crops different images from larger images is suggested to enrich the training data set. In the testing phase, either single-cropping or 10-cropping can be adopted. According to single-cropping, a 227×227 image is cropped from the center of a larger image. According to 10-cropping, 10 images are cropped from the center and four corners of a larger image and its mirror.

Fig. 5 demonstrates the impact of image resolution on the classification accuracy. Five resolution values of 227×227, 256×256, 300×300, 400×400, and 500×500 are considered under the scenarios of SNR $= 0$, 4, and 10 dB. The classification accuracy is evaluated based on 1000 tests per modulation category. As shown in this figure, higher image resolution does not incur a better classification accuracy. This phenomenon can be explained as follows. If the image resolution is higher, cropping is implemented to obtain a 227×227 image. According to the Caffe framework, random cropping is used in the training phase, while single-cropping/10-cropping is used in the testing phase. The mismatch of cropping methods impairs the accuracy of modulation classification. Obviously, the best resolution for AlexNet is 227×227, and cropping should be avoided. In addition, 10-cropping performs better than single-cropping. The reason lies in the fact that 10-cropping crops 10 times of images for testing.

D. Classification Performance

Classification performance results of eight modulation categories are presented using a confusion matrix in Table II, which is obtained based on three-channel images and GoogLeNet with SNR $=$ 4 dB. In each modulation category, 1000 tests are implemented. It is clear from Table II that classification/identification of low-order modulation is relatively easy. The classification accuracy of BPSK, 4ASK, QPSK, OQPSK, and 8PSK is 100%. When the modulation order becomes higher (16QAM, 32QAM, and 64QAM), the classification performance becomes worse, especially for 16QAM and 32QAM. Notice that, as shown in Fig. 3, 16QAM and 32QAM have similar square constellation diagrams.

E. Accuracy Comparisons

For accuracy comparison, we consider five different modulation classification algorithms.

1) *Cumulant*: A traditional signal processing algorithm using the fourth-order cumulant $\widetilde{C}_{40}$ as the classification statistics [15].
2) *SVM-7*: An ML-based algorithm using the SVM with seven features, including three fourth-order cumulants C_{40}, C_{41}, and C_{42} and four sixth-order cumulants C_{60}, C_{61}, C_{62}, and C_{63} [14].
3) *SVM-5*: An ML-based algorithm using the SVM with five features, including two second-order cumulants C_{20} and C_{21} and three fourth-order cumulants C_{40}, C_{41}, and C_{42}.
4) *AlexNet*: A CNN-based algorithm using the AlexNet model.
5) *GoogLeNet*: A CNN-based algorithm using the GoogLeNet model.

Fig. 6 presents the average classification accuracy of five algorithms versus SNR. The average accuracy is obtained by averaging the classification performance of eight modulation categories. For each modulation category, 1000 tests are implemented. In each test, 1000 complex samples are used to determine the modulation category. As shown in Fig. 6, the following observations can be made.

1) For all five algorithms, the modulation classification performance improves with an increasing SNR value.
2) Given the same SNR, ML-based algorithms perform much better than the cumulant-based algorithm, and DL-based algorithms perform best.
3) Considering the two ML-based algorithms, although SVM-7 employs more features, its accuracy is only slightly better than that of SVM-5 under low SNR.

TABLE II
CONFUSION MATRIX OF GOOGLENET WITH SNR = 4 dB

MODULATION	BPSK	4ASK	QPSK	OQPSK	8PSK	16QAM	32QAM	64QAM	ACCURACY
BPSK	1000	0	0	0	0	0	0	0	100%
4ASK	0	1000	0	0	0	0	0	0	100%
QPSK	0	0	1000	0	0	0	0	0	100%
OQPSK	0	0	0	1000	0	0	0	0	100%
8PSK	0	0	0	0	1000	0	0	0	100%
16QAM	0	0	0	0	0	741	17	242	74.1%
32QAM	0	0	0	0	0	6	938	56	93.8%
64QAM	0	0	0	0	0	264	53	683	68.3%

