# Intelligent Radio Signal Processing: A Survey

**QUOC-VIET PHAM** [ID]1, (Member, IEEE), **NHAN THANH NGUYEN** [ID]2, (Member, IEEE),
**THIEN HUYNH-THE** [ID]3, (Member, IEEE), **LONG BAO LE** [ID]4, (Senior Member, IEEE),
**KYUNGCHUN LEE** [ID]5, (Senior Member, IEEE),
**AND WON-JOO HWANG** [ID]6, (Senior Member, IEEE)

[1]Korean Southeast Center for the 4th Industrial Revolution Leader Education, Pusan National University, Busan 46241, South Korea
[2]Centre for Wireless Communications, University of Oulu, 90014 Oulu, Finland
[3]ICT Convergence Research Center, Kumoh National Institute of Technology, Gumi 39177, South Korea
[4]Institut National de la Recherche Scientifique (INRS), University of Quebec, Montreal, QC H5A 1K6, Canada
[5]Department of Electrical and Information Engineering, Seoul National University of Science and Technology, Seoul 01811, South Korea
[6]Department of Biomedical Convergence Engineering, Pusan National University, Yangsan 50612, South Korea

Corresponding author: Won-Joo Hwang (wjhwang@pusan.ac.kr)

**ABSTRACT** Intelligent signal processing for wireless communications is a vital task in modern wireless systems, but it faces new challenges because of network heterogeneity, diverse service requirements, a massive number of connections, and various radio characteristics. Owing to recent advancements in big data and computing technologies, *artificial intelligence* (AI) has become a useful tool for *radio signal processing* and has enabled the realization of *intelligent radio signal processing*. This survey covers four intelligent signal processing topics for the wireless physical layer, including modulation classification, signal detection, beamforming, and channel estimation. In particular, each theme is presented in a dedicated section, starting with the most fundamental principles, followed by a review of up-to-date studies and a summary. To provide the necessary background, we first present a brief overview of AI techniques such as machine learning, deep learning, and federated learning. Finally, we highlight a number of research challenges and future directions in the area of intelligent radio signal processing. We expect this survey to be a good source of information for anyone interested in intelligent radio signal processing, and the perspectives we provide therein will stimulate many more novel ideas and contributions in the future.

**INDEX TERMS** Artificial intelligence, beamforming, channel estimation, deep learning, federated learning, machine learning, modulation classification, radio frequency, signal processing.

## I. INTRODUCTION

Radio signal processing plays a vital role in the engineering of all generations of wireless networks. With the emergence of many advanced wireless technologies and massive connectivity, processing radio signals in an efficient and intelligent way presents both challenges and opportunities. Additionally, next-generation wireless systems are likely to rely not only on the sub-6 GHz, but also on the mmWave and THz frequency bands, and non-radio frequencies (RFs) such as the visible and optical bands [1], [2].

The associate editor coordinating the review of this manuscript and approving it for publication was Baoping Cai [ID].

Furthermore, the use of massive multiple input and multiple output (massive MIMO) in fifth generation (5G) wireless systems and beyond demands sophisticated radio signal processing schemes. Radio signals were conventionally processed by mathematical model-based algorithms. Despite promising results, these conventional methods have various shortcomings including high complexity as well as poor scalability, online implementation, and adaptivity to dynamic environments. Recent advancements in computing hardware and big data processing have rendered AI a useful tool for radio signal processing, thereby realizing the term *intelligent signal processing*. Undoubtedly, AI is expected to play a key role in solving many complex problems that are neither tractably

nor efficiently overcome by conventional model-based approaches.

## A. INTELLIGENT SIGNAL PROCESSING: AN OVERVIEW

The past three years have witnessed growing interest in the application of AI to wireless signal processing. A good example is the IEEE initiative (https://mlc.committees.comsoc.org/) to promote the use of AI for physical layer signal processing, e.g., modulation recognition (also known as modulation classification), channel estimation, signal detection, channel encoding and decoding, localization, and beamforming. Various AI-based algorithms and deep learning (DL)-based models have been proposed as alternatives to the present model-based approaches. An unspoken consensus is that model- and AI-based approaches have different particularities but complementary capabilities, i.e., AI is not a universal solution and should be used for tasks that cannot be efficiently attempted by conventional approaches. For instance, the globally optimal solution for signal processing problems can be obtained via existing model-based mechanisms such as optimal signal detection [3] and optimal beamforming [4]. In general, AI-based algorithms cannot outperform optimal model-based schemes if they are used to solve the same problem, but they have the potential for real-time signal processing. Moreover, several scenarios exist in which AI may significantly improve radio signal processing over conventional model-based approaches. In the following, we briefly discuss these scenarios along with representative examples.

### 1) ALGORITHMIC APPROXIMATION

A common limitation preventing algorithms from finding the optimal solution is the difficulty of real-time executions; therefore, they are impractical for real-time implementation. Several approaches, e.g., heuristics, metaheuristics, and problem decomposition, have been proposed to optimize the tradeoff between computational complexity and performance. However, the real-time implementation of the underlying algorithms is quite challenging. For this case, the use of AI techniques appears to be a promising solution. In particular, the data generation and training phases can be executed offline while the system operates in real time by using the trained model. For instance, Huynh-The *et al.* [5] proposed a DL architecture for automatic modulation classification (AMC), namely MCNet, which was 93.59% accurate at a signal-to-noise ratio (SNR) of 20 dB with an inference time of only 0.095 ms.

### 2) UNKNOWN MODEL AND NONLINEARITIES

Many physical phenomena cannot be accurately modeled. Therefore, conventional model-based algorithms usually fail to obtain efficient solutions. For instance, fiber nonlinearities (e.g., signal distortion and self-phase modulation) in optical systems together with the adoption of coherent communication render model-based methods ineffective for network optimization [6]. To mitigate the nonlinearities and

perform signal detection, AI techniques (e.g., an end-to-end learning approach [7]) can be utilized with very low bit error rates (BER). The end-to-end learning approach [8] has found many applications in scenarios in which the channel model is unknown or well-established mathematical models are unavailable. Another application that involves the use of DL to address hardware nonlinearities in MIMO systems (e.g., hardware impairments) was presented [9]. These researchers proposed two DL-based estimators to exploit the nonlinear characteristics with the aim of improving the estimation performance. Nonlinearity was also observed in MIMO systems with low-bit analog-to-digital converters. In an attempt to mitigate this nonlinear effect, Nguyen [10] proposed a DNN model to jointly optimize the channel estimation and training signal. The model outperformed the linear channel estimator in various practical settings.

### 3) ALGORITHM ACCELERATION

Another direction intelligent signal processing has been taking is to use AI to facilitate and accelerate existing algorithms. This approach differs markedly from the two scenarios discussed above in that an existing model-based algorithm is completely replaced by an AI-based algorithm, i.e., an end-to-end learning paradigm. For instance, many DL-based algorithms have been proposed to improve and accelerate near-optimal detection schemes. Nguyen and Lee [11] employed a DL model, namely FS-Net, to initialize the highly reliable solution for the tabu search (TS) detection scheme, and also proposed an early termination scheme to further accelerate the optimization process. Compared with the original TS scheme, the DL-aided TS detector can reduce the computational complexity by approximately 90% at an SNR of 20 dB with similar performance. DL was also employed to generate the initial radius for the sphere decoding (SD) detector [12].

## B. STATE-OF-THE-ART

Owing to the importance of AI for physical layer signal processing, a number of surveys and magazine articles have been published on this topic over the past few years. DL techniques for solving physical layer signal processing problems such as modulation, channel coding, detection, and end-to-end learning were reviewed [13]. However, this survey mainly focused on reviewing DL techniques and did not include many up-to-date studies as it was published quite a long time ago. The concept of end-to-end DL was first introduced in 2017 [8] to model the entire physical communication as an autoencoder DNN. This discovery constituted a major breakthrough in the design of communication systems and has been widely employed in many research efforts. A chapter in a recent book [14] described the benefits and the use of end-to-end learning for channel estimation, signal identification, and wireless security. Qin *et al.* [15] demonstrated the applications of DL to the optimization of individual signal processing blocks in the physical layer (e.g., signal compression and detection) and also end-to-end design.

**TABLE 1.** Summary of existing surveys and tutorials on AI techniques for wireless networking and signal processing.

| Paper | AI Models | | | | Applications |
|---|---|---|---|---|---|
| | ML | DL | DRL | FL | |
| [6] | ✓ | | | | IoT applications, e.g., smart health, smart city, smart transportation, and smart industry. |
| [8] | | ✓ | | | Proposal of end-to-end learning and example of modulation classification. |
| [13] | | ✓ | | | DL for modulation, channel coding, detection, and end-to-end learning. |
| [14] | | ✓ | | | End-to-end learning for wireless networks: channel estimation, signal identification, and wireless security. |
| [15] | | ✓ | | | Applications of DL for block optimization and end-to-end design. |
| [16] | | ✓ | | | Model-driven DL and demonstrations of receiver design, signal detection, and channel estimation. |
| [17] | | ✓ | | | Applications of deep unfolding for MIMO systems: signal detection and beamforming. |
| [18] | ✓ | ✓ | ✓ | ✓ | Discussions and examples of model-based, AI-based, and hybrid methods for wireless networks. |
| [20] | ✓ | ✓ | | | ML techniques for solving challenges in massive machine-type communications. |
| [21] | | ✓ | ✓ | | Security preservation and threats in IoT. |
| [23] | ✓ | ✓ | | | Intrusion detection in wireless networks. |
| [24] | ✓ | ✓ | | | Network intrusion detection systems. |
| [25] | | ✓ | | | Traffic control in network systems, e.g., sensor networks, flow prediction, social networks, and cognitive radio. |
| [26] | ✓ | | | | Network traffic classification. |
| [28] | ✓ | | | | Software-defined networking, e.g., traffic classification, routing, security, and quality of service (QoS) prediction. |
| [29] | ✓ | ✓ | ✓ | | 5G applications, e.g., UAV communications, wireless virtual reality, self-organized networks, and IoT. |
| [30] | | ✓ | ✓ | | Mobile big data applications, e.g., physical coding, spectrum allocation, and routing protocols. |
| [31] | | ✓ | ✓ | | DL for 5G applications, e.g., network security, network control, localization, mobility analysis, and data analytics. |
| [32] | | | ✓ | | Network access, caching and offloading, security and privacy, resource scheduling, and data collection. |
| [33] | | | | ✓ | Edge computing applications, e.g., cyberattack detection, edge caching, and user association. |
| [34] | ✓ | ✓ | ✓ | | ML applications for wireless networks. |

He *et al.* [16] discussed the significance of model-driven DL techniques in physical layer design and illustrated use cases for receiver design, signal detection, and channel estimation. A brief on model-driven deep unfolding for MIMO signal detection and beamforming was presented [17]. Furthermore, Zappone *et al.* [18] discussed model-based, AI-based, or hybrid methods and presented examples for designs of the wireless physical layer. A brief with demonstration of modulation and classification was carried out [19].

Another line of work included various surveys and tutorials on the applications of AI to wireless networking. In particular, the use of AI for Internet of Things (IoT) applications and massive connectivity, privacy, and security was reviewed [20], [21]. Gu *et al.* [22] conducted a survey on AI applications for optical communications and networking. AI-based solutions for cybersecurity problems (e.g., misuse detection and anomaly detection) were discussed [23], [24]. Fadlullah *et al.* [25] reviewed the adoption of AI for network traffic control systems. The use of machine learning (ML) techniques for designing traffic classification strategies was studied [26]. ML techniques for applications including computational offloading, mobile big data, and mobile crowdsensing at the network edge were reviewed [27]. Xie *et al.* [28] discussed opportunities and challenges arising from the use of ML techniques for software-defined networking (SDN). A tutorial on artificial neural networks (ANN) for wireless networking was presented [29]. Mao *et al.* [30] conducted a survey of mobile networking from the mobile big data perspective for which a top–down approach was used. Another survey on DL for 5G mobile and wireless networking was presented [31]. The use of deep reinforcement learning (DRL) for wireless communications and

networking was reviewed [32]. Federated learning (FL) in mobile edge networks was surveyed [33]. Four main types of ML (i.e., supervised learning, unsupervised learning, DL, and reinforcement learning) and their application to wireless networks were considered [34]. The use of swarm intelligence for next-generation wireless networks was recently reviewed in [35]. Table 1 summarizes existing surveys and tutorials on AI for wireless networking and radio signal processing.

## C. CONTRIBUTIONS AND ORGANIZATION OF THIS PAPER

Notwithstanding the plethora of surveys on AI applications for research topics, we are still unaware of any comprehensive survey on the use of AI techniques for intelligent radio signal processing. Existing surveys (e.g., [21], [22], [25], [28]) are limited to the scope of mobile networking and communications. Furthermore, most existing studies focus on certain AI techniques and their applications to wireless research such as channel encoding and decoding [8], [36], unfolding DL for MIMO systems [17], DL for wireless networks [13], [15], [16], and tracking and localization [37]. In contrast, our aim was to provide a comprehensive survey of AI applications for various aspects of wireless physical signal processing. In this vein, we first provide the fundamentals of AI techniques, including ML, DL, and FL and discuss the need to apply AI approaches to design intelligent methods to process radio signals. Then, we review AI applications pertaining to four different key signal processing areas, namely modulation classification, signal detection, channel estimation, and MIMO beamforming optimization. We also highlight a number of challenges and future research directions in the area of intelligent radio signal processing. Our contributions can be summarized as follows.

- We present an overview of AI techniques with potential application to radio signal processing. Specifically, in Section II we present the fundamentals of AI, ML, DL, DRL, and FL. We also summarize the motivations and advantages of using AI for radio signal processing.
- We survey the application of AI techniques to four signal processing themes, namely, modulation classification, signal detection, channel estimation, and MIMO beamforming optimization. We present these themes along with basic information to aid readers to design intelligent signal processing frameworks.
- We provide a set of research challenges and also highlight a number of potential directions along which future investigations would ideally need to follow to enable performance improvement of intelligent radio signal processing methods.

The remainder of this survey is organized as follows. Section II presents the fundamentals of AI techniques with application to the intelligent processing of radio signals in wireless and communication networks. Section III discusses the AI applications for AMC. In Section IV, state-of-the-art AI applications for signal detection are reviewed. Section V reviews the literature on AI techniques for channel estimation and MIMO beamforming. In Section VI, we present the challenges associated with and unresolved challenges arising from existing research devoted to AI for radio signal processing and further highlight potential research directions. The survey is concluded in Section VII.

## II. ARTIFICIAL INTELLIGENCE: BACKGROUND INFORMATION

This section provides an overview of ML, DL, DRL, and FL.

### A. MACHINE LEARNING: PRELIMINARIES

AI is one of the most pioneering sciences and has been in development since 1956 when the name AI was adopted by McCarthy and colleagues. The foundations of AI are based in many long-standing disciplines, e.g., philosophy, mathematics, economics, neuroscience, psychology, computer engineering, control theory, cybernetics, and linguistics [38]. Today, AI is a thriving field which has found many applications in the field of engineering. ML, which is the principal AI discipline, allows patterns to be mined/learned from raw data to gain knowledge. In general, ML can be classified into three main types: supervised learning, unsupervised learning, and reinforcement learning (RL).

Supervised learning is concerned with mapping known inputs with known outputs given a training set including both inputs and outputs. Basically, supervised learning can be divided into two types: regression and classification, the respective output values of which are continuous and discrete. Examples of popular supervised learning techniques are support vector machine (SVM), K-nearest neighbor (KNN), Naïve Bayesian model, and decision trees [34]. On the other hand, in unsupervised learning, the labels of the output are not included in the training data and the goal of unsupervised learning is to learn useful representations and properties from the input data. Increasing efforts have been made to utilize unsupervised learning for designing wireless and communication networks. In fact, massive amounts of unstructured and unlabeled data are generated by wireless devices and emerging applications. Moreover, annotating the ground truth for a large number of examples has a huge associated cost. The performance of unsupervised learning models is typically inferior to that of models that use supervised learning. Lastly, RL learns from interactions with the environment, i.e., the learning agent continuously interacts with the environment and adopts good policies to make decisions so as to maximize the reward. Two main features of RL are trial-and-error learning (i.e., using error information and evaluative feedback to update actions/policies) and delayed reward (i.e., an action does not only affect the immediate reward but also the future reward) [39].

### B. DEEP LEARNING

The appropriate presentation of handcrafted features extracted from raw data is a prerequisite for conventional ML algorithms, whereas DL is able to directly learn a complicated model from raw data by using a neural network [40]. The input and output are presented at the first layer (i.e., the visible layer) and the last layer (i.e., the output layer), respectively. Hidden layers are used to increase the level of feature abstraction, i.e., they calculate representational features at multiscale resolutions. Specifically, the visible layer receives the input data and then extracts simple features, which are further abstracted by the subsequent hidden layers, and the output layer uses additional functions to transform the features received from the last hidden layer into the output. Owing to developments in computing infrastructure, big data, and data science, DL has been recognized as a crucial technology and found many practical applications, e.g., image and speech recognition, natural language processing, drug discovery, self-driving vehicles, and mobile communications and networking.

Depending on the structure of the neural network, different ANN architectures have been developed for various learning tasks and data modalities. In the following, we briefly introduce important ANN architectures that have been widely used in the studies we reviewed.

#### 1) FEEDFORWARD NEURAL NETWORKS (FNN)

An FNN is composed of a visible layer, an output layer, and one or more hidden layers. The information traverses the neural network and feedback connections from the outputs are not used. Similar to a general neural network, the basic components of an FNN are neurons (i.e., the nodes in the network), weights (i.e., the numerical values representing connections), and activation functions, which are used to determine the output of neurons in the neural network. Currently, most ANN models use the rectified linear unit (ReLU) $f(z) = \max\{0, z\}$, the logistic sigmoid $f(z) = 1/(1 + \exp(-z))$, and the hyperbolic tangent (tanh) $f(z) = \tanh(z)$

for nonlinear transformation. Other activation functions that have been investigated to improve ANNs include the adaptive piecewise linear [41] and the Swish [42].

Previously [43], it was found that any continuous function can be approximated by an FNN composed of one hidden layer and a finite number of neurons. To optimize the neural network, several approaches have been employed including first-order gradient-descent algorithms (e.g., backpropagation and its variants such as Quickpro and resilient propagator), second-order minimization algorithms (e.g., conjugate gradient, quasi-Newton, Gauss–Newton), and metaheuristics (e.g., the whale optimization algorithm and Harris Hawks optimization) [44]. We invite interested readers to refer to [45] for a survey on metaheuristic optimization for FNNs and [40, Section 6.5] for an overview of the backpropagation method and derivative-based algorithms. An example of FNN architecture with two hidden layers is shown in Fig. 1. In the following, the two terms, FNN and DNN, are usually used interchangeably.
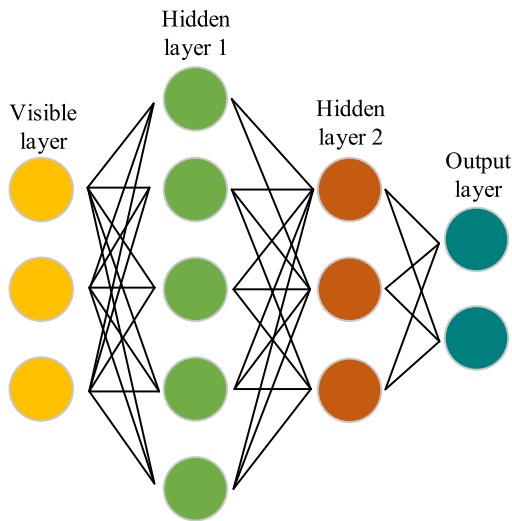


**FIGURE 1.** Architecture of an FNN with two hidden layers.

### 2) CONVOLUTIONAL NEURAL NETWORKS (CNN)

In general, CNNs are suitable for processing high-dimensional unstructured data such as images, where many backbone CNNs are initially introduced for image classification. The first advantage of a CNN is its *sparse interactions* feature, which is enabled by setting the size of the kernel to be smaller than that of the input, thus improving the storage requirements and statistical efficiency [40]. The second advantage originates from the *parameter sharing* concept used in CNNs, i.e., the kernel is applied across the entire input to create the feature map. In addition, the convolution and pooling operations render CNNs *invariant* and *equivariant* to translations of the input. Finally, CNNs are proficient in automatically extracting high-level representational features for mining intrinsic information. As shown in Fig. 2, the CNN architecture has three basic components: convolution,
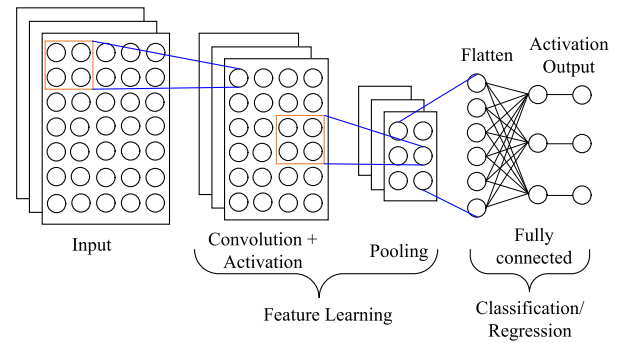


**FIGURE 2.** Architecture of a CNN.

a nonlinear activation function, and pooling (i.e., down-sampling in the literature). To date, different efficient CNN architectures have been proposed such as GoogleNet, ResNet, Inception-ResNet-v2, SENet, and EfficientNet-B7 [46].

The convolution operation is the core feature of CNNs and has two parts: input and kernel (i.e., filter). The kernel in a convolutional layer is specified by a predefined kernel spatial size, where its depth size is identical to the number of input channels. The feature map generated by the convolutional operation may have different spatial dimensionality compared with the input. In particular, zero padding and valid padding are used to maintain and change the dimensionality, respectively. Notably, the size of the kernel (width and height) indicates the number of neurons in the input used to infer a neuron in the output feature map. For example, a kernel of size $5 \times 4$ implies that 20 neurons are used to calculate an output neuron via the dot product of kernel weights and input elements. The next component of a CNN is the activation function, which does not change the size of the input it receives and processes. The pooling layer, the last principal component of a CNN, usually has the function of reducing the spatial dimension of feature maps. This family of layers, including max and average pooling, has a similar operating principal to that of the convolutional layer without learnable parameters. For instance, the max-pooling layer returns the maximum value of entries of the input with a receptive field, the so-called pool size. Therefore, a pooling map can be considered as a lower-resolution version of the feature map when it is down-sampled along the vertical and/or horizontal dimensions. As pooling operates over spatial regions, pooling can help the features to become invariant to small translations of the input. Common pooling methods are average pooling, max pooling, and L2-pooling.

### 3) RECURRENT NEURAL NETWORKS (RNN)

While CNNs are suitable for processing high-dimensional data, RNNs are usually used to process sequential data, i.e., in situations in which the prediction depends on not only the current sample but also on previous samples. Each neuron in RNNs has the capability to memorize the output, which is fed into the neuron as a subsequent input. An illustration of an RNN with one hidden layer and a length sequence of
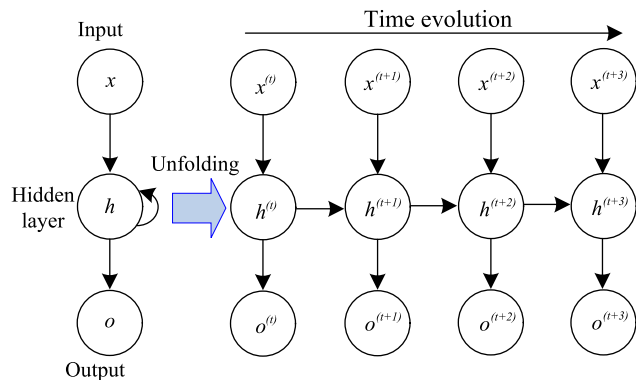
**FIGURE 3. Architecture of an RNN.**



**FIGURE 4. Illustration of the structure of a DRL algorithm.**

four inputs is shown in Fig. 3. In particular, the value $h^{(t)}$ to be predicted at time $t$ is a function $g^{(t)}(\cdot)$ of the input sequence $\left(x^{(t)}, x^{(t-1)}, \ldots, x^{(1)}, x^{(0)}\right)$. To apply a backpropagation algorithm to RNNs, the unfolding concept is used to transform an RNN into a computational graph, which has a repetitive structure and thus enables the sharing of learning parameters across the neural network. Mathematically, the unfolding recurrence at the time $t$ can be modeled as $h^{(t)} = g^{(t)}\left(x^{(t)}, x^{(t-1)}, \ldots, x^{(1)}, x^{(0)}\right)$.

Echo state networks, liquid state machines, gated RNNs, and long short-term memory are popular variants of RNNs. RNNs have been used to solve many practical problems, e.g., speech recognition, human activity recognition, and bioinformatics [40]. In wireless and communication networks, RNNs have also found many applications, for example, a bidirectional neural model was used [47] to learn the proactive caching policy at the network edge, and a variant of RNNs was employed [48] for Wi-Fi indoor localization.

The unfolding concept has been used to improve many iterative algorithms. The key idea is that each iterative cycle of an iterative algorithm is modeled as a layer of the neural network, which is trained to enable the algorithm to converge to the optimum. One such application [49] was based on a DetNet model that was proposed for signal detection by unfolding the projected gradient descent method. The main advantage is that this model obviates the need to determine the network configurations such as the number of hidden layers. Specifically, the number of layers of the neural network is equivalent to the number of iterative cycles of the iterative algorithm, and the number of neurons is specified by the sizes of the input, output, and optimizing variables. Furthermore, deep unfolding incorporated in certain advanced model-based algorithms and transfer learning can improve the model efficiency e.g., faster convergence while requiring a smaller dataset to deliver the same performance [18].

Apart from the three types of ANNs described above, numerous other ANNs have been proposed with different design philosophies, e.g., an autoencoder and deep generative models such as the restricted Boltzmann machine and deep belief network. An autoencoder, which is a specialized ANN for 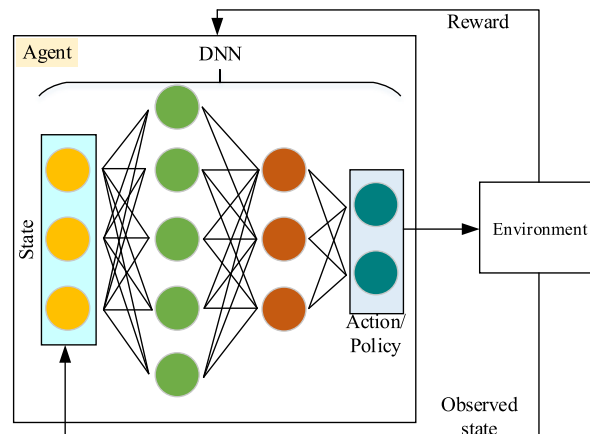learning useful properties of the data, is effective with unlabeled examples. A restricted Boltzmann machine is a kind of deep generative model for learning the probability distribution over a set of examples. We invite interested readers to refer to a recent book [40] and surveys on ANNs and their practical applications including speech recognition, pattern recognition, computer vision, agriculture, arts, and nanotechnology [50], [51].

### 4) DEEP REINFORCEMENT LEARNING

DRL leverages the strengths of DNNs to improve the performance of RL algorithms [52]. Three main approaches exist to solve RL problems: the value function-based approach, policy-based approach, and hybrid actor–critic approach. The value function-based method relies on the estimation of the expected reward of each state, whereas the policy search-based method directly finds the optimal policy. The actor–critic method learns both the policy and value functions, and effectively overcomes the imbalance between variance and bias owing to the policy search and value function methods. For high-dimensional problems, DNNs can be exploited to learn the optimal value function, the optimal policy, or both in case of the actor–critic method [32]. An illustration of the structure of the DRL algorithm is presented in Fig. 4, where DNNs are used to approximate the control policy. Inspired by a proposal [52], DRL has found many successes in various domains and has been widely used in signal processing studies. For further details, we invite interested readers to refer to a recent survey [32].

### C. FEDERATED LEARNING

To protect sensitive information as well as preserve individual privacy, Google invented the concept of FL [53]. FL enables an AI model to be trained without requiring all data to be stored and processed at a centralized server, which is typically referred to as the aggregation server in FL literature. In other words, the data of individual users remain in local storage in their end devices (i.e., the participant) and do not need to be transmitted to the server [54]. In FL, the server receives
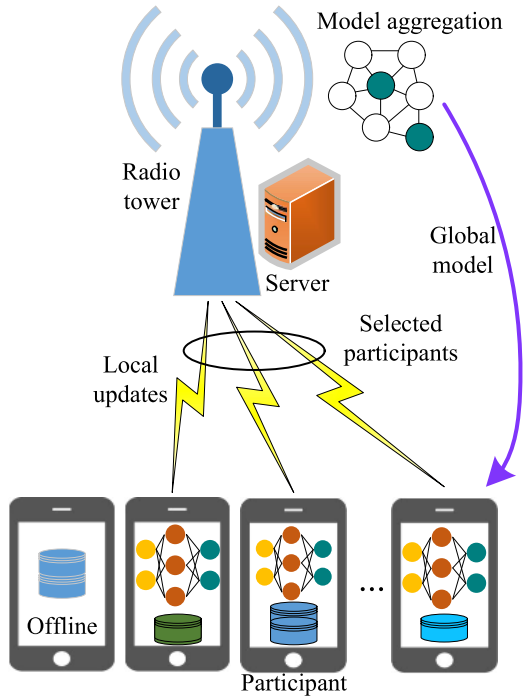
**FIGURE 5.** Illustration of an FL system.

locally computed models from a number of devices, which are then aggregated to update the global model. In this way, FL can preserve data privacy and user security, although FL still relies on the trust of the aggregation server. FL has found many promising applications in various fields. The notable success of FL in Google's next word prediction application has motivated the adoption of FL for many other applications [53] including smart retail, multiparty database querying, smart healthcare, and vehicular networks [55]. The FL system is illustrated in Fig. 5.

Since the first FL paper from Google, a large number of studies on FL have been conducted over the last few years. For a communication perspective, interested readers may refer to review articles [33], [56] and references therein. Lim *et al.* [33] first discussed the challenges of FL implementations, including communication cost, resource allocation, and privacy and security issues. In this paper, we further review existing FL applications at the network edge such as cyber-attack detection, edge caching and computational offloading, user association, and vehicular networks. The interplay between edge computing and DL has been reviewed in terms of *intelligent edge* and *edge intelligence* [56].

## III. MODULATION CLASSIFICATION
This section presents a review of applications of AI techniques for automatic modulation classification.

### A. FUNDAMENTALS OF MODULATION CLASSIFICATION
The last decades have seen tremendous advancement and development of innovative communication standards and

technologies to satisfy the ever-increasing demand of many wireless applications and services. Dense network deployment with aggressive spectrum reuse to meet the growing mobile traffic demand has resulted in various undesirable effects such as signal distortion and co-channel interference. Signal recognition and modulation classification allow us to more effectively monitor and manage spectrum usage and sharing, which can potentially enhance the network performance [57]. AMC, a fundamental process to analyze the characteristics of radio signals in the physical layer, plays a vital role in intelligent spectrum monitoring and management and is typically deployed in AI-powered wireless communication systems. This is because it enables the system to blindly identify the modulation format of an incoming radio signal at the receiver [58]. From the ML perspective, modulation classification can be framed as a multiclass decision-making problem, where the intrinsic radio characteristics are obtained using conventional feature engineering algorithms for learning a trainable classification model. The ability to correctly distinguish advanced modulations (e.g., high-order digital modes) under harmful transmission environments, such as a multipath fading channel with additive noise, remains a challenging research topic [59] and has received considerable attention from the signal processing and communication communities.

Modern communication systems employ different advanced analog and digital modulation techniques to achieve good tradeoff between spectrum efficiency and transmission reliability. Fundamentally, an analog modulation technique encodes an analog baseband signal onto a high-frequency periodic waveform (i.e., carrier signal), whereas digital modulation techniques allow a digital low-frequency baseband signal to be transmitted over a high-frequency carrier waveform. These two modulation families can modify different waveform characteristics of the carrier signal, including the amplitude, frequency, phase, and a combination of amplitude and phase. At the receiver, the considered modulated signal must be assigned to the most appropriate modulation class by exploiting certain radio characteristics and a trained classifier. The complex envelope of received radio signal $y(n)$ can be written as follows:

$$y(n) = x(n, H_k) + g(n), \qquad (1)$$

where $g(n)$ is the additive white Gaussian noise (AWGN). The noiseless signal $x(n, H_k)$ under transmission channel effects can be expressed as follows:

$$x(n, H_k) = Ae^{2\pi f_o n\kappa + \zeta_n} \sum_{k=-\infty}^{\infty} x(k) h(n\kappa - k\kappa + \xi_\kappa \kappa), \qquad (2)$$

where $A$ is the signal amplitude, $f_o$ is the carrier frequency offset, $\kappa$ is the symbol spacing (or interval), $h(\cdot)$ refers to the synthetic effect of the residual baseband channel, $\zeta_n$ is the varying phase offset, $x(k)$ refers to the symbol sequence of the original data over a specific modulation scheme, and $\xi_\kappa$ is the timing error (or timing offset between the transmitter

and the receiver). In general, an AMC scheme is developed to accurately predict the modulation format of $x(n, H_k)$ that is performed by a trained classifier that effectively learns the informative features of $y(n)$ by using some specific ML algorithm. However, this classification task is challenging because it is necessary to process many high-order modulation schemes, considering synthetic channel deterioration.

### B. STATE-OF-THE-ART AMC METHODS

Numerous methods have been proposed for modulation classification in communications, where AI methods have been widely used to improve the performance in terms of classification accuracy and processing speed. Based on the progressive development of AI in the last decades, especially the recent explosion of DL, AMC methods that have been reported in the literature can be grouped into two major categories as follows:

- *Conventional approaches*: Various methods in this group have employed conventional AI techniques and traditional ML algorithms, which can be further divided into two sub-classes: likelihood-based and feature-based approaches. For the likelihood-based approaches, the output of modulation classification is determined with the aim of maximizing the probability of a received signal associated with a certain modulation scheme. The underlying distribution parameters of the scheme are estimated by using expectation/conditional maximization (ECM) algorithms [60]. Formulated as a composite problem for hypothesis testing, the maximum-likelihood modulation classification draws the decision as follows:

$$\hat{\mathcal{H}} = \arg\max_{\mathcal{H}_{ij}} \ln p(y_1, \ldots, y_T | \mathcal{H}_i), \qquad (3)$$

where $\mathcal{H}_{ij}$ is the hypothesis model associated with the modulation format $\mathcal{M}_i$ ($i = 1, \ldots, K$), where $K$ is the number of modulation formats, deduced from the observation signal $y_j$ ($j = 1, \ldots, T$), where $T$ is the number of observations, and $\ln p(y | \mathcal{H}_i)$ refers to the log-likelihood function [61]. In fact, the likelihood-based approaches can achieve optimal performance with perfect knowledge based on information of signal and channel models [3], but they are computationally expensive in terms of parameter estimation [62].

Compared with likelihood-based approaches, feature-based methods have been widely deployed in practical systems thanks to their easy implementation, lower complexity, and stronger robustness with various transmission channel scenarios. A typical ML framework requires feature engineering and classification processes, where certain handcrafted feature extractors (i.e., descriptors) are used to mine radio characteristics and traditional classifiers can be employed to learn the modulation patterns in the supervised manner. Certain methods in this subclass have achieved a good trade-off between model accuracy and complexity by

using advanced feature selection schemes and sophisticated ML algorithms [60].

- *Innovative DL-based approaches*: Inspired by great success in the fields of image processing and computer vision [63]–[65], the DL technique has been exploited for modulation classification, wherein several deep network architectures, such as RNN, long short-term memory (LSTM), and CNN, have been considered. Compared with traditional ML, DL has important advantages because it can automatically learn high-level features for more effective modulation discrimination and it can effectively process wireless big data [66]. With an appropriately built computing platform with graphics processing units (GPUs), the execution speed of both learning and inference (i.e., prediction) processes can be accelerated significantly to satisfy the high reliability and low latency requirements of emerging wireless applications and services.

### 1) CONVENTIONAL AMC APPROACHES

In the last decades, many conventional AMC methods have been proposed to enable dynamic spectrum access and intelligent spectrum management, where the expectation-maximization (EM) algorithms were employed to build maximum in likelihood-based classifiers [67]–[73]. Zhang *et al.* [67] took advantage of the EM algorithm to estimate the maximum-likelihood of the unknown for modulation classification in a cooperative multiuser scenario. For each hypothesis, the EM algorithm performs an expectation step (E-step) and a maximization step (M-step) at each iterative step to estimate the unknown channel amplitude $a_{c,km}^{t+1}$ and phase $\phi_{c,km}^{t+1}$ that are associated with the radio signal encoded by the modulation format $c$ and transmitted from the $m$-th user to the $k$-th receiver. The calculations in these two steps can be expressed as follows:

$$\mathrm{E-step} : J\left(\boldsymbol{\theta}|\boldsymbol{\theta}_c^{(t)}\right) = \mathbb{E}_{z|\boldsymbol{x},\boldsymbol{\theta}_c^{(t)}}\left[\ln p(z|\boldsymbol{\theta})\right], \qquad (4)$$

$$\mathrm{M-step} : \boldsymbol{\theta}_c^{(t+1)} = \arg\max_{\boldsymbol{\theta}} J\left(\boldsymbol{\theta}|\boldsymbol{\theta}_c^{(t)}\right), \qquad (5)$$

where $J\left(\boldsymbol{\theta}|\boldsymbol{\theta}_c^{(t)}\right)$ refers to the expected value of the log likelihood function of unknown parameters $\boldsymbol{\theta}$ with respect to the conditional distribution $z$ given the received samples $\boldsymbol{x}$ and the current estimates of parameters $\boldsymbol{\theta}^{(t)}$. By decoupling the multivariate maximum-likelihood problem into multiple separated optimization problems, the proposed method can estimate the complete data and unknown parameters more effectively. The Cramér-Rao lower bounds (CRLBs) for estimating unknown multipath channels were applied to enable the EM algorithm to reach the performance upper bound of modulation classification [68]. To distinguish continuous phase modulation signals, a maximum-likelihood-based classifier [69] was introduced with the Baum–Welch (BW) algorithm to estimate the unknown fading channel coefficients. In the E-step of the EM algorithm, the BW method was applied to iteratively

maximize the auxiliary function as follows:

$$J\left(\boldsymbol{\theta}|\boldsymbol{\theta}_c^{(t)}\right) = \sum_v p_c\left(\boldsymbol{x}, \boldsymbol{v}|\boldsymbol{\theta}_c^{(t)}\right) \log p_c\left(\boldsymbol{x}, \boldsymbol{v}|\boldsymbol{\theta}\right), \qquad (6)$$

where $\boldsymbol{v}$ refers to the hidden variables obtained by a hidden Markov model (HMM).

For identification of the modulation parameters, including the signal constellation and the number of subcarriers, in orthogonal frequency division multiplexing (OFDM) with index modulation, Zheng and Lv [70] recommended two likelihood-based classifiers of an average likelihood ratio test (ALRT) and a hybrid likelihood ratio test (HLRT). For the known channel state information (CSI) scenario, the HLRT classifier first determines the indices of active subcarriers and estimates transmitted symbols subsequently, whereas the ALRT classifier averages out the set of subcarrier indices with higher complexity. On the contrary, a blind AMC only employs the HLRT classifier to distinguish interesting subcarriers corresponding to each hypothesis of modulation parameter combination via an energy-based detector. To accelerate the convergence process of the ECM algorithm to blindly estimate the channel parameters in flat fading and nonGaussian noise impairments, Chen *at el.* [72] upgraded the squared extrapolation method by adding a parameter-checking scheme. The enhanced method also derives a convergence point of log-likelihood function more reliably compared with the original one.

The design of AMC strategies for MIMO systems is more challenging than those for single-input single-output systems because the incoming signal at the receiver is a mixture of multiple signals transmitted by different antennas. Therefore, the effects of channel impairment on the modulation signals are dissimilar. As a result, it is difficult to properly estimate channel characteristics via EM algorithms. Another approach involved grouping the received signals with the same observation interval outline and mining the hidden relationship between uncorrelated modulation classes. This enabled the modulation classification to be studied as a multiple-clustering problem, where the final modulation decision making is accomplished by evaluating the maximum-likelihood of multiple clusters corresponding to modulations in a given dataset [71]. The learning efficiency can be increased by recovering the centroids of all clusters by a constellation-structure-based reconstruction algorithm for parameter reduction with good convergence performance. Adaptive CSI estimation for modulation classification in MIMO systems was introduced by Abdul Salam *et al.* [73] to jointly exploit the Kalman filter (KF) and an adaptive interacting multiple model (IMM). The IMM–KF output was subsequently analyzed by a quasi-likelihood ratio test (QLRT) algorithm for modulation identification. It is worth noting that EM is derived for recursively computing estimates in IMM–KF and making decisions by the QLRT-based classifier.

Apart from likelihood-based approaches, numerous AMC methods that follow a typical ML framework with two principal steps, feature extraction and model learning, have been introduced. These feature-based methods mostly rely on sophisticated handcrafted feature engineering techniques for an improved description of radio characteristics, and conventional classifiers are employed for learning modulation patterns from extracted features. High-order cumulants (HOCs) of the amplitude, phase, real, and imaginary components of received signals were calculated for radio characteristic representation [80]. To flexibly accommodate different channel scenarios, linear SVM (LSVM) and the approximate maximum-likelihood (AML) algorithms were developed assuming that the channel condition is known, whereas a backpropagation neural network (BPNN) was considered for unknown channel conditions. Remarkably, phase and frequency offsets were estimated from high-order moments (HOMs) to enhance the performance of modulation classification in the unknown channel scenario. In another study, Huang *et al.* [81] performed independent component analysis (ICA) to select highly effective HOCs of arbitrary orders and lags. Furthermore, a maximum-likelihood-based multicumulant classification (MLMC) algorithm was proposed to identify the most appropriate modulation format by maximizing the posterior probability of the multicumulant vector. The elementary cumulant and cyclic cumulant calculated at the second and fourth orders were fused in a hierarchical hypothesis-based classification framework [82] to improve the system performance in terms of the classification rate for a flat fading channel. Despite being faster than conventional EM-based methods, this approach still requires huge amounts of computing resources for macro- and micro-classifiers in hierarchical association.

Hybrid approaches that combine the likelihood-based and feature-based classifiers have been proposed to achieve a good trade-off between accuracy and processing speed. For example, Abu-Ramoh *et al.* [83] derived a maximum-likelihood classifier capable of handling HOMs as features. Compared with conventional algorithms that manipulate the received sequence of modulation symbols, this approach can exploit few statistical moments to evaluate the likelihood function. As a result, the proposed AMC method induces lower complexity if a large number of modulations are considered for classification. This hybrid strategy for modulation classification was also extended [84] by leveraging a mixture of HOCs and HOMs at the second and fourth orders to design the maximum-likelihood classifier. Notably, a hierarchical classification framework is considered to improve the accuracy; however, the overall system complexity significantly increases in situations in which three binary classifiers are required to process four effortless modulations. Zhang *et al.* [85] built a dictionary set of high-order statistics for learning modulation patterns using block coordinate descent dictionary learning, where the modulation format of a signal is given by referring the sparse representation of statistical features to the dictionary.