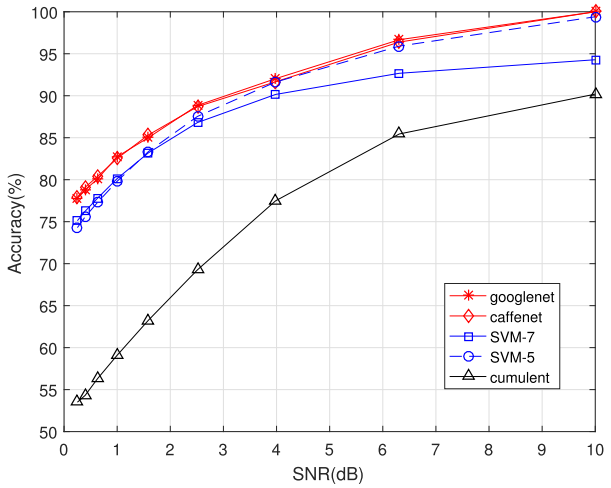


Fig. 6. Average classification accuracy versus SNR.

For higher SNR values, SVM-5 performs much better than SVM-7. This implies that the selection of features in ML (e.g., SVM-5 and SVM-7) can be a challenging issue and affects classification performance noticeably. Note that DL-based algorithms eliminate the need for manual feature selections.

4) The performance results of two DL-based algorithms are almost the same and they outperform all other algorithms.

F. Complexity and Feasibility

In order to evaluate computational complexity and application feasibility, Table III shows the average calculation time of ML and DL-based algorithms to complete one modulation classification task. The experiments are performed on our computing server equipped with an Intel Xeon X5690 central processing unit and a Nvidia Tesla K40m GPU. Calculation time is averaged over 8000 tests of modulation classification. Library for Support Vector Machines [25] and Caffe are adopted for calculation. No cropping is implemented during testing.

According to Table III, DL-based algorithms consume more calculation time and have a higher computational complexity than ML-based algorithms. It is because DL-based algorithms utilize deep networks to avoid manual feature extraction and achieve a higher modulation classification accuracy. Note that the calculation time of ML-based algorithms increases as either SNR decreases or the feature number increases. For SVM-7 at SNR = 0 dB, it reaches as much as 4.2 ms. Calculation time of ML-based algorithms varies with the SNR, since numerical calculation is adopted to solve the convex optimization problem of the SVM. When the SNR is lower, numerical calculation is more difficult to converge and requires more iterations. On the contrary, the calculation time of DL-based algorithms always keeps constant. For AlexNet and GoogLeNet, it is about 6.6 and 18 ms, respectively. Such a computational complexity is acceptable for many practical communications systems, especially in scenarios where modulation does not change frequently.

TABLE III
AVERAGE CALCULATION TIME OF DIFFERENT ALGORITHMS

SNR (dB)	Calculation Time(ms)			
	SVM-5	SVM-7	AlexNet	GoogLeNet
0	3.2	4.2	6.6	18.2
4	1.2	2.5	6.7	18.5
10	0.1	1.2	6.5	18.1

Note that Table III is obtained based on our experiment settings. Other DL frameworks (such as Tenserflow and Torch) and GPUs (such as Nvidia Tesla K80 and P100) that have more computational efficiency and capability can be exploited to reduce the calculation time. In the future days, the feasibility of our algorithms will be further enhanced due to the development of the DL framework and evolution of GPU.

VI. CONCLUSION

This paper studies DL models and their use for modulation classification. Several data conversion methods are introduced to generate gray images, enhanced gray images, and three-channel images. Compared with the traditional cumulant and ML-based modulation classification algorithms, DL-based approaches avoid manual feature selections and provide a superior classification accuracy. Currently, we are transplanting our algorithms into an embedded GPU platform (e.g., Nvidia Jetson TX2) to build up a DL-based modulation

classifier. In the next days, the impacts of fading channel and interference will be investigated.

REFERENCES

[1] B. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[2] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.

[3] Y. Wu *et al.* (2016). "Google's neural machine translation system: Bridging the gap between human and machine translation." [Online]. Available: https://arxiv.org/abs/1609.08144

[4] N. Jean, M. Burke, M. Xie, W. M. Davis, D. B. Lobell, and S. Ermon, "Combining satellite imagery and machine learning to predict poverty," *Science*, vol. 353, no. 6301, pp. 790–794, 2016.