Sophisticated feature extraction algorithms to more accurately describe the radio signal characteristics have recently

been proposed. To overcome the limitations of global features, including HOCs and HOMs, Xiong *et al.* [86] recommended two novel signal signatures: the first is modulation-specific transition features in the time domain, and the second is sequential features from the Fisher kernel. With Gaussian mixture model dictionary learning, exploitation of these local features results in an improvement of up to 30% in terms of classification accuracy compared to conventional approaches. Another approach involved capturing the normalized HOCs on the frequency domain by using discrete Fourier transform (DFT) for modulation classification in an OFDM system. Simulations were used to show that modulation classification strategies based on HOCs of DFT are more effective than those calculated in the time domain; however, the computational complexity significantly increases as the number of cumulant values increases [87].

### 2) INNOVATIVE DL-BASED AMC APPROACHES

Over the last few years, DL [88] has emerged as a vital ML tool for many applications ranging from natural language processing to vision recognition and bioinformatics. With many advantages such as automatic learning of high-level features and the effective exploitation of big data, DL has achieved remarkable success in many applications where it has become a core technique of model learning and pattern analysis. For AI-powered communications, DL is being exploited to address many challenging design tasks including network traffic control [25] and intelligent resource allocation [89]. For AMC, DNN [90]–[92] has been recommended to replace traditional classifiers for learning statistical features. For example, two sparse autoencoder-based DNNs were developed [90], [91] to improve the accuracy of high-order and intraclass digital modulations. Although their performance is slightly higher than that of LSVM and approximately maximum-likelihood classifiers, they are computationally more complex because of the requirement to compute a large number of neurons in hidden layers. Selection of the most relevant HOC features for learning a sparse autoencoder DNN [92] makes it possible to substantially reduce the overall complexity of the classifier without performance loss. LSTM, an advanced architecture of RNN that exploits the long-term dependencies between temporal attributes in sequential data, was further studied for modulation classification [93]. Three stacked LSTM layers configured in the underlying architecture allows the network to capture the temporal relation of in-phase and quadrature (IQ) samples while remaining flexible by accepting variable length input.

Among several DL architectures, CNN [75], [78], [79], [98] is more useful than DNN and RNN thanks to its ability to learn multiscale representational features from high-dimensional and unstructured data. In addition to releasing RadioML 2018.01A, a rich modulation classification dataset containing more than 2.5 million radio signals covering up to 24 analog and digital modulation formats in a wide range SNR $[-20 : 2 : 30]$ dB, O'Shea *et al.* [78] investigated the classification performance of two CNNs inspired by

VGG [99] and ResNet [100] originally proposed for image classification. Compared with the baseline approach, which calculates high-order statistics for learning an ensemble model of gradient boosted trees (XGBoost), the accuracy of these CNNs is significantly superior at different SNR levels under synthetic channel impairments, such as carrier frequency offset, symbol rate offset, delay spread, and additive noise. Notably, by exploiting skip connection in residual stacks, ResNet classifies modulations more precisely than VGG at high SNRs. In addition, the accuracy of these two CNNs is investigated under different parameter configurations, in particular, the number of convolutional layers (in VGG), the number of residual stacks (in ResNet), and the signal length (i.e., the number of IQ samples in a partitioned signal), to analyze the performance sensitivity. Meng *et al.* [79] introduced an end-to-end CNN, namely, CNN-AMC, for identifying the modulation of a long symbol-rate signal sequence, in which supplementary information in the form of the SNR is incorporated in fully connected layers via a concatenation operation to improve the accuracy. Even though CNN-AMC has the potential to obtain remarkable accuracy, the training and prediction processes have very high computational complexity because of the huge number of connections between a flatten layer and a fully connected layer. A compact-sized CNN, namely, VTCNN2 [75], was developed for cost-efficient modulation classification in edge devices, in which a pruning technique is applied to optimize the processing speed. This method allows the network to ignore the low-impact parameters (i.e., weight and bias) of the convolutional layers. Despite achieving a good tradeoff between accuracy and computing cost (measured by the number of floating point operations), the network size is still heavy because of the large number of trainable parameters distributed across the two fully connected layers. The pruning technique was further studied and led to the proposal of LightAMC [98], a CNN-based AMC method, to significantly accelerate the processing speed with negligible accuracy loss in IoT applications and unmanned aerial vehicle (UAV) systems.

Advanced CNN-based modulation classification methods have been recommended for performance enhancement by using a specialized novel structure of convolutional layers [5], [101] and using fusion mechanisms [74], [94]. An efficient CNN, namely, MCNet [5], was introduced for robust automatic modulation recognition under various channel impairments, in which the network architecture is specialized by several processing blocks associated via skip connection to prevent MCNet from experiencing a vanishing gradient and preserve the information identity by using many nonlinear operations. Moreover, to gain rich features and reduce the number of trainable parameters, each block in MCNet is configured by different one-dimensional asymmetric kernels (i.e., filter). Skip connection was also studied to design several specific blocks for feature learning in MBNet [102], Chain-Net [103], SCGNet [104], and RefNet [105]. Feature-level fusion and

decision-level fusion models, that is, early fusion and late fusion, were cleverly exploited in recent CNN-based modulation classification methods to counter channel deterioration. For example, Wang *et al.* introduced a decision-level fusion model for processing different incoming signals received by multiple antennas in a MIMO system, where a five-layer CNN performs the function of feature extraction in the proposed cooperative modulation classification method, namely Co-AMC [74]. The CNN induces the classification scores of MIMO signals that are cooperated via a weighted averaging decision rule to infer the final class of modulation. Moreover, a multilevel fusion architecture [94] was introduced with three fusion mechanisms (including feature-based, confidence-based, and majority voting-based fusion) to take advantage of meaningful information ranging from coarse to fine. These fusion models concurrently handle multiple CNN streams, in which each stream takes into consideration a fixed-length signal partitioned from a long sequence of IQ samples. Although the overall classification performance is improved, the computation is highly complex, including the computational cost and memory utilization. This prevents the potential application of this method for low-latency communication services.

Apart from processing IQ samples directly, several modulation classification methods have used deep models for the graphical presentations of signals, such as a constellation diagram [77], [95], [96], [106], [107] and spectrogram [76], [97]. Huang *et al.* [95], [96] designed a compressive CNN to learn the visual features of different modulation patterns from constellation diagrams. To improve the classification accuracy, both a regular constellation (RC) image (i.e., plotting the real and imaginary parts of the modulation signal as scattered points on a two-dimensional diagram) and contrast enhanced grid (i.e., RC image with probability distribution of scattered points) were jointly exploited via a fusion module specified in a single CNN. Furthermore, a synthetic loss function was formulated from cross-entropy loss, L2 regularization, and contrastive loss to maximize the difference between interclass features. Zeng *et al.* [97] used short-time discrete Fourier transform (STFT), a fundamental time-frequency analysis algorithm, for visualizing the spectrum of frequencies of the modulation signal. The set of transformed spectrogram images is then processed by a conventional CNN by adopting architecture with four convolutional layers for learning high-level features. Hybrid approaches [76], [77] were recommended for simultaneously using IQ data and image data. A hierarchical framework with two classifiers was proposed by Wang *et al.* [77] who used a dataset consisting of IQ samples for the first CNN-based classifier for inter-group modulation discrimination. A dataset consisting of constellation diagrams was leveraged for the second CNN-based classifier for intergroup modulation identification. In another study [76], a compact-sized two-branch CNN with two processing streams organized in parallel was proposed for identifying the modulation format of a signal. The meaningful

features that were independently extracted from the image representations of the constellation diagram and cyclic spectra were intensively fused at the end of the network for classification. Even though these methods based on constellation images or spectrogram images are more accurate compared with IQ-based approaches, they require more computational resources for data transformation, visualization, and storage. The CNNs of DL-based modulation classification approaches are described in Fig. 6, wherein most of them, typically designed to accept IQ data and image data as their input, are based on a simple network architecture (a straightforward connected structure of convolutional layers, activation layers, and pooling layers).

## C. SUMMARY AND TAKEAWAY POINTS

In this section, we reviewed state-of-the-art AMC methods, which are categorized as being either conventional (including likelihood-based and feature-based approaches) or innovative (including DL-based approaches), as summarized in Table 2. Most of the likelihood-based approaches are very computationally costly in terms of parameter estimation under unknown channel conditions, whereas the feature-based methods achieve moderate performance in terms of their classification rate because of the sensitivity of handcrafted statistical features and limited learning capacity of traditional classifiers. To overcome these limitations in conventional methods, researchers have exploited the excellent advantages of DL, such as automatic learning of high-level features and effective handling of big communication data to enhance the performance of modulation classification. Additionally, CNN-based approaches are applicable to numerous digital and analog modulations under different channel impairments, such as a flat fading channel, multipath fading channel with attenuation, and additive noise, as summarized in Table 3. Interestingly, a DNN not only accepts a sequence of IQ samples (partitioned with a fixed length) as its input, but also accepts other transformed data (for instance, a constellation diagram in the form of a scattered plot and a spectrogram image via time-frequency analysis). Other than a few notable CNN models, such as ResNet [78] and MCNet [5], which were introduced for discriminating 24 challenging modulations, several other neural networks do not effectively leverage the powerful learning capability of CNN for effortless classification, for example, Co-AMC [74] processes four low-order digital modulations based on a huge dataset containing more than 1.3 million samples. Despite the superiority of deep learning over conventional approaches, certain aspects deserve further investigation when developing a DL model for modulation classification:

- Deploying many fully connected layers without global average pooling [79] can rapidly increase the network size (usually measured by the number of trained parameters), leading to extremely high computational complexity.
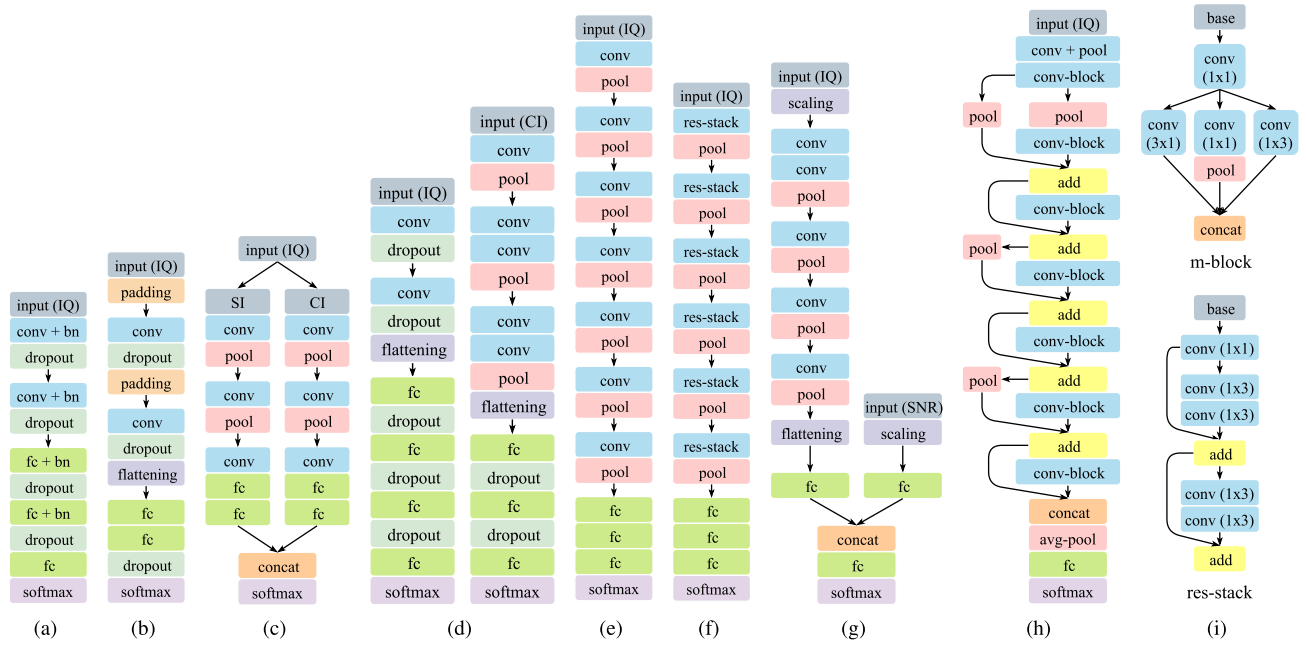
**FIGURE 6.** CNN architectures of several state-of-the-art modulation classification approaches: (a) Co-AMC [74], (b) VTCNN2 [75], (c) Two-branch CNN [76], (d) CNN1 (for IQ data) and CNN2 (for CI data) in a hierarchical classifier [77], (e) VGG [78], (f) ResNet [78], (g) CNN-AMC [79] with a supplementary input of SNR values, MCNet [5], and (i) the structures of m-block and res-stack in MCNet and ResNet. The input of these networks can be the signal sequence of IQ samples, an image of the constellation diagram (CI), and an image of the spectrogram (SI). Notations presented in figures: conv (convolutional layer), bn (batch normalization layer), pool (max-pooling layer), avg-pool (average pooling layer), concat (depthwise concatenation layer), fc (fully connected layer or dense layer), add (elementwise addition layer). Customized layers are defined as follows: padding layer for adding zero values to input borders, flattening layer for collapsing the spatial dimensions of input to the channel dimension (or the depth dimension of feature maps), scaling layer for normalizing data values into a specific range. Each conv is followed by an activation layer such as ReLU, leaky ReLU, and exponential linear unit (ELU).

- Configuring kernels of various sizes [5], such as unit $1 \times 1$, symmetric $n \times n$, and asymmetric $1 \times n$, potentially enriches the representational feature maps.
- Sophisticated techniques such as skip connection and dropping out can be leveraged to prevent the network from experiencing vanishing gradient descent and overfitting.
- Deep fusion frameworks with multiple processing streams to process different types of input data [76] are recommended to more effectively learn intrinsic radio characteristics.
- Balancing the accuracy and computational cost should be an important design objective to meet the requirements of modern communication services.

## IV. SIGNAL DETECTION

In this section, we discuss applications of AI techniques for intelligent signal detection.

### A. FUNDAMENTALS OF SIGNAL DETECTION

In MIMO communication systems with $N$ transmit and $M$ receive antennas, the received baseband signal can be expressed as

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{n}, \tag{7}$$

where $\mathbf{s} = [s_1, s_2, \ldots, s_N]^T$ and $\mathbf{y} = [y_1, y_2, \ldots, y_M]^T$ represent the transmitted and received signal vectors, respectively,

with $(\cdot)^T$ denoting the transpose of a vector. Further, $\mathbf{H}$ of size $M \times N$ represents the channel matrix between the transmitter and receiver, and $\mathbf{n}$ is a Gaussian noise vector. The goal of signal detection is to determine $\mathbf{s}$ from $\mathbf{y}$. This can be achieved via classical detection schemes such as the optimal maximum likelihood, near-optimal sphere decoding (SD) [108], tabu search (TS) [109], [110], suboptimal linear zero-forcing (ZF), minimum mean square error (MMSE), and successive interference cancellation (SIC) receivers. Furthermore, interest in the development of ML-based detectors (MLDs) has recently been growing. In the following subsections, we review selected fundamental classical detection schemes and discuss ML-based approaches for signal detection.

### 1) OPTIMAL MAXIMUM-LIKELIHOOD DETECTOR

The optimal maximum-likelihood solution is obtained by an exhaustive search as follows:

$$\hat{\mathbf{s}}_{\text{opt}} = \arg \min_{\mathbf{s} \in \mathcal{A}^N} \|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2, \tag{8}$$

where $\mathcal{A}$ is an alphabet containing all possible transmitted signals $s_n, n = 1, \ldots, N$. The computational complexity of the maximum-likelihood detector increases exponentially with $N$, which is prohibitive even for a small value of $N$. To overcome this challenge, near-optimal reduced-complexity detection schemes have been proposed, such as SD [108] and TS [109], [110].

**TABLE 2.** Summary of state-of-the-art AMC methods for communication systems.

| Category | Year | Paper | Highlights | Limitations |
|---|---|---|---|---|
| Likelihood-based | 2017 | [67] | Decoupling the interactive multivariate maximum-likelihood problem into multiple separated optimization problems. | Expensive computational complexity of multiple optimizations. |
| | | [68] | Improving [67] with CRLBs of the joint unknown estimates. | Conventional accuracy on few effortless modulations. |
| | 2018 | [69] | Formulating continuous phase modulation signals as HMM variables. Applying BW for unknown fading estimation. | Forward–backward algorithm in HMM is more expensive. |
| | | [70] | Two classifiers based on ALRT and HLRT for known/unknown CSI scenarios. | Low accuracy of blind AMC (without CSI information). Poor trade-off between classification rate and computational cost. |
| | | [71] | Clustering received modulation signals of same observation characteristics. Constellation-structure-based centroid reconstruction. | System complexity progressively increases along the number of modulations. |
| | 2019 | [72] | Squared extrapolation method with a parameter checking scheme | Insubstantial classification at low SNRs. |
| | | [73] | Estimating CSI robustly via an IMM–KF model. QLRT-based classifier. | Performance is sensitive to the parameter initialization |
| Feature-based | 2017 | [80] | LSVM and AML algorithms for a known channel condition. BPNN for an unknown channel condition. HOMs for phase and frequency offsets estimation. | Conventional performance of blind modulation classification (without channel information). |
| | | [81] | MLMC classifies modulation based on posterior probability. ICA for HOC feature selection. | Feature efficiency strongly depends upon the quality of channel estimation. |
| | | [82] | Hierarchical hypothesis-based classification framework. Fusing elementary and cyclic cumulants. | High computational complexity. |
| | 2018 | [83] | HOMs-based maximum-likelihood classification algorithm. | Expensive cost if considering few modulations. |
| | | [84] | Mixture of HOMs and HOCs for likelihood-based hierarchical classification framework. | Complicated classification framework with three binary classifiers. |
| | | [85] | Organizing a dictionary of high-order statistics in sparse representation. | High memory consumption for dictionary ensemble. |
| | 2019 | [86] | Local transition features and Fisher-based sequential features. Gaussian mixture model dictionary learning. | Accuracy is sensitive to the dictionary size specified in dictionary initialization. |
| | | [87] | Normalizing HOCs in the frequency domain via DFT. | Computing the DFT is computationally expensive. |
| DL-based | 2017 | [91] | $k$-Sparse autoencoder for low-complexity input reconstruction with DNN. | Requires a large number of symbols to train the classification model. |
| | 2018 | [78] | Introducing the RadioML 2018.01A dataset of modulation classification. Analyzing the performance of VGG and ResNet for modulation classification. | Following network backbones for image classification without architecture fine tuning. |
| | | [79] | CNN-AMC for processing long symbol-rate signals. | Requires SNR information. |
| | 2019 | [90] | Cooperative classifier with radial basic function network (RBFN) and sparse autoencoder DNN. | Initializing many neurons in hidden layers inducing a heavy load. |
| | | [94] | Introducing a CNN-based multilevel fusion architecture for effectively learning intrinsic information from coarse to fine. | High computation and memory consumption for processing multiple CNN streams. |
| | | [95] | Plotting IQ samples into a scattered diagram for constellation image with contrast enhancement. | Additional computing resource for data transformation. Performance is overly sensitive to the output size (a.k.a. resolution) of constellation and spectrogram images. |
| | | [96] | Formulation of a synthetic loss function with contrastive loss. | |
| | | [97] | Transformation of IQ data to spectrogram image by STFT. | |
| | | [77] | Hierarchical classification with two CNNs for handling IQ data and constellation image. | |
| | | [76] | A compact-sized two-stream CNN to simultaneously learn the constellation diagram and cyclic spectra. | |
| | 2020 | [92] | HOC feature selection for learning sparse autoencoder DNN. | Taking into account a limited number of modulations. |
| | | [93] | Deploying LSTM for learning long-term dependencies of IQ samples. | Poor accuracy when the channel condition is unknown. Extremely high training complexity. |
| | | [75] | Cost-efficiency CNN-based modulation classifiers. | Only applicable to convolutional layers. |
| | | [98] | Accelerating the processing speed with pruning technique. | |
| | | [5] | Introducing MCNet with multiple associated blocks. Each block is specified by asymmetric kernels. | High-order modulation classification is less robust. |
| | | [74] | Cooperative CNN-based approach for MIMO system with a weighted averaging decision rule. | Conventional accuracy of four-modulation classification under non-channel impairment condition. |

## 2) LINEAR DETECTORS

In linear detectors, the discrete alphabet $\mathcal{A}$ is relaxed to a continuous space, allowing closed-form solutions of (8) to be found by solving the nonconstrained convex optimization problem min $\|\boldsymbol{y} - \boldsymbol{Hs}\|^2$. The obtained solution is then quantized to the nearest vector in $\mathcal{A}^N$ [111]. The ZF and MMSE receivers are two typical linear detectors, the solutions of which are given by

$$\hat{\boldsymbol{s}}_{\text{ZF}} = \mathcal{Q}\left((\boldsymbol{H}^H\boldsymbol{H})^{-1}\boldsymbol{H}^H\boldsymbol{y}\right), \qquad (9)$$

$$\hat{\boldsymbol{s}}_{\text{MMSE}} = \mathcal{Q}\left((\boldsymbol{H}^H\boldsymbol{H} + \sigma^2\boldsymbol{I})^{-1}\boldsymbol{H}^H\boldsymbol{y}\right), \qquad (10)$$

respectively, where $\sigma^2$ is the variance of Gaussian noise, and $\mathcal{Q}(\cdot)$ is the element-wise quantization operator that

**TABLE 3.** Summary of simulation configuration of DL-based modulation classification approaches.

| Paper | Model | Channel impairments | | | No. modes | SNR (dB) | Dataset remarks |
|-------|-------|------|-----------|------|-----------|----------|-----------------|
| | | Flat | Multipath | AWGN | | | |
| [91] | DNN | ✓ | | ✓ | 4 | $[0:1:15]$ | 2000 1000-symbol signals of {4PSK, 16PSK, 16QAM, 128QAM}. |
| [78] | CNN | | ✓ | ✓ | 24 | $[-20:2:30]$ | RadioML 2018: 2,555,904 IQ signals of {BPSK, QPSK, 8PSK, 16PSK, 32PSK, 16APSK, 32APSK, 64APSK, 128APSK, 4ASK, 8ASK, 16QAM, 32QAM, 64QAM, 128QAM, 256QAM, OOK, GMSP, OQPSK, FM, AM-SSB-WC, AM-SSB-SC, AM-DSB-WC, AM-DSB-SC}. |
| [79] | CNN | ✓ | | ✓ | 7 | $[-6:2:10]$ | 341,000 IQ signals of {BPSK, 4PSK, 8PSK, 16QAM, 16APSK, 32APSK, 64QAM}. |
| [90] | DNN | | ✓ | ✓ | 4 | $[0:1:15]$ | 96,000 IQ signals of {GMSK, GFSK, CPFSK,OQPSK}. |
| [94] | CNN | | ✓ | ✓ | 12 | $[-20:2:30]$ | 624,000 IQ signals of {BPSK, QPSK, 8PSK, OQPSK, 2FSK, 4FSK, 8FSK, 16QAM, 32QAM, 64QAM, 4PAM, 8PAM}. |
| [95] | CNN | | | ✓ | 5 | $[-5:2:15]$ | 2,750,000 CIs of {BPSK, QPSK, 8PSK, 16QAM, 64QAM}. |
| [96] | CNN | | | ✓ | 5 | $[-5:2:15]$ | 1,9250,000 CIs of {BPSK, QPSK, 8PSK, 16QAM, 64QAM}. |
| [97] | CNN | | ✓ | ✓ | 11 | $[-20:2:18]$ | RadioML 2016: 220,000 SIs of {BPSK, QPSK, 8PSK, 16QAM, BFSK, CPFSK, PAM4,WB-FM,AM-SSB,AM-DSB}. |
| [77] | CNN | | ✓ | ✓ | 8 | $[-8:2:18]$ | 84,000 IQ samples and 84,000 CIs of {BPSK, QPSK, 8PSK, GFSK, CPFSK, PAM4, 16QAM, 64QAM}. |
| [76] | CNN | | ✓ | ✓ | 11 | $[-20:2:18]$ | RadioML 2016 |
| [93] | LSTM | | ✓ | ✓ | 4 | $[0:2:20]$ | 440,000 IQ signals of {BPSK, QPSK, 8PSK, 16QAM}. |
| [75] | CNN | | ✓ | ✓ | 11 | $[-20:2:18]$ | RadioML 2016 |
| [98] | CNN | ✓ | | ✓ | 4 | $[-10:1:10]$ | 505,000 IQ signals of {BPSK, QPSK, 8PSK, 16QAM}. |
| [5] | CNN | | ✓ | ✓ | 24 | $[-20:2:18]$ | RadioML 2018 |
| [74] | CNN | ✓ | | ✓ | 4 | $[-10:2:10]$ | 1,320,000 IQ signals of {BPSK, QPSK, 8PSK, 16QAM}. |

**Modulation abbreviation**

PSK: Phase-shift keying
BPSK: Binary phase-shift keying
QPSK: Quadrature phase-shift keying
OQPSK: Offset quadrature phase-shift keying
CPFSK: Continuous phase frequency-shift keying
SSB-WC: Single-sideband modulation with carrier
DSB-WC: Double-sideband modulation with carrier
SSB-SC: Single-sideband suppressed-carrier modulation
DSB-DC: Double-sideband suppressed-carrier modulation

ASK: Amplitude-shift keying
FSK: Frequency-shift keying
OOK: On–off keying
GFSK: Gaussian frequency-shift keying
GMSK: Gaussian minimum-shift keying
AM: Amplitude modulation
QAM: Quadrature amplitude modulation
PAM: Pulse-amplitude modulation
WB-FM: Wide band frequency modulation

quantizes elements in $(\cdot)$ to the nearest elements in $\mathcal{A}$. The ZF detector performs poorly in the case of ill-conditioned channels due to noise enhancement. By contrast, the MMSE detector reduces noise enhancement and attains improved performance with respect to ZF. Both the ZF and MMSE receivers have low computational complexity. However, their performance is far from optimal, especially in square systems, i.e., when $N \approx M$.

### 3) MLD
The key idea of MLD is to model and train an ML algorithm such that its output $\hat{s}_{\text{ML}}$ can approximate the transmitted signal vector $s$ with high accuracy. In general, an ML-based solution for signal detection can be formulated as

$$\hat{s}_{\text{ML}} = \mathcal{Q}\left(\Pi(x, \mathcal{P})\right), \qquad (11)$$

which represents a nonlinear transformation with the input vector $x$ and the trainable parameter set $\mathcal{P}$, followed by quantization. It is observed from (11) that the performance of an MLD depends on the input signal vector $x$, nonlinear function $\Pi$, and the learnable parameter set $\mathcal{P}$. In particular, $x$ contains information about the received signals and CSI if available. Furthermore, the nonlinear transformation $\Pi$ and trainable set $\mathcal{P}$ are determined by the underlying ML model

and training process, which are the deciding factor for the learning ability and the accuracy of the ML model. These configurations, which result in various MLDs with different performance and computational complexity, are reviewed in the next subsection.

### B. STATE-OF-THE-ART MLDs
Various studies have considered the application of ML to signal detection, leading to numerous MLDs. Using different ML tools, a training process, and CSI models, existing MLDs can be classified as follows:

- *ML tools for signal detection*: Various ML techniques have been considered for signal detection. Among them, DNN is the most widely used owing to its powerful learning capability [112]–[127]. However, its computational cost and energy consumption are generally high [115] owing to its large number of neurons and layers, especially in the case of those developed for large-scale systems. Other ML tools, such as CNN [116], [122], [128]–[130], recurrent NN (RNN) [116], [122], extreme learning machine (ELM) [131], [132], auto encoder (AE) [131], [133], and ensemble learning [134], were also leveraged for signal detection.
- *CSI requirement*: In classical signal detection schemes, CSI is crucial for obtaining the estimate of the

transmitted signal, as seen in (8)–(10). However, it becomes optional in MLDs. Specifically, while both the received signal $y$ and CSI, i.e., $H$, are taken as the input of ML algorithms in [112]–[115], [117]–[120], [125], [126], [130], [135], only $y$ is required for the MLDs in [116], [121], [122], [124], [127], [129], [131]–[133]. The omission of CSI can simplify the communication system, in which the channel estimation block is removed, or it can reduce the size of the input signal vector. As a result, a considerable reduction in overall computational complexity as well as power consumption can be attained. However, this could result in potential performance degradation in some scenarios, especially in block-fading channels [116] and in multiple-antenna systems, where the CSI is crucial for removing the intersymbol interference.

- *Training approaches*: Unlike the classical methods in which a hand-engineered detection scheme is applied in an online method, an MLD needs to train an ML model before it is used for signal detection. For example, in a DNN-based detector, the weights and biases of the DNN are trained to minimize the distance between the ML-based solution and the labels, i.e., $\hat{s}_{\mathrm{ML}}$ and $s$ [11], [49], [118], [120]. In particular, the training can be carried out either offline [115]–[120], [122]–[130], [133], [135] or online [114], [121], [132]. In offline training, the computational complexity of the training process can generally be ignored, and the parameters of an ML model can be readily optimized by using a sufficiently large amount of training data. However, this training method is only suitable for certain channel models such as an independent and identically distributed (i.i.d.) Rayleigh fading channel. In contrast, the method may become less impractical in real-world communication systems, where the channel characteristics change rapidly or the channel statistics are unavailable, e.g., in molecular communication [122]. In these scenarios, a good solution would be to apply an online training method (e.g., [114], [121], [132], at the cost of increased latency and computational complexity.

The aspects listed above distinguish existing MLDs based on their configurations. However, we found it to be more beneficial to classify MLDs based on the way ML techniques are leveraged for signal detection. This is useful not only for analyzing and synthesizing existing MLDs, but also for providing methodologies for further development in this area. Our literature review shows that ML techniques can be leveraged for signal detection in three ways, namely, black-box, unfolding, and classical detector-based MLDs. First, an ML tool can be modeled as a *black-box MLD*, i.e., it outputs the estimate of the transmitted signal vector $s$ as an independent detector. Second, in *unfolding MLDs*, each layer of the DNN is constructed based on the operations in each iteration of the classical projected gradient descent (PGD) algorithms. Finally, existing iterative or near-optimal detection algorithms can be further optimized by using ML,

resulting in so-called *classical detector-based MLDs*. These three groups of MLDs are discussed in the following.

### 1) BLACK-BOX MLDs

Inspired by the learning capability of ML techniques, certain detector designs attempted to replace the classical detectors with black-box MLDs [115]–[117], [122], [124], [125], [127], [132], [133]. We note that in this work, the term *black-box* reflects the fact that an ML model can learn and output the desired solution, which is $\hat{s}_{\mathrm{ML}}$ for signal detection, without any expert knowledge. The ingenuity of this application lies in choosing an appropriate ML tool and optimizing the model configurations, e.g., the number of layers, number of nodes in each layer, and activation functions for the underlying DNN.

DNNs were used for black-box MLDs in OFDM systems [125], [127]. Unlike the classical OFDM receivers that first require conducting CSI estimation, the DNN-based detectors proposed in [125] and [127] perform the signal detection directly. In other words, the DNN is a black box that takes $y$ as input and outputs $\hat{s}_{\mathrm{ML}}$, without requiring the estimated CSI. Notably, these black-box MLDs are shown to outperform the conventional least-square (LS) and MMSE receivers, which compute the estimates of transmitted signals based on CSI, as shown in (10). An end-to-end OFDM communication system was modeled as a single AE [133], in which the DNN-based detector was trained offline irrespective of the channel. The performance of the black-box MLD was also investigated with the use of ELM [132], RNNs, and CNNs [122]. In particular, the ELM-based black-box MLD in [132] outperformed the DNN-based MLDs proposed in [125], with less complexity and excellent robustness under different multipath fading channels. Furthermore, the CNN- and RNN-based black-box MLDs in [122] were evaluated using experimental data collected by a chemical communication platform, for which the channel model was unknown and which was difficult to model analytically.

A common observation from the studies discussed above is that the developed black-box MLDs can perform well without the CSI. This is advantageous compared with the classical detectors when the CSI is either not available or the wireless channel is fast fading. Furthermore, the black-box MLDs achieve not only improved performance, as shown in [122], [127], [132], but also reduced complexity with respect to the simple linear receivers [115]. The reason for this reduction in complexity is that these black-box MLDs only perform matrix multiplications and additions, whereas computationally expensive matrix inversions or factorizations (such as singular-value decomposition or QR decomposition) are required for most conventional MIMO receivers including linear ZF, MMSE, and SIC. However, it is worth noting that most of these black-box MLDs are proposed for general OFDM systems. By contrast, the signal detection in multiple-antenna systems requires more sophisticated MLDs, which are discussed in the next subsections.

## 2) UNFOLDING MLDs

While various ML tools are leveraged for black-box MLDs, unfolding MLDs employ DNNs because they can learn complicated nonlinear functions. However, the ingenuity of detectors in this group is that, instead of directly using the well-known fully connected DNN (FC-DNN), they employ unfolding layer-by-layer architectures. In these architectures, the layers have the same structure and are constructed following the classical PGD algorithm with different weights because of their different input signals.

Intuitively, (8) can be solved by the iterative PGD optimization method. Motivated by this, a series of unfolding MLDs, including the *detection network* (DetNet) [49], [118], *sparsely-connected DNN* (ScNet) [120], *fast-convergence sparsely connected DNN* (FS-Net) [11], *multilayer DNN* (Twin-DNN) [119], and *Cascade DNN* (Cascade-Net) [135] were proposed for signal detection. In these schemes, $\hat{s}_{\mathrm{ML}}$ is updated over $L$ layers of the DNN as follows:

$$
\begin{aligned}
\hat{s}^{[l]} &= f\left( s - \delta^{[l]} \frac{\partial \|y - Hs\|^2}{\partial s} \right)_{s=\hat{s}^{[l-1]}} \\
&= f\left( \hat{s}^{[l]} - \delta^{[l]} H^T y + \delta^{[l]} H^T H \hat{s}^{[l-1]} \right),
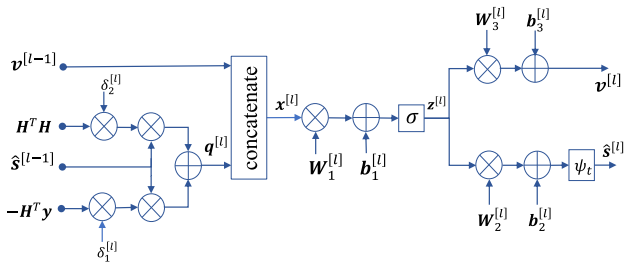\end{aligned}
\tag{12}
$$



**FIGURE 7.** Illustration of layer *l* of the DetNet [11], [49], [118].

where $f(\cdot)$ denotes a nonlinear projection operator, $\delta^{[l]}$ is the step size, and $l = 1, \ldots, L$. Inspired by (12), DetNet was introduced [49], [118]. The *l*th layer of DetNet is illustrated in Fig. 7, and its operations are summarized as follows:

$$
q^{[l]} = \hat{s}^{[l-1]} - \delta_1^{[l]} H^T y + \delta_2^{[l]} H^T H \hat{s}^{[l-1]},
\tag{13}
$$

$$
x^{[l]} = \left[ v^{[l-1]}, q^{[l]} \right]^T,
\tag{14}
$$

$$
z^{[l]} = \sigma\left( W_1^{[l]} x^{[l]} + b_1^{[l]} \right),
\tag{15}
$$

$$
\hat{s}^{[l]} = \psi_t\left( W_2^{[l]} z^{[l]} + b_2^{[l]} \right),
\tag{16}
$$

$$
v^{[l]} = W_3^{[l]} z^{[l]} + b_3^{[l]},
\tag{17}
$$

where $\hat{s}^{[0]} = v^{[0]} = \mathbf{0}$, with $\mathbf{0}$ being an all-zero vector of an appropriate size, and $\{W_1^{[l]}, W_2^{[l]}, W_3^{[l]}, b_1^{[l]}, b_2^{[l]}, b_3^{[l]}, \delta_1^{[l]}, \delta_2^{[l]}\}$ are the training parameters, including the weights, biases, and step size, in the *l*th layer of DetNet. Furthermore, $\sigma(\cdot)$ represents the rectified linear unit (ReLU) activation function, and $\psi_t(\cdot)$ guarantees that the amplitudes of the elements of $\hat{s}^{[l]}$ are in an appropriate range of desired signals.

The final solution of DetNet is $\hat{s}_{\mathrm{ML}} = \mathcal{Q}\left( \hat{s}^{[L]} \right)$. Although DetNet achieves promising performance, it has several drawbacks. Specifically, the significance of the intermediate signal vector $v^{[l]}$ is not clear and considerably enlarges the size of the input vector $x^{[l]}$, thereby complicating the network architecture. Consequently, the computational complexity of DetNet is extremely high. Furthermore, although the performance of DetNet is shown to be good for the case $N \ll M$, subsequent work [11], [120] showed the network performance to be far from optimal for square systems, i.e., $N \approx M$.

ScNet [120] and FS-Net [11] were proposed to overcome the drawbacks of DetNet. ScNet focuses on simplifying the network architecture of DetNet by removing the intermediate vector $v^{[l]}$ and redundant connections in the network, whereas FS-Net reduces the number of training parameters and optimizes the loss function to accelerate the convergence of DetNet and ScNet. Although the network architecture of DetNet was significantly simplified, ScNet and FSNet improved the performance remarkably. Specifically, an improvement of approximately 2-dB and 3-dB in SNR was achieved by ScNet and FS-Net, respectively, with respect to DetNet, with the computational complexity being approximately 40% of that of DetNet [11]. Unlike ScNet and FS-Net, Twin-DNN [119] uses two parallel DetNets with different input vectors $\hat{s}^{[0]}$: the first is $\hat{s}_{\mathrm{ZF}}$ and the other is randomly generated. The solution of Twin-DNN is the more accurate of the two output vectors of the two DetNets, motivated by ensemble learning. As a result, Twin-DNN improves the performance at the cost of approximately double the complexity of DetNet. A detailed comparison of DetNet, ScNet, FSNet, and Twin-DNN in terms of their network architecture, performance, and computational complexity was reported [11]. Another variant of DetNet is Cascade-Net, which was proposed for single-antenna systems [135]. In Cascade-Net, a DNN is cascaded with a ZF preprocessor to prevent the network from converging to a saddle point or local minimum point. Simulation results in [135] show that Cascade-Net performs much better than DetNet and the classical ZF detector.

## 3) CLASSICAL DETECTOR-BASED MLDs

Although the black-box and unfolding MLDs discussed above outperform simple detectors, their performance is still far from optimal, especially in challenging scenarios such as square systems with high-order QAM signaling. To overcome this limitation, we can incorporate ML algorithms with classical hand-engineered detectors such as iterative or near-optimal detection schemes. This incorporation is useful to further optimize the classical detection schemes in terms of their computational complexity and/or performance.

### a: ITERATIVE ALGORITHM-BASED MLDs

He *et al.* [112] proposed an iterative orthogonal approximate message passing (OAMP) algorithm-based network (OAMP-Net). In the *l*th layer of OAMP-Net, the output $\hat{s}^{[l]}$ is

updated as follows:

$$z^{[l]} = \hat{s}^{[l-1]} + \gamma^{[l]} A^{[l]}(y - H z^{[l-1]}), \quad (18)$$

$$s^{[l]} = \mathbb{E}\left\{ s | z^{[l]}, \tau^{[l]} \right\}, \quad (19)$$

where (19) represents the MMSE denoiser, and

$$A^{[l]} = \frac{N v^{[l]^2} H^H (v^{[l]^2} H H^H + \sigma^2 I)^{-1}}{\text{trace}(v^{[l]^2} H^H (v^{[l]^2} H H^H + \sigma^2 I)^{-1} H)}, \quad (20)$$

$$v^{[l]^2} = \frac{\left\| y - H \hat{s}^{[l]} \right\|^2 - M \sigma^2}{\text{trace}(H^H H)},$$

$$\tau^{[l]^2} = \frac{1}{2N} \text{trace}((I - A^{[l]})(I - A^{[l]})^H) v^{[l]^2}$$
$$+ \frac{\theta^{[l]} \sigma^2}{4N} \text{trace}(A^{[l]} A^{[l]^H}), \quad (21)$$

with $s^{[0]} = 0$ and $\tau^{[0]} = 1$. We note that in the classical OAMP scheme, $\gamma^{[l]}$ and $\theta^{[l]}$ in (18) and (21), respectively, are both set to one. By contrast, they emerged as learnable variables and were optimized by using training to provide appropriate step sizes for updating the mean and variance of the MMSE denoiser. Compared to DetNet, OAMP-Net is easier and faster to train because only a few adjustable parameters need to be optimized. Furthermore, OAMP-Net achieves an SNR improvement of approximately 2-dB with respect to the classical OAMP scheme.

More recently, OAMP-Net2 was proposed [113] as an improved version of OAMP-Net. Specifically, two additional learnable parameters were added to the denoiser (19) in OAMP-Net2 to construct the nonlinear estimator of $s^{[l]}$ to satisfy the divergence-free requirement. This update improves the SNR of OAMP-Net2 with approximately 3-dB with respect to OAMP-Net [113]. Furthermore, OAMP-Net2 was demonstrated to outperform prior detection schemes such as DetNet and MMSE-based SIC. Although OAMP-Nets perform impressively for i.i.d. Gaussian channels, it is shown in [112] and [114] that they may not perform very well for realistic channels with spatial correlations [114]. Moreover, OAMP-Nets require performing matrix inversion in each iteration, making them even more computationally expensive than DetNet [113]. To overcome these limitations of OAMP-Nets, Khani *et al.* proposed the MMNet [114]. Similar to OAMP-Nets, MMNet follows an iterative approach. However, instead of being computed as in (20), $A^{[l]}$ is considered as an $N \times M$ trainable weight matrix. This is more advantageous than OAMP-Nets in the following respects. First, it allows $z^{[l]}$ to be obtained with flexible trainable parameters optimized for each channel realization, thereby facilitating online training to adapt to channel variation. Second, $A^{[l]}$ can be obtained without performing computationally expensive matrix inversion as in (20) for OAMP-Nets. The simulation results in [114] showed that, in practical 3GPP channels, MMNet achieves an improvement of 3-dB in SNR compared to OAMP-Nets with less computational complexity by a factor of $10 - 15$.

### b: NEAR-OPTIMAL DETECTOR-BASED MLDs

The two well-known near-optimal detection schemes are TS and SD. The DL-aided TS algorithm was introduced in [11]. Specifically, Nguyen *et al.* proposed employing FS-Net to generate the highly reliable initial solution of the TS scheme. Furthermore, in this algorithm, an adaptive early termination and a modified searching process are performed based on the predicted approximation error, which is determined from the FS-Net-based initial solution, to enable the final solution to be reached earlier. The simulation results in [11] demonstrated that the proposed DL-aided TS algorithm reduces the complexity by approximately 90% with respect to the existing classical TS algorithms, while maintaining almost the same performance.