[5] S. Min, B. Lee, and S. Yoon. (2016). "Deep learning in bioinformatics." [Online]. Available: https://arxiv.org/ftp/arxiv/papers/1603/1603.06430.pdf

[6] J. Fu, C. Zhao, B. Li, and X. Peng, "Deep learning based digital signal modulation recognition," in *Proc. 3rd Int. Conf. Commun., Signal Process., Syst.* (Lecture Notes in Electrical Engineering), vol. 322. Hohhot, China: Springer, 2015, pp. 955–964.

[7] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *Proc. 17th Int. Conf. Eng. Appl. Neural Netw. (EANN)*. Aberdeen, U.K.: Springer, 2016, pp. 213–226.

[8] S. Peng, H. Jiang, H. Wang, H. Alwageed, and Y.-D. Yao, "Modulation classification using convolutional neural network based deep learning model," in *Proc. 26th Wireless Opt. Commun. Conf. (WOCC)*, Apr. 2017, pp. 1–5.

[9] T. J. 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.

[10] A. Ali and F. Yangyu, "Automatic modulation classification using deep learning based on sparse autoencoders with nonnegativity constraints," *IEEE Signal Process. Lett.*, vol. 24, no. 11, pp. 1626–1630, Nov. 2017.

[11] F. Wang, Y. Wang, and X. Chen, "Graphic constellations and DBN based automatic modulation classification," in *Proc. 85th IEEE VTC-Spring*, Jun. 2017, pp. 1–5.

[12] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-air deep learning based radio signal classification," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 168–179, Feb. 2018.

[13] S. Hu, Y.-D. Yao, and Z. Yang, "MAC protocol identification using support vector machines for cognitive radio networks," *IEEE Wireless Commun.*, vol. 21, no. 1, pp. 52–60, Feb. 2014.

[14] M. W. Aslam, Z. Zhu, and A. K. Nandi, "Automatic modulation classification using combination of genetic programming and KNN," *IEEE Trans. Wireless Commun.*, vol. 11, no. 8, pp. 2742–2750, Aug. 2012.

[15] A. Swami and B. M. Sadler, "Hierarchical digital modulation classification using cumulants," *IEEE Trans. Commun.*, vol. 48, no. 3, pp. 416–429, Mar. 2000.

[16] O. A. Dobre, A. Abdi, Y. Bar-Ness, and W. Su, "A survey of automatic modulation classification techniques: Classical approaches and new trends," *IET Commun.*, vol. 1, no. 2, pp. 137–156, Apr. 2007.

[17] A. K. Nandi and E. E. Azzouz, "Algorithms for automatic modulation recognition of communication signals," *IEEE Trans. Commun.*, vol. 46, no. 4, pp. 431–436, Apr. 1998.

[18] M. L. D. Wong and A. K. Nandi, "Automatic digital modulation recognition using artificial neural network and genetic algorithm," *Signal Process.*, vol. 84, no. 2, pp. 351–365, 2004.

[19] B. Ramkumar, "Automatic modulation classification for cognitive radios using cyclic feature detection," *IEEE Circuits Syst. Mag.*, vol. 9, no. 2, pp. 27–45, Jun. 2009.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

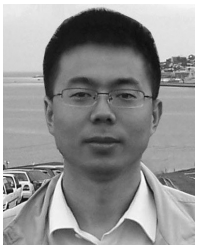
[21] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[22] Y. LeCun, "Generalization and network design strategies," Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, Tech. Rep. CRG-TR-89-4, 1989.

[23] Y. T. Zhou and R. Chellappa, "Computation of optical flow using a neural network," in *Proc. IEEE Int. Conf. Neural Netw.*, vol. 2, Jul. 1988, pp. 71–78.

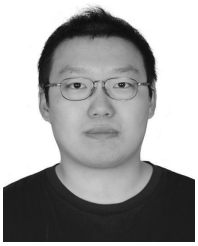
[24] Y. Jia *et al.* (2014). "Caffe: Convolutional architecture for fast feature embedding." [Online]. Available: https://arxiv.org/abs/1408.5093

[25] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 27:1–27:27, 2011. [Online]. Available: http://www.csie.ntu.edu.tw/~cjlin/libsvm



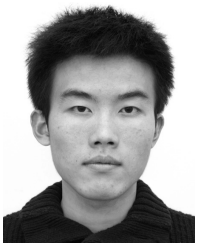
Shengliang Peng (S'08–M'12) received the Ph.D. degree in electric engineering from the Nation Mobile Communications Laboratory, Southeast University, Nanjing, China, in 2011.