DL-aided SD schemes were recently proposed [12], [136]–[138]. Specifically, a DNN was used to learn the initial radius for SD [12], [136], [137]. While a single radius is used in [12], multiple radii are employed in [136]. Furthermore, as an improvement of existing schemes [12], [136], Weon and Lee [137] proposed a learning-aided deep path prediction scheme for sphere decoding in large multiple-antenna systems. In particular, the minimum radius for each sub-tree is learned by a DNN, resulting in a more significant complexity reduction with respect to the prior DL-aided SD schemes in [12] and [136]. In all three DL-aided SD schemes mentioned above, the common idea is to predict radii for the sequential SD. This approach has certain limitations in the offline learning phase, as well as during online application to SD. First, in the DNN training phase [12], [136], [137], conventional SD needs to be performed first to generate training labels, i.e., the radii. Consequently, these DNNs are time-consuming and computationally complex to train, especially in the case of large MIMO systems. Second, although the radius plays an important role in the search efficiency of conventional Fincke–Pohst SD, it becomes less significant in the Schnorr–Euchner SD, especially for high SNRs, for which a relatively reliable radius can be computed using the conventional formula [108]. To overcome these limitations, the fast DL-aided SD (FDL-SD) and fast DL-aided $K$-best SD (KSD) (FDL-KSD) algorithms were proposed [138]. The idea of FDL-SD and FDL-KSD is to use FS-Net [11] to generate a highly reliable initial candidate for the search in SD/KSD, which facilitates a candidate/layer-ordering scheme and an early rejection scheme to significantly reduce the complexity of the conventional SD schemes. The simulation results in [138] showed that, for moderate SNRs, FDL-SD reduces the complexity by more than 90% with respect to the classical Fincke–Pohst SD scheme, compared with those of the DL-aided SD schemes [12], [136], [137], which achieve a reduction of approximately 60%.

### C. SUMMARY AND TAKEAWAY POINTS

In this section, we reviewed the typical existing MLDs in the literature. Based on the application of ML models to the detection process, they are divided into three groups: *black-box MLDs*, *unfolding MLDs*, and *classical*

**TABLE 4.** Summary of existing MLDs.

| MLD group | Detector | Paper | ML models | Interesting observations |
|---|---|---|---|---|
| Black-box MLDs | ML model | [115], [116], [117], [122], [124], [125], [133], [132], [127] | DNN, CNN, RNN, ELM, AE | • No CSI requirement<br>• Suitable for fast varying channels or unavailable CSI<br>• Outperforms linear receivers<br>• Matrix inversion and factorization are unnecessary<br>• Generally proposed for OFDM rather than for MIMO systems |
| Unfolding MLDs | Projected GD-based | [49], [118] | DetNet | • Performs well for $N \ll M$, BPSK, QPSK, and i.i.d. Rayleigh fading channels<br>• Performance degrades for $N \approx M$ and higher-order modulation<br>• High computational complexity caused by complicated network architecture and use of an intermediate input vector<br>• Difficult to train because of many learning variables |
| | | [120] | ScNet | • Simplifies the network connections of DetNet<br>• Improved performance, reduced complexity, and fewer learning variables with respect to DetNet |
| | | [11] | FS-Net | • Simplifies the input vector and network connection, and optimizes the loss function of ScNet and DetNet<br>• Improved performance, reduced complexity, and fewer learning variables with respect to DetNet and ScNet |
| | | [119] | Twin-DNN | • Improved performance with respect to DetNet<br>• Much higher complexity than DetNet owing to the requirement for two parallel DetNets and a ZF solution |
| | | [135] | cascade-Net | • Outperforms both DetNet and the classical ZF detectors |
| Classical detector-based MLDs | Iterative algorithm | [112] | OAMP-Net | • Easier and faster to train compared with DetNet<br>• Improved performance with respect to the classical OAMP scheme<br>• Performance degradation in correlated channels |
| | | [113] | OAMP-Net2 | • Outperforms DetNet, OAMP-Net, and classical MMSE-based SIC |
| | | [114] | MMNet | • Generalizes well to correlated channels<br>• Outperforms OAMP-Nets with lower complexity |
| | TS | [11] | FS-Net | • Leverages the FS-Net solution to generate an initial solution, modify the search procedure, and terminate the TS algorithm early<br>• 90% complexity reduction with almost no performance loss with respect to the classical TS schemes |
| | SD | [12], [136], [137] | FC-DNN | • Use FC-DNNs to predict initial radii for SD<br>• Complexity reduction of approximately 60% at moderate SNRs with marginal performance loss<br>• Computationally expensive and time-consuming training phase for performing the classical SD scheme to collect training data |
| | | [138] | FS-Net | • Uses FS-Net to generate initial candidate, facilitating candidate ordering, layer ordering, and early pruning in SD/KSD schemes<br>• Complexity reduction of 90% at moderate SNRs with no performance loss<br>• Unnecessary to perform the classical SD to collect training data |

*detector-based MLDs*, as summarized in Table 4. In the first group, ML tools are used to design independent detectors, and they are trained to learn the transmitted signal vector $s$ without requiring any expert knowledge. Various ML models, including DNN, CNN, RNN, ELM, AE, and ensemble learning are used for black-box MLDs. DNNs are the most widely used for the other two groups of MLDs. In unfolding MLDs, the FC-DNN is unfolded to construct an unfolding layer architecture following the PGD algorithm. By contrast, the last group of MLDs leverages the learning capability of DNNs to further optimize well-known classical detectors such as OAMP, TS, and SD. While black-box and unfolding MLDs are shown to outperform linear receivers with lower complexity, the OAMP-, TS-, and SD-based MLDs guarantee near-optimal performance with reduced complexity owing to the aid of DL.

Depending on the system configurations such as the size and modulation scheme, an appropriate MLD can be chosen for signal detection. For example, black-box MLDs and unfolding MLDs perform relatively well for a MIMO system with $N \ll M$ and low-order modulation such as

BPSK and QPSK. By contrast, in the case of $N \approx M$ and higher-order QAM signaling, the black-box and unfolding MLDs experience substantial performance loss. In this case, the incorporation of ML with classical iterative or near-optimal detection schemes can be a more appropriate choice to guarantee good performance for those challenging systems. Furthermore, we note that different groups of MLDs have different computational complexities. Therefore, for an optimal performance-complexity tradeoff, the performance and complexity of both the ML model and hand-engineered algorithm would have to be considered for the design of nonblack-box MLDs.

## V. BEAMFORMING AND CHANNEL ESTIMATION
In this section, we review applications of AI techniques for MIMO beamforming and channel estimation.

### A. FUNDAMENTALS OF BEAMFORMING
Beamforming is an effective means of improving the quality of the received signals in wireless communication systems. It can be realized via precoding at the transmitter and/or

combining at the receiver. In this subsection, without loss of generality, we provide the fundamentals of precoding design. Specifically, we consider the problem of beamforming design for a single-cell downlink system, where the base station (BS) is equipped with $N$ antennas and serves $K$ single-antenna users. The received signal at user $k$ can be given by

$$y_k = \boldsymbol{h}_k^H \sum_{k=1}^{K} \boldsymbol{f}_k s_k + n_k, \qquad (22)$$

where $s_k$ is the transmitted signal with $\mathbb{E}\left\{|s_k|^2\right\} = 1$, $\boldsymbol{h}_k \in \mathbb{C}^{N \times 1}$ is the channel vector, and $n_k \sim \mathcal{CN}(0, \sigma^2)$ is an AWGN noise sample at user $k$. Furthermore, $\boldsymbol{f}_k \in \mathbb{C}^{N \times 1}$ represents the beamforming vector for user $k$. The received SINR at user $k$ is given as

$$\text{SINR}_k = \frac{\left|\boldsymbol{h}_k^H \boldsymbol{f}_k\right|^2}{\sum_{i=1, i \neq k}^{K} \left|\boldsymbol{h}_i^H \boldsymbol{f}_i\right|^2 + \sigma^2}. \qquad (23)$$

Beamforming can be implemented by analog, digital, or hybrid analog/digital beamforming (HBF) architectures, leading to different optimization problems for the beamforming design. In particular, sum-rate maximization (SRM) is the most widely considered problem. Therefore, in the following, SRM is used for stating the beamforming design problem, without loss of generality. The optimization of beamformers based on other objective metrics such as SINR and BER performance are also discussed in the subsection on state-of-the-art beamforming designs.

#### c: DIGITAL BEAMFORMING
For digital beamforming (DBF), the SRM problem can be expressed as:

$$(\mathcal{P}_{\text{DBF}}): \max_{\boldsymbol{F}} \sum_{k=1}^{K} \log_2\left(1 + \text{SINR}_k\right), \quad \text{s.t. } \|\boldsymbol{F}\|_F^2 \leq P, \qquad (24)$$

where $P$ is the total power budget at the transmitter, and $\boldsymbol{F} = \left[\boldsymbol{f}_1, \ldots, \boldsymbol{f}_K\right]$ is the beamforming matrix.

#### d: ANALOG BEAMFORMING
In analog beamforming (ABF), analog circuitry with phase shifters and/or switches is used for signal processing. Therefore, the entries of the ABF matrix are required to have a constant modulus of $\frac{1}{\sqrt{N}}$, and their phases are adjustable in a given space depending on the resolution of the phase shifters. Therefore, ABF vectors belong to a feasible codebook $\mathcal{F}$, leading to the following SRM problem:

$$(\mathcal{P}_{\text{ABF}}): \max_{\boldsymbol{F}} \sum_{k=1}^{K} \log_2\left(1 + \text{SINR}_k\right), \quad \text{s.t. } \boldsymbol{F} \in \mathcal{F}. \qquad (25)$$

#### e: HYBRID ANALOG/DIGITAL BEAMFORMING
For HBF, we have $\boldsymbol{f}_k = \boldsymbol{F}^{RF} \boldsymbol{f}_k^{BB}$, where $\boldsymbol{F}^{RF} \in \mathbb{C}^{N \times N_{RF}}$ denotes the analog precoding matrix with $N_{RF}$ being the

number of RF chains, and $\boldsymbol{f}_k^{BB} \in \mathbb{C}^{N_{RF} \times 1}$ denotes the digital precoding vector for the $k$th user. Therefore, the SRM problem can be rewritten as

$$(\mathcal{P}_{\text{HBF}}): \max_{\boldsymbol{F}^{BB}, \boldsymbol{F}^{RF}} \sum_{k=1}^{K} \log_2\left(1 + \text{SINR}_k\right) \qquad (26a)$$

$$\text{s.t. } \left\|\boldsymbol{F}^{RF} \boldsymbol{F}^{BB}\right\|_F^2 \leq P, \quad \boldsymbol{F}^{RF} \in \mathcal{F}. \qquad (26b)$$

The SRM problems $(\mathcal{P}_{\text{ABF}})$, $(\mathcal{P}_{\text{DBF}})$, and $(\mathcal{P}_{\text{HBF}})$ for beamforming design are challenging because of their nonconvexity. Existing classical hand-engineered algorithms for the SRM problems only achieve suboptimal solutions but remain computationally complex because of their complex matrix operations, such as matrix inversions and factorization and iterations. To overcome this challenge, various ML-aided beamformers (MLBs) that were introduced recently are discussed in the following subsection.

### B. STATE-OF-THE-ART MLBs
The objective of MLBs is to leverage the learning capability of ML models, such as SVM, CNN, DNN, DRL, and FDL, to directly generate the beamformer [139]–[143], [161] or to suggest its design by learning its important features [144]–[148], [151]. In the former application, an ML tool is modeled as a black box that outputs either the precoding/combining vector, their index in a codebook, or the precoded/combined signals. We refer to MLBs with this application of ML as *black-box MLBs*. The term *black box* refers to the learning capability of ML to output the desired solution, which is the beamforming vector/matrix or its indicator, without expert knowledge. By contrast, the latter application leverages both expert knowledge and the learning capability of ML models. Details of these MLB groups are discussed in the following.

#### 1) BLACK-BOX MLB
Black-box MLB has found widespread use [139]–[144], [161], and has been used in combination with various ML models, such as DNNs [139]–[142], [161], SVM [143], KNNs [142], and the support vector classifier (SVC) [142]. In this group, the task of the ML model is to generate the precoder directly [139], [140], [143] or to output an indicator of the precoder [141]–[144]. An important observation is that, for black-box MLBs that output the beamforming vector/matrix directly, unsupervised learning is generally used. In contrast, supervised learning is more widely used for black-box MLBs that generate beamformer indicators. In the following, we discuss these two subgroups, which reveal interesting observations on the use of learning methods.

#### a: BLACK-BOX MLBs BASED ON UNSUPERVISED LEARNING
The black-box MLBs in this subgroup employ DNNs that are trained using unsupervised learning [139], [140] owing to the unavailability of training labels. Specifically, unlike MLDs, where the label is set exactly to the transmitted signal,

it is difficult to find an appropriate label for MLB design. It is possible to use classical beamforming algorithms to generate the label. However, in this case, the performance of the resulting MLB is limited by that of the classical beamforming scheme. Therefore, a DNN was trained to enable it to learn to optimize the analog beamformer thus maximizing the sum rate, or equivalently, to solve ($\mathcal{P}_{\text{ABF}}$) [139]. A similar method was proposed [140] to generate the digital precoder, i.e., to solve ($\mathcal{P}_{\text{DBF}}$). The proposed black-box MLB was demonstrated to improve the computational efficiency significantly with performance close to that of the classical weighted MMSE (WMMSE) algorithm [140].

### b: BLACK-BOX MLBs BASED ON SUPERVISED LEARNING

Typical black-box MLBs in this subgroup were reported by several studies [141]–[144]. Specifically, by modeling the beamforming selection as a classification problem, the optimal beamformer in the codebook $\mathcal{F}$ can be predicted by DNN, KNN, and SVC models [142], and an SVM model [143], [144]. In particular, simulation results confirmed that DNN-assisted black-box MLB performs almost optimally and outperforms both KNN- and SVC-aided methods [142]. Furthermore, as long as sufficient training data are used, the derived classification model can select the optimal analog precoder with low complexity [144]. Unlike [142]–[144], in which the classification problem is considered, the DNN in [141] is trained to predict the achievable rates corresponding to multiple precoders in the codebook $\mathcal{F}$, allowing the one with the highest predicted rate to be chosen for application. Furthermore, Huang *et al.* [161] developed a DNN-based black-box MLB to realize end-to-end hybrid precoding in mmWave massive MIMO systems. In other words, the DNN is trained to output the precoded signals, which are ready for transmission. This method is capable of minimizing the BER and enhancing the spectrum efficiency of the mmWave massive MIMO, which achieves superior performance in hybrid precoding compared with conventional schemes while substantially reducing the computational complexity.

### 2) FEATURE-BASED MLB

The common objective of MLB schemes in this group is to leverage expert knowledge to transform the task of predicting the beamformer to predicting their key features, which are sufficient to construct the beamformer. This method is beneficial for both the training and application phases. Specifically, instead of learning a beamforming vector/matrix [139], [140], [143], the ML model only needs to learn certain key features, which are usually scalars. The considerably reduced number of learning variables enables the ML model to be trained more effectively to increase the accuracy. Furthermore, the smaller-sized output and reduced number of learning variables also simplify the network architecture, leading to a reduction in the overall computational complexity in the online application phase.

Typical work in this group entailed equipping the transmitter with a uniform planar antenna array with $N_R$ rows and $N_C$ columns of antenna elements [145]. Based on this structure, the beamforming vector can be factorized as $\boldsymbol{f}_k = (\mathcal{F}^h)_r \otimes (\mathcal{F}^v)_c$. Here, $\mathcal{F}^h$ and $\mathcal{F}^v$ are the discrete Fourier transform (DFT)-based codebooks for the horizontal and vertical dimensions, respectively, and the subscript $(\cdot)_i$ indicates the codeword index in the codebook. Because the DFT codebooks are available, only the codeword indexes $r$ and $c$ are required to obtain $\boldsymbol{f}_k$. To this end, a DNN can be trained to predict $r$ and $c$ [145]. Another application of DNN to beamforming was reported by [146], which solved the SRM problem by proposing the use of a DNN to approximate the conventional beam management and interference coordination (BM-IC) algorithm. The performance of this proposed DNN-based BM-IC scheme was shown to be comparable to the conventional BM-IC algorithm, but requiring less computational time.

CNN has strong feature extraction as well as approximation abilities. In addition, compared to DNN, CNN has reduced the number of learning parameters by sharing weights and biases. Therefore, it is employed in [147], [148] to obtain key features of the beamformer. This approach is illustrated in Fig. 8. Three CNN-based MLB schemes were proposed for three typical optimization problems, i.e., SINR balancing, power minimization, and SRM problems [147]. The common procedure in [147] to solve these problems was to relax them to virtual equivalent problems such that the solution, i.e., the beamforming matrix, can be expressed as a function of unknown scalars, which are easier to predict. Then, CNNs were employed to predict these scalars, followed by a beamforming recovery module to construct the required beamforming matrix. A similar method to solve the SRM problem was used by proposing a CNN-based beamforming prediction network (BPNet) [148]. Specifically, the SRM problem is first separated into power allocation and virtual uplink beamforming design modules, which can be jointly optimized by the BPNet. The simulation results [147], [148] show that the performance of the MLB scheme proposed by Xia *et al.* [147] closely approximates that of the WMMSE, whereas the performance of BPNet [148] improves with respect to that of the WMMSE, both with much lower computational complexity.

Recently, reconfigurable intelligent surface (RIS) has emerged as a spectral- and cost-efficient approach for wireless communications systems. RISs provide passive beamforming gains to enhance the connection between the BS and MS, especially when the direct link between them are blocked or considerably attenuated by obstacles such as trees and buildings. The passive beamforming design of RISs is usually formulated in the rate/capacity maximization problem, which are non-convex and intractable. Existing hand-engineered schemes for passive beamforming design, such as the alternating optimization or projected gradient approaches, require a high computational complexity. To overcome this problem, various ML-based passive beamforming schemes have
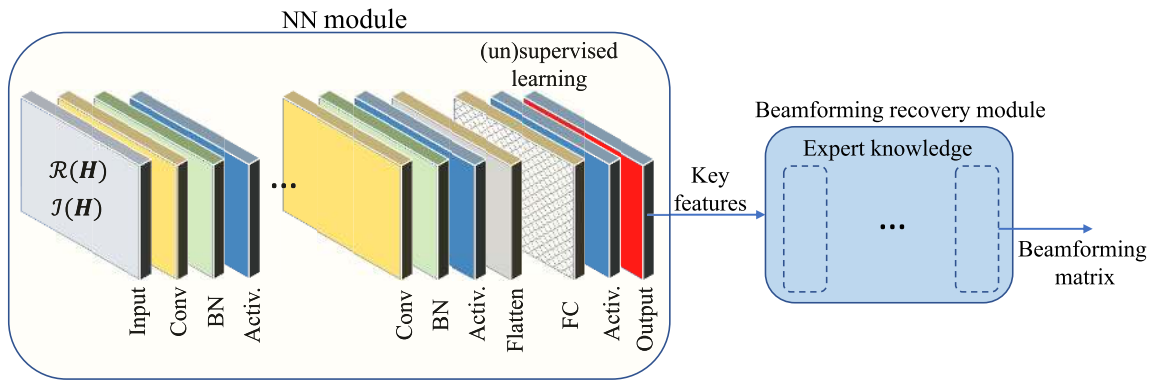
**FIGURE 8.** Feature-based DLB framework [147] with two main modules: NN and beamforming recovery. The NN module consists of input, convolutional (conv), batch normalization (BN), activation (activ.), flatten, fully connected (FC), and output layers. The NN module outputs the key features, which allow the beamforming matrix to be constructed by leveraging the expert knowledge in the beamforming recovery module.

been proposed [149], [150], [156], [160], [162]. In particular, in [149], [150], DNNs are leveraged for the passive beamforming design. Specifically, while a supervised learning strategy is employed for the DNN to predict the achievable rates associated with RIS interaction vectors in [149], unsupervised learning is used to generate the phase shifts of the RIS in [150].

### 3) RL-BASED BEAMFORMING

Several studies applied RL to beamforming design [151]–[155]. Unlike ML-based beamforming schemes, the common motivation for using RL for beamforming is to formulate the beamforming design problem based on the RL concepts, i.e., the agent, actions, and rewards, and find the solution from interaction with the defined environment. For example, Wang *et al.* [153] proposed a novel DRL-based HBF design method named PrecoderNet to design the hybrid precoding matrix and improve the spectral efficiency and BER performance of mmWave point-to-point massive MIMO systems. This proposal is based on the finding that the HBF problem can be modeled as a Markov decision process (MDP) that can be effectively solved based on DRL by defining the system sum rate, i.e., the objective function, as reward, and the beamformer, i.e., the desired solution, as state. PrecoderNet was numerically shown to significantly outperform classical HBF schemes, such as orthogonal matching pursuit (OMP) and MMSE. Mismar *et al.* [152] questioned the existence of a method that could jointly solve the nonconvex optimization problem of beamforming, power control, and interference coordination design to achieve the SINR upper bound without performing an exhaustive search over the entire solution space. Therefore, they utilized the ability of DRL to explore the solution space by learning from interaction. In particular, the proposed DRL algorithm does not require CSI, and hence, channel estimation becomes unnecessary. Furthermore, DRL is also exploited for passive beamforming problem for RIS-assisted wireless communications in [156], [157], [162].

Furthermore, DRL was also leveraged for beamforming in Fog radio access networks (RAN) (F-RAN). Specifically, DRL-based mode selection and resource management for F-RAN was proposed [154]. Using DRL, the controller can quickly control the communication modes of user equipment (UE), i.e., cloud RAN (C-RAN) and device-to-device (D2D) modes, and the on–off states of processors in the cloud. After the controller makes the decision, precoding vectors for UE in the C-RAN mode are subsequently optimized in terms of their QoS, power, and computing capability constraints. A similar application of DRL to mode selection and precoding design in F-RAN was investigated in [155]. Specifically, DRL-based adaptive selection of backhaul and fronthaul transfer modes was proposed with the aim of optimizing the performance of content delivery. Numerical results showed that the proposed DRL-based schemes in [154] and [155] outperformed baseline schemes.

### 4) FL-BASED BEAMFORMING

Because of the special characteristics of FL, its application to beamforming is discussed separately from the above groups. Particularly, FL is more related to distributed learning across the users to preserve data privacy and save network bandwidth [158] rather than to beamforming design. With respect to the beamforming problem, FL is useful for reducing the overhead for the transmission of CSI from the users to the BS, and it can also be combined with an ML technique to achieve the benefits of both ML and FL. For example, Elbir and Coleri [159] designed a CNN to generate analog beamformers at the output. Especially, in this work, instead of using global training as is usually the case in ML schemes, an FL-based framework, in which the model is trained at the BS by collecting only the gradients from users, was employed. The simulation results showed that, compared with ML, FL is more tolerant to the imperfections and corruptions of the CSI, and at the same time, it also has lower computational complexity. By contrast, beamforming was used to improve the performance and the convergence rate

for FL via over-the-air computation in [158] or to enhance the passive beamforming performance and user privacy in RIS-aided wireless communications [160]. This approach resulted in a significant reduction in training loss and an improvement in the training accuracy.

## C. CHANNEL ESTIMATION

Channel estimation is an important task in signal processing and significantly affects the performance of the signal detection and beamforming schemes. In this subsection, for completeness, we review ML-based channel estimation schemes that were recently reported in the literature.

Consider a block-fading MIMO system with the input–output relationship given in (7). To estimate the channel matrix $H$, $P$ pilot signal vectors $\{s_1, \ldots, s_P\}$, which are known at both the transmitter and receiver, are transmitted. Let $y_p$ and $n_p$ be the received signal and let the AWGN noise vectors correspond to $s_p$. Furthermore, by denoting $S = [s_1, \ldots, s_P]$, $Y = [y_1, \ldots, y_P]$, and $N = [n_1, \ldots, n_P]$, we can write

$$Y = HS + N.$$

The task of channel estimation is to recover the channel matrix $H$ based on knowledge of $S$ and $Y$. The LS and MMSE schemes are two typical classical channel estimators. In the LS scheme, the estimated channel is given as [163]

$$\hat{H}_{\mathrm{LS}} = Y S^H \left( S S^H \right)^{-1}. \tag{27}$$

This indicates that the LS estimator only requires the observations $Y$ without requiring the channel statistics. However, its mean-square error (MSE) is inversely proportional to the SNR, implying that it may be subject to noise enhancement. By contrast, the MMSE estimator improves the performance with respect to the LS scheme, but requires knowledge of the channel covariance matrix $R = \mathbb{E}\{H^H H\}$. In particular, in the MMSE method, the estimated channel is given as [163]

$$\hat{H}_{\mathrm{MMSE}} = Y \left( S^H R S + \sigma^2 M I \right)^{-1} S^H R. \tag{28}$$

The assumption of knowledge of $R$ can be unrealistic in practical scenarios, especially in fast fading channels. Furthermore, the matrix multiplications, additions, and inversion performed by the MMSE estimator in (28) are computationally intensive. In addition, the common limitation of the aforementioned classical estimators is the significant performance degradation when the pilot length is smaller than the number of transmit antennas, i.e., $P < N$. We note that the assumption of $P \geq N$ can be impractical for large-sized systems such as the downlink of massive MIMO systems, where $N$ is very large. Furthermore, the use of a long pilot sequence generates substantial training overhead, thereby reducing the overall spectral efficiency of the system as well as increasing the computational load.

The application of ML to channel estimation to overcome the limitation of classical hand-engineered channel estimation schemes, especially for massive MIMO systems,

has recently attracted research interest. Numerous ML-based channel estimators (MLCEs), which exploit various ML models, learning methods, and algorithms, have been reported in the literature. Generally, because of the heavy learning task associated with channel estimation, DNN and CNN are commonly used. Furthermore, for channel estimation, supervised learning is more widely used than unsupervised learning. In this subsection, in which we review state-of-the-art MLCEs, we focus on their design methodology, i.e., the motivation and how ML is leveraged for channel estimation. Our literature review revealed that ML can be leveraged in two different ways. First, an ML tool can be modeled as an independent channel estimator to directly output the channel matrix/vector. We refer to this group of MLCEs as *black-box MLCEs*. Second, ML can be used to either estimate key features, which are then used to reconstruct the channel matrix, or to support classical well-known channel estimation algorithms. Therefore, we refer to the MLCEs in this group as *feature-based MLCEs*. These typical groups of MLCEs are discussed in the following.

### 1) BLACK-BOX MLCEs

The task of ML in black-box MLCEs is to learn and output the channel matrix $H$ without expert knowledge. Because this is a computationally intensive task, complicated NN architectures such as DNN [164]–[166], CNN [167]–[169], and RNN [167] are typically used. However, to focus on the design of MLCEs rather than on the ML tools and learning techniques, we classify the black-box MLCEs based on their design architectures.

#### a: SINGLE-STAGE MLCEs

A very straightforward black-box MLCE, which fully exploits the learning ability of a DNN, is the direct-input DNN (DI-DNN) proposed by Gao *et al.* [164]. In the DI-DNN, the pilot signals are all set to 1, and the DNN can learn the channel matrix $H$ based only on the received signal $Y$. In particular, to reflect practical systems in which the resolution patterns of analog-to-digital converters (ADCs) differ, Gao *et al.* considered mixed-resolution ADCs, i.e., ADCs with both high and low resolution components. Therefore, the input of the DNN can be the exact received signal $Y$ or its quantized version $\mathcal{Q}(Y)$. The simulation results showed that the proposed scheme is less affected by the error floor. As a result, it outperforms the classical linear MMSE channel estimation method across the entire SNR regimes. Other than the aforementioned approach [164], an MLCE that uses a transformed version of the received signals as the input of the DNN was proposed [165]. Specifically, to minimize user operations and reduce the feedback overhead, the signal received by the user is compressed to a scalar value and provided as feedback to the BS for channel estimation. This scheme requires only limited feedback and is proposed for channel estimation in frequency division duplex massive MIMO systems.

### b: TWO-STAGE MLCEs

A two-stage channel estimation process was proposed in [166]. In the first stage, the pilot and received signals are input into a DNN, which learns and outputs the first estimate of the channel matrix. Then, the estimated channel is further improved by another DNN, which, at this time, exploits the data and received signals in the transmission phase. The advantage of this two-stage MLCE compared with the conventional single-stage channel estimator is that the quality of the estimator can be improved, not only in the training phase but also in the transmission phase. Therefore, a larger portion of the coherent time can be allocated for the transmission phase to achieve higher spectral efficiency, as shown by the simulation results in [166]. Liao *et al.* [167] proposed architecture named ChanEstNet, which uses two networks: a CNN to extract channel response feature vectors, which are then processed by an RNN to obtain the final estimate of the channel.

The two-stage MLCE [166], [167] fully exploits the ML models in both stages. In contrast, other researchers proposed to perform the first-stage estimation based on conventional schemes [168], [169]. Specifically, in the first stage, a coarse estimate of the channel matrix was obtained by removing the effects of the beamformers [168] or by using an LS filter [169]. The coarse channel estimate is then processed by a CNN to generate an estimate of the channel matrix with higher performance. Both of the proposed MLCEs [168], [169] outperform the classical MMSE estimator and its variants. The application of two-stage MLCE for wireless energy transfer was also considered [170]. Specifically, the use of a shallow FNN to obtain the energy feedback information harvested at the output was proposed [170]. Then, a DNN is used to learn and output the channel from the energy feedback information.

### 2) FEATURE-BASED MLCE

Instead of using a single or multiple NNs to generate the channel matrix, as in black-box MLCEs, feature-based MLCEs leverage expert knowledge such as the known channel characteristics and structure. In this way, the ML model only needs to estimate an intermediate parameter, which is then used to reconstruct an estimate of the channel matrix. For example, given the channel model, channel parameters such as the gains or angles can be learned, which can be readily exploited to build the channel matrix. Another approach is to leverage ML-aided image processing schemes for channel estimation based on the similarity between sparse channels and the natural image. These approaches are reviewed in the following paragraphs.

### a: PARAMETER ESTIMATION

The parameters of mmWave channels model were estimated [171], [172]. Specifically, in a typical uplink channel of a massive MIMO system, the channel between the BS, equipped with $N$ antennas, and the $k$th user

can be expressed as

$$h_k = \sum_{i=1}^{N_p} g_{k,i} a(\theta_{k,i}) = A_k g_k, \qquad (29)$$

where $N_p$ is the number of paths, $g_{k,i}$ is the complex gain of the $i$th path between user $k$ and the BS, and $A_k = \left[ a(\theta_{k,1}), a(\theta_{k,2}), \ldots, a(\theta_{k,N_p}) \right]$. Here, $\theta_{k,i}$ denotes the physical angle of arrival (AoA) of the $i$th path, and $a(\theta_{k,i})$ is a steering vector given as

$$a(\theta_{k,i}) = \frac{1}{\sqrt{N}} \left[ 1, e^{j\pi\theta_{k,i}}, \ldots, e^{j(N-1)\pi\theta_{k,i}} \right]^T.$$

From (29), it is observed that the information of the complex gains and AoAs, i.e., $\{g_{k,i}, \theta_{k,i}, i = 1, \ldots, P\}$, are key characteristics that model the channel vector $h_k$. Motivated by this, Huang *et al.* [171] proposed using DNNs to estimate the complex gains and AoDs. This scheme outperforms many existing classical channel estimation schemes such as compressed sensing (CS). Another method of leveraging the channel structure in (29) was proposed [172]. In particular, instead of estimating the parameters of $h_k$, a DNN was employed to estimate the amplitudes of the elements of the beamspace channel vector $\tilde{h}_k = A_k^H h_k$ [172]. Based on the estimated amplitudes of $\tilde{h}_k$, the indices of dominant entries in $h_k$ are determined. This allows $h_k$ to be approximately reconstructed based on the sparsity of $h_k$ and the fact that $A_k^H A_k = I$. The simulation results [172] verify the performance improvement of the proposed scheme compared to the classical OMP and distributed grid matching pursuit (DGMP) schemes.

### b: IMAGE PROCESSING-BASED MLCEs

Unlike the above mentioned studies [171], [172], in which the channel parameters were estimated by ML models, the sparse channel matrix was considered as a natural image, and this motivated the application of DL-aided image processing techniques for channel estimation [173]–[176]. For example, He *et al.* [174] found that the correlation among the elements of the channel matrix is very similar to that of a 2D natural image: the channel is sparse and the changes between adjacent elements are subtle. Therefore, the authors proposed to leverage the learned denoising-based approximate message passing (LDAMP) network, originated from image recovery, for channel estimation. In the LDAMP network, the CNN-based denoiser (DnCNN), illustrated in Fig. 9, can handle Gaussian denoising problems. In this scheme, a noisy channel $h + \beta z$ is processed by the DnCNN to estimate the residual noise $\hat{z}$, which is then subtracted from the input to obtain the estimate of the channel $\hat{h}$. Interestingly, the LDAMP network outperforms many CS-based channel estimation algorithms. However, a drawback of the LDAMP scheme is that the DnCNN denoiser is tailored to a specific noise level, and it only performs well for this trained noise level [173]. This motivated the proposal of the fast and flexible denoising CNN (FFDNet) [173]. In contrast to LDAMP,
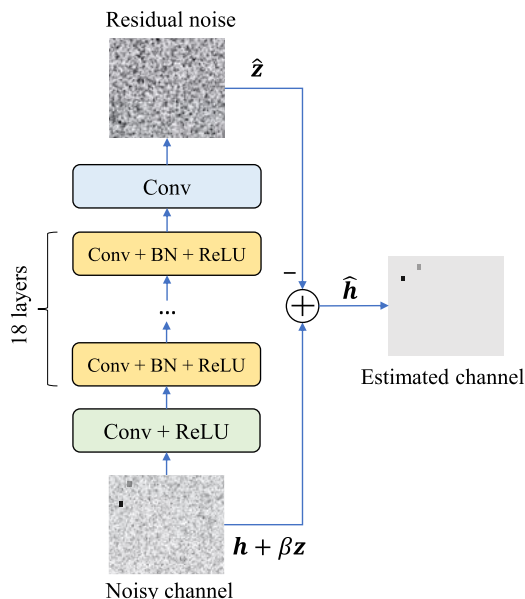
**FIGURE 9.** Network architecture of the DnCNN denoiser [174], consisting of 20 convolutional layers. The first layer employs 64 filters sized $3 \times 3 \times 1$, followed by a ReLU. Each of the next 18 successive layers contains 64 filters sized $3 \times 3 \times 64$, each followed by a BN and ReLU. The last layer uses a $3 \times 3 \times 64$ filter to reconstruct the signal.

FFDNet, with a flexible noise level map at the input, is suitable for a wide range of SNRs. As a result, it outperforms the LDAMP scheme across a large range of noise levels.

Another MLCE [175] is also based on the image recovery problem. However, it focuses on the optimization of CSI feedback. In particular, to avoid excessive feedback overhead of the CSI, the authors proposed the CsiNet, which encodes the original channel matrix to compress codewords at the receiver before feeding back to the transmitter for beamforming. At the transmitter, the compressed CSI is decoded by the CsiNet to the original channel. Here, the CsiNet is constructed based on CNNs. The aforementioned schemes [173]–[175] are based on supervised learning, which requires a large number of parameters to be trained before online application becomes possible. To overcome this limitation, Balevi *et al.* [176] proposed an untrained DNN based on the deep image prior network for channel estimation. The idea of this scheme is that, instead of denoising the channel matrix, the received signal is first denoised by a DNN, followed by the classical LS channel estimation scheme.

#### c: CLASSICAL LS/MMSE-BASED MLCEs
The incorporation of ML in a classical channel estimator was proposed [177], [178]. Specifically, CNNs were used to approximate the MMSE channel estimator [177]. Alternatively, the LS estimate of the channel was exploited as an input of the DNN-based estimator [178]. These combinations of a classical estimator and ML models result in significant performance improvement with respect to conventional

hand-engineered estimators. In particular, the MLCE can compensate for the performance loss resulting from insufficient pilot signals [178]; hence, it outperforms the linear MMSE estimator. On the other hand, a hierarchical learning algorithm was proposed to avoid convergence to local optima during the training of the NN [177]. As a result, the trained NN can be optimized for a general channel model using stochastic gradient methods. The simulation results [177] indicated that, for a simple channel model, the performance of the proposed learning-based methods is less promising but is significantly improved for a more realistic channel model such as the 3GPP model with multiple paths and different path gains.

### D. SUMMARY AND TAKEAWAY POINTS
In this section, typical MLBs and MLCEs were discussed on the basis of their designs, which are summarized in Tables 5 and 6, respectively. Whereas black-box MLBs/MLCEs focuses on the learning ability of ML models to learn and output beamforming/channel matrices, feature-based MLBs/MLCEs utilizes both expert knowledge and machine intelligence to construct the beamformer and channel estimator. For the beamforming problem, the exploited expert knowledge can either be the known optimal solution or well-developed hand-engineered iterative algorithms to solve SRM problems. For channel estimation, channel characteristics such as the path gains, angles, or sparsity are leveraged. In particular, DL-aided image processing techniques can be utilized for channel estimation owing to the similarity between a 2D natural image and a sparse channel matrix. The proposed MLBs and MLCEs are shown to have advantages in terms of performance improvement and/or complexity reduction with respect to classical schemes.

Various ML models were deployed for MLBs and MLCEs, ranging from simple SVM, SVC, and KNN models to more sophisticated DNN, CNN, FL, and RL models. Especially, DNNs and CNNs are widely used owing to their powerful feature extraction capabilities as well as their wide application to image processing. Most of the MLBs and MLCEs outperform classical hand-engineered schemes such as WMMSE for beamforming and LS, linear MMSE, OMP, and CS-based methods for channel estimation. MLBs and MLCEs have developed along two major well-exploited directions: designs based on black-box and feature-based beamformers/estimators. The former are generally more computationally complex than the latter because ML models need to learn and output a solution, which is a vector/matrix. However, it is independent from the solution structure. By contrast, feature-based MLBs/MLCEs generally have reduced complexity because of the simpler learning task. However, they depend on the structure and characteristics of the solution. For example, the beamformers in analog, digital, and hybrid beamforming have different constraints on the phases and amplitudes of the entries, or, for channel estimation, the channel models and sparsity vary depending on the communication scenario. These practical aspects

**TABLE 5.** Summary of existing MLBs.

| MLB group | Sub-group | Paper | ML model | Interesting observation |
|---|---|---|---|---|
| Black-box MLBs | Unsupervised | [139], [140] | DNN | • Directly generate the beamforming matrix<br>• ML models are trained using unsupervised learning. |
| | Supervised | [141]–[144] | DNN, CNN, SVM, KNN, SVC | • Output an indicator of the beamformer<br>• ML models are trained using supervised learning. |
| Feature-based MLBs | DFT-based factorization | [145] | DNN | • Beamforming vector is modeled as $f_k = (\mathcal{F}^h)_n \otimes (\mathcal{F}^v)_m$.<br>• Used to predict the codeword indices $n$ and $m$ |
| | BM-IC algorithm-based | [146] | DNN | • Used to approximate the conventional BM-IC algorithm<br>• Performance comparable to that of the BM-IC algorithm, but requires much less computational time |
| | Optimization-based | [147], [148] | CNN, DNN | • Use a CNN to predict scalars, which are solutions of optimization problems, i.e., SINR balancing, power minimization, and SRM problems.<br>• Use a beamforming recovery module to construct the beamforming matrix<br>• Performance comparable to or higher than that of the WMMSE scheme with much lower complexity |
| | Passive beamfomring for RIS | [149], [150] | DNN | • Use a DNN with supervised or unsupervised learning to predict the achievable rates associated with RIS interaction vectors or to output the phase shifts of RIS<br>• Need a beamforming recovery module to construct the diagonal passive beamforming matrix from the interaction vectors or the set of phase shifts |
| RL-based MLBs | N/A | [151], [152]–[157] | RL, NN | • Formulate the beamforming design problem using the concepts of RL<br>• Can be applied to various networks such as MIMO, D2D, F-RAN, RIS-aided wireless networks |
| FL-based MLBs | N/A | [158]–[160] | FL, CNN | • Useful to reduce the overhead for CSI<br>• More tolerant to the imperfections and corruptions of CSI<br>• Enhanced passive beamforming performance and user privacy in RIS-aided wireless systems<br>• Lower computational complexity |

**TABLE 6.** Summary of MLCEs and selected properties.

| MLCE group | Sub-group | Paper | ML model | Interesting observation |
|---|---|---|---|---|
| Black-box MLCEs | Single-stage | [164], [165] | DNN | • Uses a single ML model to learn and output the channel matrix based on the received signal or its compression, without expert knowledge |
| | Two-stage | [166]–[170] | CNN, DNN, RNN, AE | • In the first stage, a coarse estimate of the channel is obtained based on classical or ML schemes, which are then processed by another ML model to output the high-accuracy channel.<br>• Outperform classical estimators such as LS and MMSE<br>• Generally have higher computational complexity than the single-stage black-box MLCEs and simple classical estimators |
| Feature-based MLCEs | Parameter estimation | [171], [172] | DNN | • Key parameters such as the gains and angles of the channel are estimated, which are then used to reconstruct the channel matrix based on expert knowledge.<br>• Outperform classical channel estimation schemes such as CS and OMP<br>• Requires expert knowledge on the channel model |
| | Image processing-based | [173]–[176] | CNN, DNN | • Apply DL-aided image processing techniques for channel estimation<br>• Leverage denoising techniques<br>• Outperform classical channel estimation schemes such as LS and CS<br>• Require similarity between the channel and the natural image, more suitable to sparse channels |
| | Classical scheme-based | [177], [178] | DNN, CNN | • Use a classical estimator to generate the input for ML models or approximate a classical scheme using ML models for performance improvement and complexity reduction |

challenge the design of MLBs/MLCEs and create room for further studies in these fields.

## VI. RESEARCH CHALLENGES AND FUTURE DIRECTIONS
This section discusses the challenges presented by intelligent radio signal processing and highlights several promising directions.

### A. END-TO-END LEARNING
Most studies on intelligent radio signal processing focused on modular design; that is, the entire system is composed of blocks such as transmitters, amplifiers, channels,

and receivers, each of which has its individual process. However, the number of services and applications that require an end-to-end performance guarantee is expected to increase [179], [180]. Wireless channels are becoming increasingly complex and are affected by many factors, adversaries, channel nonlinearities, and hardware impairments. Additionally, the performance metrics (e.g., rate 0.1-1 Tbps, and end-to-end delay 1 ms), frequency bands (sub-6 GHz, mmWave, THz bands), waveforms, and modulation techniques (e.g., index and spatial modulation) are expected to increase in next-generation networks [1]. Generally, AI has great potential to provide effective end-to-end solutions.

Motivated by recent seminal work [8], many studies have been dedicated to improving wireless systems, e.g., modulation identification and channel estimation for spectrum situation awareness and wireless security [36]. However, many challenges remain for end-to-end learning when various features and new requirements are taken into consideration.

### B. DISTRIBUTED, CENTRALIZED, OR HYBRID LEARNING

Most AI-enabled techniques for signal processing focus on achieving the best performance (e.g., accuracy and spectral/energy efficiency), but the implementation complexity is often neglected. Indeed, when the training server has sufficient computing and storage capabilities, centralized learning is preferred over distributed learning. This is a reasonable choice because large amounts and various types of data can be collected and trained centrally. Meanwhile, problems arising from synchronization, data distribution, and communication dynamics can be safely ignored. The preference for centralized and distributed learning is similar to, for example, centralized optimization vs. distributed game approaches, cloud vs. edge computing, and RAN vs. C-RAN, which has been developed for wireless communications for decades. Centralized learning has three main drawbacks: high computational complexity, security and data privacy (owing to the high concentration of data), and poor scalability. This approach also fails to exploit the large number of computing resources distributed over the network and is not applicable to scenarios in which a centralized entity is not available, e.g., ad hoc and wireless sensor networks. An example of distributed learning for online medium access control in a spectrum sharing network has been discussed [181]. On the other hand, hybrid learning is a promising solution to these drawbacks as it potentially maintains a balance between complexity and performance. In particular, a learning model can be divided into smaller tasks and then trained hierarchically, e.g., cloud, edge, and end devices. However, selecting the learning mode largely depends on problems and design objectives, and it should be carefully considered for practical implementation. Transfer learning is also a good AI technique to be considered for radio signal processing since learning model can be improved by utilizing retrained AI models using a similar task.