He has been with the School of Information Science and Engineering, Huaqiao University, Quanzhou, China, since 2011, where he is currently an Associate Professor. From 2016 to 2017, he was a Visiting Research Scholar with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ, USA. His current research interests include cognitive radio, low-power wide-area networks, and deep learning.



Hanyu Jiang (S'14) received the B.S. degree in control science and engineering from the Harbin Institute of Technology, Harbin, China, in 2012, and the M.Eng. degree in computer engineering from the Stevens Institute of Technology, Hoboken, NJ, USA, in 2014, where he is currently pursuing the Ph.D. degree.

He has been a Research Co-Op at Nokia Bell Labs, Murray Hill, NJ, USA, since 2017. His current research interests include heterogeneous and parallel computing, multi-/many-core processor architecture, bioinformatics, and artificial intelligence.

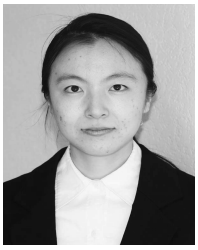


Huaxia Wang (S'13) received the B.Eng. degree in information engineering from Southeast University, Nanjing, China, in 2012, and the Ph.D. degree in electric engineering from the Stevens Institute of Technology, Hoboken, NJ, USA, in 2018.

In 2016 and 2017, he was a Research Intern with the Mathematics of Networks and Systems Research Department, Nokia Bell Labs, Murray Hill, NJ, USA. He joined FutureWei Technologies, Inc., Bridgewater, NJ, USA, in 2018. His current research interests include wireless communications, cognitive radio networks, and deep learning.

Hathal Alwageed (S'10) received the B.Sc. degree in computer engineering from King Saud University, Riyadh, Saudi Arabia, in 2003, and the M.Eng. degree in computer engineering from the University of South Carolina, Columbia, SC, USA, in 2010. He is currently pursuing the Ph.D. degree in computer engineering with the Stevens Institute of Technology, Hoboken, NJ, USA.

His current research interests include deep learning and communications systems.



Yu Zhou (S'15) received the B.Eng. degree in software engineering from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2011, and the M.Sc. degree in financial engineering and the Ph.D. degree in computer engineering from the Stevens Institute of Technology, Hoboken, NJ, USA, in 2013 and 2018, respectively.

Her current research interests include deep learning and cognitive radio networks.

Marjan Mazrouei Sebdani received the B.Sc. and M.Sc. degrees in electrical engineering from the Isfahan University of Technology, Isfahan, Iran, and the Ph.D. degree in electrical engineering from the Stevens Institute of Technology, Hoboken, NJ, USA, in 2017.

She is currently a Data Scientist at Publicis Media, New York, NY, USA. Her current research interests include machine learning, deep learning, and wireless communication.

Yu-Dong Yao (S'88–M'88–SM'94–F'11) received the B.Eng. and M.Eng. degrees from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 1982 and 1985, respectively, and the Ph.D. degree from Southeast University, Nanjing, in 1988, all in electrical engineering.

He was a Visiting Student with Carleton University, Ottawa, ON, Canada, from 1987 to 1988. From 1989 to 2000, he was with Carleton University, Ottawa, ON, Canada; Spar Aerospace Ltd., Montreal, QC, Canada; and Qualcomm Inc., San Diego, CA, USA. He has been with the Stevens Institute of Technology, Hoboken, NJ, USA, since 2000. He served as the Chair of the Electrical and Computer Engineering Department, Stevens Institute of Technology, from 2007 to 2018. He is also the Director of the Stevens Wireless Information Systems Engineering Laboratory, Stevens Institute of Technology. He holds one Chinese patent and 13 U.S. patents. His current research interests include wireless communications, cognitive radio, machine learning, and deep learning.

Dr. Yao was an Associate Editor of the IEEE COMMUNICATIONS LETTERS from 2000 to 2008 and the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY from 2001 to 2006 and an Editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS from 2001 to 2005. For his contributions to wireless communications systems, he was elected as a fellow of the National Academy of Inventors in 2015 and the Canadian Academy of Engineering in 2017. In 2018, he received the M.E. degree *(Honoris Causa)* from the Stevens Institute of Technology.