### C. MODEL COMPRESSION AND ACCELERATION

AI-enabled approaches have been shown to have many advantages over conventional approaches, but they are usually computationally and memory intensive. Compared with datasets in computer vision and natural language processing, the datasets used to train signal processing models are typically smaller. However, IoT/edge devices are limited in terms of computing capability and memory and storage capacity. Additionally, because many AI-based services will be available at the network edge, being able to deploy intelligent signal processing algorithms in resource-limited devices (e.g., sensors, industrial IoT and wearable devices, and drones) with comparable performance plays a vital role.

For instance, training the MCNet architecture [5] to classify 24 modulations takes approximately 24 h on a computer with 16 GB memory and a NVIDIA GeForce GTX 1080Ti GPU, and the number of trainable parameters can reach 220,000. The training time would increase markedly and the classification accuracy decreases if the model is trained on a personal computer with less processing power. A promising solution to overcome these challenges is model compression and acceleration. Compressing and accelerating DL techniques can be classified into four main groups [182]: 1) parameter pruning and quantization, 2) low-rank factorization, 3) transferred/compact convolutional filters, and 4) knowledge distillation. Over the last few years, DL has employed a large number of techniques and interested readers are invited to refer to a recent survey [183] for more details. Our observation is that most existing solutions focus on achieving high performance while ignoring the suitability of the proposed DL models for resource-constrained devices. Another promising direction is to offload the training and inference processes to the edge (i.e., edge intelligence in the literature), i.e., AI techniques are employed at the network edge to intelligently process radio signals. For example [184], the joint task allocation and downlink beamforming problem was optimized to minimize the total energy consumption. Applications of a 6G technology, namely reconfigurable intelligent surface, for edge inference, was reported [185].

### D. DATASET GENERATION AND UNIFICATION

The quality of training data greatly affects the performance and prediction accuracy of AI-enabled techniques. Compared with other fields such as computer vision and healthcare, the communication and signal processing communities were not ready to standardize data generation methods and unify the datasets for performance evaluation. The reason for this is that previous network generations can operate effectively with conventional approaches, such as queuing models, convex optimization, and game theory. However, some initiatives and competitions have been launched to resolve this issue, for example, IEEE ICC 2020 for vision-aided wireless systems [186] and IEEE CTW 2020 for user localization [187]. The Machine Learning for Communications ETI led by the IEEE is one of the best initiatives as it maintains a collection of datasets, papers, reproducible codes, etc., about the use of ML for wireless communications. However, the number of datasets is quite small and completely inadequate for a substantial number of network scenarios and problems in wireless systems and radio signal processing. Moreover, the size and quality of these datasets remain questionable as only a few are ready for practical implementation. Additionally, datasets would need to be continuously updated as the network will become highly dynamic, dense, and heterogeneous.

### E. UNIVERSALITY AND PRACTICALITY

Despite a few years of development and usage, the number of published papers and preprints are increasing daily, and many studies are devoted to solving the same problem. Moreover,

some researchers prefer to use simulated and private datasets, whereas others use public ones. This makes it difficult to compare the proposed algorithms when they are implemented in the same network setting and design objective. For example, the RadioML dataset [78] was created for the classification of 11 modulation types (8 digital and 3 analog), which have since become 24 modulations.[1] Utilizing the same RadioML dataset, Huynh-The *et al.* [5] applied their DL models to all the modulation types, whereas other researchers only considered a set of selected modulation formats, e.g., 5 types (BPSK, QPSK, 8PSK, 16QAM, 64QAM) [96], and 11 types (the RadioML2016.10a dataset) [188]. Additionally, other researchers tested their classification approaches using simulated datasets, wherein emulating realistic channel impairments is difficult [73], [79]. The fact is that, the higher the modulation order, the lower the classification accuracy [5]. Therefore, certain studies only experimented with low-order modulation formats and the applicability to high-order modulations is, therefore, questionable. All the aforementioned considerations encourage research communities to follow a standard methodology to enhance the practicality. Additionally, existing approaches need to be examined to make sure that they and/or their modifications can support new services and scenarios in the future.

### F. DEEP SEMISUPERVISED AND ACTIVE LEARNING

In our summary of recent studies on intelligent signal processing, we observed that they mostly focus on (deep) supervised learning. The main motivations for the use of supervised learning in wireless communications are its high performance (e.g., classification accuracy) and exploitation of available labeled datasets. The use of supervised learning in wireless communications would also be highly desirable; however, labeled datasets for a learning task are not always available and would even be time consuming and costly to construct in many cases. Most recent AI research is based on simulated datasets that were created using communication tools, and wherein channel characteristics and design features are emulated. In many cases, optimization problems are formulated and solved to obtain the locally optimal solutions, which are then considered as labeled instances of the supervised learning model. Therefore, the performance of an unsupervised learning method could possibly be higher than that of its supervised counterpart. An example is the optimization of the weighted sum rate problem, which was achieved by optimizing the transmit beamforming for a downlink MIMO system [140]. The unsupervised learning approach outperforms the supervised learning method for various settings such as the SNR and number of transmit antennas. This interesting observation shows that (deep) unsupervised learning has the potential to deliver high performance, especially when truly labeled datasets are not available. The availability of domain knowledge and truly labeled examples (i.e., real data or data generated by global optimization algorithms)

would render semisupervised learning a promising solution for this problem as it combines both the supervised and unsupervised approaches to exploit their respective advantages. Active learning provides other ways to solve the challenge of creating labeled instances. In particular, the central server (e.g., eNBs equipped with computing capabilities) may query end devices to provide labeled instances, thereby improving the learning performance using the collected labeled data. Applications of active learning for signal processing can be found in a few recent studies of the initial access problem in mmWave systems (for example, [189]) and of joint power and resource block allocation in vehicular networks [190].

### G. STANDARDIZATION AND OPEN-SOURCE ACTIVITIES

Although AI for radio signal processing has been a hot topic and has received much interest in the last few years, the successful integration of AI for practical applications and standardization activities is still in its infancy. This could be due to several reasons, including the lack of cooperation between academia and industry, competition among high-tech countries, the impact of the coronavirus pandemic, etc. However, certain bodies have already made initial efforts to integrate AI techniques into both existing and future networks. The Telecommunication Standardization Sector of the International Telecommunication Union (ITU) proposed a unified architecture for ML in future networks [191]. The unified architecture is composed of three main blocks, including 1) *a management subsystem*, which allows operators to deploy on-top services without affecting the underlying infrastructure, 2) *an ML pipeline*, which defines a set of logical entities to perform specific functions, and 3) *a closed-loop subsystem*, which helps the ML pipeline adapt to network dynamics. In particular, the ITU architecture needs to satisfy four high-level requirements: use of multisource and correlated data; support for multiple technologies and network layers; support for multilevel, multidomain, and distributed services; and negligible effect on the underlying infrastructure. Very recently, the 3GPP started a study item focusing on traffic characteristics and performance requirements of AI models in 5G. Methods of overcoming particular challenges are specified [192]; for example, achieving AI inference on devices with limited computing and battery capacities, and designing on-board AI inference according to network dynamics.

In addition to standardization, open-source activities play an important role in accelerating the adoption of AI-based signal processing schemes in real wireless communications. However, it is worth noting that the availability of many open-source platforms would limit industrial collaboration and reduce the verification of intelligent signal processing solutions [27].

### H. SUPPORT FOR NEW SERVICES

Although few metrics are used in most AI-enabled schemes to evaluate their performance, emerging applications and new services are expected to have more stringent requirements [1]. For example, smart railway station services could

---

[1]The dataset is available at https://www.deepsig.ai/datasets

be considered as a type of mission-critical services, but they have additional considerations, which may include massive devices with different characteristics, heterogeneous environments, and use cases (e.g., energy charging, emergency, and public safety) [193]. Current AI-enabled signal processing frameworks are often unable to meet the requirements of these services. Embedding AI techniques in signal processing and signaling protocols is a challenging task as the specification is still under development. As a result, partners in academia and industry need to cooperate more closely to accelerate the standardization and specification phase. Moreover, more performance metrics and factors would have to be considered in the design of AI solutions, for example, the training and inference time, reliability, explainability, scalability, simplicity, and security.

## VII. CONCLUSION

In this paper, we presented a survey of state-of-the-art AI solutions for the intelligent processing of radio signals. We first presented a brief overview of AI, ML, FL, and well-known DL architectures such as DNN, CNN, RNN, and DRL. Then, we reviewed applications of AI techniques for three main themes of radio signal processing: modulation classification, signal detection, and channel estimation and beamforming. We discussed these themes according to various classifications and provided necessary background knowledge to enable readers with different types of related expertise to obtain a good understanding of the corresponding topic. We also emphasized a number of challenges that remain unsolved and also offered several suggestions and directions for future research. We hope that this paper can serve as an important reference for both academic and industrial audiences and drive further research and innovation in the domain.

## REFERENCES

[1] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *IEEE Netw.*, vol. 34, no. 3, pp. 134–142, May/Jun. 2020.

[2] C. D. Alwis, A. Kalla, Q.-V. Pham, P. Kumar, K. Dev, W.-J. Hwang, and M. Liyanage, "Survey on 6G frontiers: Trends, applications, requirements, technologies and future research," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 836–886, 2021.

[3] W. Wei and J. M. Mendel, "Maximum-likelihood classification for digital amplitude-phase modulations," *IEEE Trans. Commun.*, vol. 48, no. 2, pp. 189–193, Feb. 2000.

[4] Y. Huang and D. P. Palomar, "Rank-constrained separable semidefinite programming with applications to optimal beamforming," *IEEE Trans. Signal Process.*, vol. 58, no. 2, pp. 664–678, Feb. 2010.

[5] T. Huynh-The, C.-H. Hua, Q.-V. Pham, and D.-S. Kim, "MCNet: An efficient CNN architecture for robust automatic modulation classification," *IEEE Commun. Lett.*, vol. 24, no. 4, pp. 811–815, Apr. 2020.

[6] F. Musumeci, C. Rottondi, A. Nag, I. Macaluso, D. Zibar, M. Ruffini, and M. Tornatore, "An overview on application of machine learning techniques in optical networks," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1383–1408, 2nd Quart., 2019.

[7] B. Karanov, M. Chagnon, F. Thouin, T. A. Eriksson, H. Bülow, D. Lavery, P. Bayvel, and L. Schmalen, "End-to-end deep learning of optical fiber communications," *J. Lightw. Technol.*, vol. 36, no. 20, pp. 4843–4855, Oct. 15, 2018.

[8] 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.

[9] O. T. Demir and E. Björnson, "Channel estimation in massive MIMO under hardware non-linearities: Bayesian methods versus deep learning," *IEEE Open J. Commun. Soc.*, vol. 1, pp. 109–124, 2020.

[10] D. H. N. Nguyen, "Neural network-optimized channel estimator and training signal design for MIMO systems with few-bit ADCs," *IEEE Signal Process. Lett.*, vol. 27, pp. 1370–1374, 2020.

[11] N. T. Nguyen and K. Lee, "Deep learning-aided tabu search detection for large MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 19, no. 6, pp. 4262–4275, Jun. 2020.

[12] A. Askri and G. R.-B. Othman, "DNN assisted sphere decoder," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Paris, France, Jul. 2019, pp. 1172–1176.

[13] T. Wang, C.-K. Wen, H. Wang, F. Gao, T. Jiang, and S. Jin, "Deep learning for wireless physical layer: Opportunities and challenges," *China Commun.*, vol. 14, no. 11, pp. 92–111, Nov. 2017.

[14] T. Erpek, T. J. O'Shea, Y. E. Sagduyu, Y. Shi, and T. C. Clancy, "Deep learning for wireless communications," in *Development and Analysis of Deep Learning Architectures*. Cham, Switzerland: Springer, 2020, pp. 223–266.

[15] Z. Qin, H. Ye, G. Y. Li, and B.-H.-F. Juang, "Deep learning in physical layer communications," *IEEE Wireless Commun.*, vol. 26, no. 2, pp. 93–99, Apr. 2019.

[16] H. He, S. Jin, C.-K. Wen, F. Gao, G. Y. Li, and Z. Xu, "Model-driven deep learning for physical layer communications," *IEEE Wireless Commun.*, vol. 26, no. 5, pp. 77–83, Oct. 2019.

[17] A. Balatsoukas-Stimming and C. Studer, "Deep unfolding for communications systems: A survey and some new directions," in *Proc. IEEE Int. Workshop Signal Process. Syst. (SiPS)*. Nanjing, China: IEEE, Oct. 2019, pp. 266–271.

[18] A. Zappone, M. Di Renzo, and M. Debbah, "Wireless networks design in the era of deep learning: Model-based, AI-based, or both?" *IEEE Trans. Commun.*, vol. 67, no. 10, pp. 7331–7376, Oct. 2019.

[19] R. Zhou, F. Liu, and C. W. Gravelle, "Deep learning for modulation recognition: A survey with a demonstration," *IEEE Access*, vol. 8, pp. 67366–67376, 2020.

[20] S. K. Sharma and X. Wang, "Toward massive machine type communications in ultra-dense cellular IoT networks: Current issues and machine learning-assisted solutions," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 426–471, 1st Quart., 2020.

[21] F. Hussain, R. Hussain, S. A. Hassan, and E. Hossain, "Machine learning in IoT security: Current solutions and future challenges," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1686–1721, 3rd Quart., 2020.

[22] R. Gu, Z. Yang, and Y. Ji, "Machine learning for intelligent optical networks: A comprehensive survey," *J. Netw. Comput. Appl.*, vol. 157, May 2020, Art. no. 102576.

[23] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, and C. Wang, "Machine learning and deep learning methods for cybersecurity," *IEEE Access*, vol. 6, pp. 35365–35381, 2018.

[24] P. Mishra, V. Varadharajan, U. Tupakula, and E. S. Pilli, "A detailed investigation and analysis of using machine learning techniques for intrusion detection," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 686–728, 1st Quart., 2019.

[25] Z. M. Fadlullah, F. Tang, B. Mao, N. Kato, O. Akashi, T. Inoue, and K. Mizutani, "State-of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2432–2455, 4th Quart., 2017.

[26] F. Pacheco, E. Exposito, M. Gineste, C. Baudoin, and J. Aguilar, "Towards the deployment of machine learning solutions in network traffic classification: A systematic survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1988–2014, 2nd Quart., 2019.

[27] Q.-V. Pham, F. Fang, V. N. Ha, M. J. Piran, M. Le, L. B. Le, W.-J. Hwang, and Z. Ding, "A survey of multi-access edge computing in 5G and beyond: Fundamentals, technology integration, and state-of-the-art," *IEEE Access*, vol. 8, pp. 116974–117017, 2020.

[28] J. Xie, F. R. Yu, T. Huang, R. Xie, J. Liu, C. Wang, and Y. Liu, "A survey of machine learning techniques applied to software defined networking (SDN): Research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 393–430, 1st Quart., 2019.

[29] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Artificial neural networks-based machine learning for wireless networks: A tutorial," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3039–3071, 4th Quart., 2019.

[30] Q. Mao, F. Hu, and Q. Hao, "Deep learning for intelligent wireless networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2595–2621, 4th Quart., 2018.

[31] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2224–2287, 3rd Quart., 2019.

[32] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y.-C. Liang, and D. I. Kim, "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3133–3174, 4th Quart., 2019.

[33] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 2031–2063, 3rd Quart., 2020.

[34] J. Wang, C. Jiang, H. Zhang, Y. Ren, K.-C. Chen, and L. Hanzo, "Thirty years of machine learning: The road to Pareto-optimal wireless networks," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 3, pp. 1472–1514, 3rd Quart., 2020.

[35] Q.-V. Pham, D. C. Nguyen, S. Mirjalili, D. T. Hoang, D. N. Nguyen, P. N. Pathirana, and W.-J. Hwang, "Swarm intelligence for next-generation networks: Recent advances and applications," *J. Netw. Comput. Appl.*, Jul. 2021. [Online]. Available: https://arxiv.org/abs/2007.15221

[36] T. Erpek, T. J. O'Shea, Y. E. Sagduyu, Y. Shi, and T. C. Clancy, *Deep Learning for Wireless Communications*. Cham, Switzerland: Springer, 2020, pp. 223–266.

[37] F. Zafari, A. Gkelias, and K. K. Leung, "A survey of indoor localization systems and technologies," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2568–2599, 3rd Quart., 2019.

[38] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.

[39] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

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

[41] F. Agostinelli, M. D. Hoffman, P. J. Sadowski, and P. Baldi, "Learning activation functions to improve deep neural networks," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, 2015. [Online]. Available: https://iclr.cc/archive/www/doku.php%3Fid=iclr2015:main.html and https://arxiv.org/pdf/1412.6830.pdf

[42] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," in *Proc. 6th Int. Conf. Learn. Represent. (ICLR)*, Vancouver, BC, Canada, 2018. [Online]. Available: https://research.google/pubs/pub46503/ and https://arxiv.org/abs/1710.05941v2

[43] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Netw.*, vol. 4, no. 2, pp. 251–257, 1991.

[44] Q.-V. Pham, T. Huynh-The, M. Alazab, J. Zhao, and W.-J. Hwang, "Sum-rate maximization for UAV-assisted visible light communications using NOMA: Swarm intelligence meets machine learning," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 10375–10387, Oct. 2020.

[45] V. K. Ojha, A. Abraham, and V. Snášel, "Metaheuristic design of feedforward neural networks: A review of two decades of research," *Eng. Appl. Artif. Intell.*, vol. 60, pp. 97–116, Apr. 2017.

[46] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. 36th Int. Conf. Mach. Learn.*, Long Beach, CA, USA, vol. 97, 2019, pp. 6105–6114.

[47] L. Ale, N. Zhang, H. Wu, D. Chen, and T. Han, "Online proactive caching in mobile edge computing using bidirectional deep recurrent neural network," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5520–5530, Jun. 2019.

[48] M. T. Hoang, B. Yuen, X. Dong, T. Lu, R. Westendorp, and K. Reddy, "Recurrent neural networks for accurate RSSI indoor localization," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10639–10651, Dec. 2019.

[49] N. Samuel, T. Diskin, and A. Wiesel, "Deep MIMO detection," in *Proc. IEEE 18th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jul. 2017, pp. 1–5.

[50] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, Apr. 2017.

[51] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad, "State-of-the-art in artificial neural network applications: A survey," *Heliyon*, vol. 4, no. 11, Nov. 2018, Art. no. e00938.

[52] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, and S. Petersen, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[53] (2017). *Federated Learning: Collaborative Machine Learning Without Centralized Training Data*. [Online]. Available: https://ai.googleblog.com/2017/04/federated-learning-collaborative.html

[54] Q.-V. Pham, M. Zeng, R. Ruby, T. Huynh-The, and W.-J. Hwang, "UAV communications for sustainable federated learning," *IEEE Trans. Veh. Technol.*, vol. 70, no. 4, pp. 3944–3948, Apr. 2021.

[55] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–19, Jan. 2019.

[56] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen, "Convergence of edge computing and deep learning: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 869–904, 2nd Quart., 2020.

[57] M. Naderpour and H. K. Bizaki, "Low overhead NOMA receiver with automatic modulation classification techniques," *IET Commun.*, vol. 14, no. 5, pp. 768–774, Mar. 2020.

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

[59] 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.

[60] F. Hameed, O. Dobre, and D. Popescu, "On the likelihood-based approach to modulation classification," *IEEE Trans. Wireless Commun.*, vol. 8, no. 12, pp. 5884–5892, Dec. 2009.

[61] Z. Zhu and A. K. Nandi, "Blind digital modulation classification using minimum distance centroid estimator and non-parametric likelihood function," *IEEE Trans. Wireless Commun.*, vol. 13, no. 8, pp. 4483–4494, Aug. 2014.

[62] E. Soltanmohammadi and M. Naraghi-Pour, "Blind modulation classification over fading channels using expectation-maximization," *IEEE Commun. Lett.*, vol. 17, no. 9, pp. 1692–1695, Sep. 2013.

[63] C.-H. Hua, T. Huynh-The, K. Kim, S.-Y. Yu, T. Le-Tien, G. H. Park, J. Bang, W. A. Khan, S.-H. Bae, and S. Lee, "Bimodal learning via trilogy of skip-connection deep networks for diabetic retinopathy risk progression identification," *Int. J. Med. Informat.*, vol. 132, Dec. 2019, Art. no. 103926.

[64] T. Huynh-The, C.-H. Hua, and D.-S. Kim, "Encoding pose features to images with data augmentation for 3-D action recognition," *IEEE Trans. Ind. Informat.*, vol. 16, no. 5, pp. 3100–3111, May 2020.

[65] T. Huynh-The, C.-H. Hua, T.-T. Ngo, and D.-S. Kim, "Image representation of pose-transition feature for 3D skeleton-based action recognition," *Inf. Sci.*, vol. 513, pp. 112–126, Mar. 2020.

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

[67] J. Zhang, F. Wang, Z. Zhong, and D. Cabric, "Cooperative multiuser modulation classification in multipath channels via expectation-maximization," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Paris, France, May 2017, pp. 1–6.

[68] J. Zhang, D. Cabric, F. Wang, and Z. Zhong, "Cooperative modulation classification for multipath fading channels via expectation-maximization," *IEEE Trans. Wireless Commun.*, vol. 16, no. 10, pp. 6698–6711, Oct. 2017.

[69] J. Zhang, F. Wang, Z. Zhong, and S. Wang, "Continuous phase modulation classification via Baum-Welch algorithm," *IEEE Commun. Lett.*, vol. 22, no. 7, pp. 1390–1393, Jul. 2018.

[70] J. Zheng and Y. Lv, "Likelihood-based automatic modulation classification in OFDM with index modulation," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8192–8204, Sep. 2018.

[71] J. Tian, Y. Pei, Y.-D. Huang, and Y.-C. Liang, "Modulation-constrained clustering approach to blind modulation classification for MIMO systems," *IEEE Trans. Cognit. Commun. Netw.*, vol. 4, no. 4, pp. 894–907, Dec. 2018.

[72] W. Chen, Z. Xie, L. Ma, J. Liu, and X. Liang, "A faster maximum-likelihood modulation classification in flat fading non-Gaussian channels," *IEEE Commun. Lett.*, vol. 23, no. 3, pp. 454–457, Mar. 2019.

[73] A. O. A. Salam, R. E. Sheriff, Y.-F. Hu, S. R. Al-Araji, and K. Mezher, "Automatic modulation classification using interacting multiple model Kalman filter for channel estimation," *IEEE Trans. Veh. Technol.*, vol. 68, no. 9, pp. 8928–8939, Sep. 2019.

[74] Y. Wang, J. Wang, W. Zhang, J. Yang, and G. Gui, "Deep learning-based cooperative automatic modulation classification method for MIMO systems," *IEEE Trans. Veh. Technol.*, vol. 69, no. 4, pp. 4575–4579, Apr. 2020.

[75] 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.

[76] H. Wu, Y. Li, L. Zhou, and J. Meng, "Convolutional neural network and multi-feature fusion for automatic modulation classification," *Electron. Lett.*, vol. 55, no. 16, pp. 895–897, Aug. 2019.

[77] Y. Wang, M. Liu, J. Yang, and G. Gui, "Data-driven deep learning for automatic modulation recognition in cognitive radios," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 4074–4077, Apr. 2019.

[78] 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.

[79] 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.

[80] L. Han, F. Gao, Z. Li, and O. A. Dobre, "Low complexity automatic modulation classification based on order-statistics," *IEEE Trans. Wireless Commun.*, vol. 16, no. 1, pp. 400–411, Jan. 2017.

[81] S. Huang, Y. Yao, Z. Wei, Z. Feng, and P. Zhang, "Automatic modulation classification of overlapped sources using multiple cumulants," *IEEE Trans. Veh. Technol.*, vol. 66, no. 7, pp. 6089–6101, Jul. 2017.

[82] S. Majhi, R. Gupta, W. Xiang, and S. Glisic, "Hierarchical hypothesis and feature-based blind modulation classification for linearly modulated signals," *IEEE Trans. Veh. Technol.*, vol. 66, no. 12, pp. 11057–11069, Dec. 2017.

[83] M. Abu-Romoh, A. Aboutaleb, and Z. Rezki, "Automatic modulation classification using moments and likelihood maximization," *IEEE Commun. Lett.*, vol. 22, no. 5, pp. 938–941, May 2018.

[84] M. Abdelbar, W. H. Tranter, and T. Bose, "Cooperative cumulants-based modulation classification in distributed networks," *IEEE Trans. Cognit. Commun. Netw.*, vol. 4, no. 3, pp. 446–461, Sep. 2018.

[85] K. Zhang, E. L. Xu, H. Zhang, Z. Feng, and S. Cui, "Data driven automatic modulation classification via dictionary learning," *IEEE Wireless Commun. Lett.*, vol. 7, no. 4, pp. 586–589, Aug. 2018.

[86] W. Xiong, P. Bogdanov, and M. Zheleva, "Robust and efficient modulation recognition based on local sequential IQ features," in *Proc. IEEE Conf. Comput. Commun. (IEEE INFOCOM)*, Apr. 2019, pp. 1612–1620.

[87] R. Gupta, S. Kumar, and S. Majhi, "Blind modulation classification for OFDM in the presence of timing, frequency, and phase offsets," in *Proc. IEEE 90th Veh. Technol. Conf. (VTC-Fall)*, Honolulu, HI, USA, Sep. 2019, pp. 1–5.

[88] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[89] W. Lee, O. Jo, and M. Kim, "Intelligent resource allocation in wireless communications systems," *IEEE Commun. Mag.*, vol. 58, no. 1, pp. 100–105, Jan. 2020.

[90] M. H. Shah and X. Dang, "Classification of spectrally efficient constant envelope modulations based on radial basis function network and deep learning," *IEEE Commun. Lett.*, vol. 23, no. 9, pp. 1529–1533, Sep. 2019.

[91] A. Ali and F. Yangyu, "*k*-sparse autoencoder-based automatic modulation classification with low complexity," *IEEE Commun. Lett.*, vol. 21, no. 10, pp. 2162–2165, Oct. 2017.

[92] M. H. Shah and X. Dang, "Novel feature selection method using Bhattacharyya distance for neural networks based automatic modulation classification," *IEEE Signal Process. Lett.*, vol. 27, pp. 106–110, 2020.

[93] 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.

[94] S. Zheng, P. Qi, S. Chen, and X. Yang, "Fusion methods for CNN-based automatic modulation classification," *IEEE Access*, vol. 7, pp. 66496–66504, 2019.

[95] S. Huang, L. Chai, Z. Li, D. Zhang, Y. Yao, Y. Zhang, and Z. Feng, "Automatic modulation classification using compressive convolutional neural network," *IEEE Access*, vol. 7, pp. 79636–79643, 2019.

[96] S. Huang, Y. Jiang, Y. Gao, Z. Feng, and P. Zhang, "Automatic modulation classification using contrastive fully convolutional network," *IEEE Wireless Commun. Lett.*, vol. 8, no. 4, pp. 1044–1047, Aug. 2019.

[97] Y. Zeng, M. Zhang, F. Han, Y. Gong, and J. Zhang, "Spectrum analysis and convolutional neural network for automatic modulation recognition," *IEEE Wireless Commun. Lett.*, vol. 8, no. 3, pp. 929–932, Jun. 2019.

[98] 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.

[99] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, 2015. [Online]. Available: https://iclr.cc/archive/www/doku.php%3Fid=iclr2015:main.html and https://arxiv.org/abs/1409.1556

[100] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.

[101] T. Huynh-The, C.-H. Hua, J.-W. Kim, S.-H. Kim, and D.-S. Kim, "Exploiting a low-cost CNN with skip connection for robust automatic modulation classification," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Seoul, South Korea, May 2020, pp. 1–6.

[102] G. B. Tunze, T. Huynh-The, J.-M. Lee, and D.-S. Kim, "Multi-shuffled convolutional blocks for low-complex modulation recognition," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Jeju, South Korea, Oct. 2020, pp. 939–942.

[103] T. Huynh-The, V.-S. Doan, C.-H. Hua, Q.-V. Pham, and D.-S. Kim, "Chain-Net: Learning deep model for modulation classification under synthetic channel impairment," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Taipei, Taiwan, Dec. 2020, pp. 1–6.

[104] G. B. Tunze, T. Huynh-The, J.-M. Lee, and D.-S. Kim, "Sparsely connected CNN for efficient automatic modulation recognition," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 15557–15568, Dec. 2020.

[105] T. Huynh-The, C.-H. Hua, V.-S. Doan, and D.-S. Kim, "Accurate modulation classification with reusable-feature convolutional neural network," in *Proc. IEEE 8th Int. Conf. Commun. Electron. (ICCE)*, Phu Quoc Island, Vietnam, Jan. 2021, pp. 12–17.

[106] T. Huynh-The, C.-H. Hua, V.-S. Doan, Q.-V. Pham, T.-V. Nguyen, and D.-S. Kim, "Deep learning for constellation-based modulation classification under multipath fading channels," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Jeju, South Korea, Oct. 2020, pp. 300–304.

[107] V.-S. Doan, T. Huynh-The, C.-H. Hua, Q.-V. Pham, and D.-S. Kim, "Learning constellation map with deep CNN for accurate modulation recognition," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Taipei, Taiwan, Dec. 2020, pp. 1–6.

[108] B. Hassibi and H. Vikalo, "On the sphere-decoding algorithm I. Expected complexity," *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 2806–2818, Aug. 2005.

[109] N. T. Nguyen, K. Lee, and H. Dai, "QR-decomposition-aided tabu search detection for large MIMO systems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 4857–4870, May 2019.

[110] N. T. Nguyen and K. Lee, "Groupwise neighbor examination for tabu search detection in large MIMO systems," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 1136–1140, Jan. 2020.

[111] M. A. Albreem, M. Juntti, and S. Shahabuddin, "Massive MIMO detection techniques: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3109–3132, 4th Quart., 2019.

[112] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "A model-driven deep learning network for MIMO detection," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Anaheim, CA, USA, Nov. 2018, pp. 584–588.

[113] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "Model-driven deep learning for MIMO detection," *IEEE Trans. Signal Process.*, vol. 68, pp. 1702–1715, 2020.

[114] M. Khani, M. Alizadeh, J. Hoydis, and P. Fleming, "Adaptive neural signal detection for massive MIMO," *IEEE Trans. Wireless Commun.*, vol. 19, no. 8, pp. 5635–5648, Aug. 2020.

[115] S. Xue, Y. Ma, N. Yi, and T. E. Dodgson, "A modular neural network based deep learning approach for MIMO signal detection," 2020, *arXiv:2004.00404*. [Online]. Available: http://arxiv.org/abs/2004.00404

[116] M.-S. Baek, S. Kwak, J.-Y. Jung, H. M. Kim, and D.-J. Choi, "Implementation methodologies of deep learning-based signal detection for conventional MIMO transmitters," *IEEE Trans. Broadcast.*, vol. 65, no. 3, pp. 636–642, Sep. 2019.

[117] X. Liu and Y. Li, "Deep MIMO detection based on belief propagation," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Guangzhou, China, Nov. 2018, pp. 1–5.

[118] N. Samuel, T. Diskin, and A. Wiesel, "Learning to detect," *IEEE Trans. Signal Process.*, vol. 67, no. 10, pp. 2554–2564, May 2019.

[119] V. Corlay, J. J. Boutros, P. Ciblat, and L. Brunel, "Multilevel MIMO detection with deep learning," in *Proc. 52nd Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, USA, Oct. 2018, pp. 1805–1809.

[120] G. Gao, C. Dong, and K. Niu, "Sparsely connected neural network for massive MIMO detection," in *Proc. IEEE 4th Int. Conf. Comput. Commun. (ICCC)*, Salamanca, Spain, Dec. 2018, pp. 397–402.

[121] N. Shlezinger, R. Fu, and Y. C. Eldar, "Deep soft interference cancellation for MIMO detection," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Barcelona, Spain, May 2020, pp. 8881–8885.

[122] N. Farsad and A. Goldsmith, "Detection algorithms for communication systems using deep learning," 2017, *arXiv:1705.08044*. [Online]. Available: http://arxiv.org/abs/1705.08044

[123] X. Tan, Z. Zhong, Z. Zhang, X. You, and C. Zhang, "Low-complexity message passing MIMO detection algorithm with deep neural network," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Anaheim, CA, USA, Nov. 2018, pp. 559–563.

[124] S. Xue, Y. Ma, N. Yi, and R. Tafazolli, "Unsupervised deep learning for MU-SIMO joint transmitter and noncoherent receiver design," *IEEE Wireless Commun. Lett.*, vol. 8, no. 1, pp. 177–180, Feb. 2019.

[125] 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.

[126] X. Gao, S. Jin, C.-K. Wen, and G. Y. Li, "ComNet: Combination of deep learning and expert knowledge in OFDM receivers," *IEEE Commun. Lett.*, vol. 22, no. 12, pp. 2627–2630, Dec. 2018.

[127] R. Yao, S. Wang, X. Zuo, J. Xu, and N. Qi, "Deep learning aided signal detection in OFDM systems with time-varying channels," in *Proc. IEEE Pacific Rim Conf. Commun., Comput. Signal Process. (PACRIM)*, Victoria, BC, Canada, Aug. 2019, pp. 1–5.

[128] J. Xia, K. He, W. Xu, S. Zhang, L. Fan, and G. K. Karagiannidis, "A MIMO detector with deep learning in the presence of correlated interference," *IEEE Trans. Veh. Technol.*, vol. 69, no. 4, pp. 4492–4497, Apr. 2020.

[129] K. He, Z. Wang, W. Huang, D. Deng, J. Xia, and L. Fan, "Generic deep learning-based linear detectors for MIMO systems over correlated noise environments," *IEEE Access*, vol. 8, pp. 29922–29929, 2020.

[130] C. Fan, X. Yuan, and Y.-J. Zhang, "CNN-based signal detection for banded linear systems," *IEEE Trans. Wireless Commun.*, vol. 18, no. 9, pp. 4394–4407, Sep. 2019.

[131] X. Yan, F. Long, J. Wang, N. Fu, W. Ou, and B. Liu, "Signal detection of MIMO-OFDM system based on auto encoder and extreme learning machine," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Anchorage, Alaska, May 2017, pp. 1602–1606.

[132] J. Liu, K. Mei, X. Zhang, D. Ma, and J. Wei, "Online extreme learning machine-based channel estimation and equalization for OFDM systems," *IEEE Commun. Lett.*, vol. 23, no. 7, pp. 1276–1279, Jul. 2019.

[133] E. Balevi and J. G. Andrews, "One-bit OFDM receivers via deep learning," *IEEE Trans. Commun.*, vol. 67, no. 6, pp. 4326–4336, Jun. 2019.

[134] C.-B. Ha and H.-K. Song, "Signal detection scheme based on adaptive ensemble deep learning model," *IEEE Access*, vol. 6, pp. 21342–21349, 2018.

[135] Q. Huang, C. Zhao, M. Jiang, X. Li, and J. Liang, "Cascade-net: A new deep learning architecture for OFDM detection," 2018, *arXiv:1812.00023*. [Online]. Available: http://arxiv.org/abs/1812.00023

[136] M. Mohammadkarimi, M. Mehrabi, M. Ardakani, and Y. Jing, "Deep learning-based sphere decoding," *IEEE Trans. Wireless Commun.*, vol. 18, no. 9, pp. 4368–4378, Sep. 2019.

[137] D. Weon and K. Lee, "Learning-aided deep path prediction for sphere decoding in large MIMO systems," *IEEE Access*, vol. 8, pp. 70870–70877, 2020.

[138] N. N. Thanh, L. Kyungchun, and D. Huaiyu, "Application of deep learning to sphere decoding for large MIMO systems," *IEEE Trans. Wireless Commun.*, early access, May 6, 2021, doi: 10.1109/TWC.2021.3076527.

[139] T. Lin and Y. Zhu, "Beamforming design for large-scale antenna arrays using deep learning," *IEEE Wireless Commun. Lett.*, vol. 9, no. 1, pp. 103–107, Jan. 2020.

[140] H. Huang, W. Xia, J. Xiong, J. Yang, G. Zheng, and X. Zhu, "Unsupervised learning-based fast beamforming design for downlink MIMO," *IEEE Access*, vol. 7, pp. 7599–7605, 2019.

[141] A. Alkhateeb, S. Alex, P. Varkey, Y. Li, Q. Qu, and D. Tujkovic, "Deep learning coordinated beamforming for highly-mobile millimeter wave systems," *IEEE Access*, vol. 6, pp. 37328–37348, 2018.

[142] C. Anton-Haro and X. Mestre, "Learning and data-driven beam selection for mmWave communications: An angle of arrival-based approach," *IEEE Access*, vol. 7, pp. 20404–20415, 2019.

[143] M. M. Ramón, N. Xu, and C. G. Christodoulou, "Beamforming using support vector machines," *IEEE Antennas Wireless Propag. Lett.*, vol. 4, pp. 439–442, 2005.

[144] Y. Long, Z. Chen, J. Fang, and C. Tellambura, "Data-driven-based analog beam selection for hybrid beamforming under mm-wave channels," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 2, pp. 340–352, May 2018.

[145] X. Li, X. Yu, T. Sun, J. Guo, and J. Zhang, "Joint scheduling and deep learning-based beamforming for FD-MIMO systems over correlated Rician fading," *IEEE Access*, vol. 7, pp. 118297–118309, 2019.

[146] P. Zhou, X. Fang, X. Wang, Y. Long, R. He, and X. Han, "Deep learning-based beam management and interference coordination in dense mmWave networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 592–603, Jan. 2019.

[147] W. Xia, G. Zheng, Y. Zhu, J. Zhang, J. Wang, and A. P. Petropulu, "A deep learning framework for optimization of MISO downlink beamforming," *IEEE Trans. Commun.*, vol. 68, no. 3, pp. 1866–1880, Mar. 2020.

[148] 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.

[149] A. Taha, M. Alrabeiah, and A. Alkhateeb, "Deep learning for large intelligent surfaces in millimeter wave and massive MIMO systems," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Waikoloa, HI, USA, Dec. 2019, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9013256

[150] J. Gao, C. Zhong, X. Chen, H. Lin, and Z. Zhang, "Unsupervised learning for passive beamforming," *IEEE Commun. Lett.*, vol. 24, no. 5, pp. 1052–1056, May 2020.

[151] T. Maksymyuk, J. Gazda, O. Yaremko, and D. Nevinskiy, "Deep learning based massive MIMO beamforming for 5G mobile network," in *Proc. IEEE 4th Int. Symp. Wireless Syst. Int. Conf. Intell. Data Acquisition Adv. Comput. Syst. (IDAACS-SWS)*, Lviv, Ukraine, Sep. 2018, pp. 241–244.

[152] F. B. Mismar, B. L. Evans, and A. Alkhateeb, "Deep reinforcement learning for 5G networks: Joint beamforming, power control, and interference coordination," *IEEE Trans. Commun.*, vol. 68, no. 3, pp. 1581–1592, Mar. 2020.

[153] Q. Wang, K. Feng, X. Li, and S. Jin, "PrecoderNet: Hybrid beamforming for millimeter wave systems with deep reinforcement learning," *IEEE Wireless Commun. Lett.*, vol. 9, no. 10, pp. 1677–1681, Oct. 2020.

[154] Y. Sun, M. Peng, and S. Mao, "Deep reinforcement learning-based mode selection and resource management for green fog radio access networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1960–1971, Apr. 2019.

[155] J. Moon, O. Simeone, S.-H. Park, and I. Lee, "Online reinforcement learning of X-Haul content delivery mode in fog radio access networks," *IEEE Signal Process. Lett.*, vol. 26, no. 10, pp. 1451–1455, Oct. 2019.

[156] K. Feng, Q. Wang, X. Li, and C.-K. Wen, "Deep reinforcement learning based intelligent reflecting surface optimization for MISO communication systems," *IEEE Wireless Commun. Lett.*, vol. 9, no. 5, pp. 745–749, May 2020.

[157] A. Taha, Y. Zhang, F. B. Mismar, and A. Alkhateeb, "Deep reinforcement learning for intelligent reflecting surfaces: Towards standalone operation," in *Proc. IEEE 21st Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Atlanta, GA, USA, May 2020, pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9154301

[158] K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated learning via over-the-air computation," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 2022–2035, Mar. 2020.

[159] A. M. Elbir and S. Coleri, "Federated learning for hybrid beamforming in mm-wave massive MIMO," *IEEE Commun. Lett.*, vol. 24, no. 12, pp. 2795–2799, Dec. 2020.

[160] D. Ma, L. Li, H. Ren, D. Wang, X. Li, and Z. Han, "Distributed rate optimization for intelligent reflecting surface with federated learning," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, Dublin, Ireland, Jun. 2020.

[161] H. Huang, Y. Song, J. Yang, G. Gui, and F. Adachi, "Deep-learning-based millimeter-wave massive MIMO for hybrid precoding," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 3027–3032, Mar. 2019.

[162] C. Huang, R. Mo, and C. Yuen, "Reconfigurable intelligent surface assisted multiuser MISO systems exploiting deep reinforcement learning," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 8, pp. 1839–1850, Aug. 2020.

[163] M. Biguesh and A. B. Gershman, "Training-based MIMO channel estimation: A study of estimator tradeoffs and optimal training signals," *IEEE Trans. Signal Process.*, vol. 54, no. 3, pp. 884–893, Mar. 2006.

[164] S. Gao, P. Dong, Z. Pan, and G. Y. Li, "Deep learning based channel estimation for massive MIMO with mixed-resolution ADCs," *IEEE Commun. Lett.*, vol. 23, no. 11, pp. 1989–1993, Nov. 2019.

[165] H. Sun, Z. Zhao, X. Fu, and M. Hong, "Limited feedback double directional massive MIMO channel estimation: From low-rank modeling to deep learning," in *Proc. IEEE 19th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Kalamata, Greece, Jun. 2018, pp. 1–5.

[166] C.-J. Chun, J.-M. Kang, and I.-M. Kim, "Deep learning-based channel estimation for massive MIMO systems," *IEEE Wireless Commun. Lett.*, vol. 8, no. 4, pp. 1228–1231, Aug. 2019.

[167] Y. Liao, Y. Hua, X. Dai, H. Yao, and X. Yang, "ChanEstNet: A deep learning based channel estimation for high-speed scenarios," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Shanghai, China, May 2019, pp. 1–6.

[168] P. Dong, H. Zhang, G. Y. Li, I. S. Gaspar, and N. NaderiAlizadeh, "Deep CNN-based channel estimation for mmWave massive MIMO systems," *IEEE J. Sel. Topics Signal Process.*, vol. 13, no. 5, pp. 989–1000, Sep. 2019.

[169] M. Soltani, V. Pourahmadi, A. Mirzaei, and H. Sheikhzadeh, "Deep learning-based channel estimation," *IEEE Commun. Lett.*, vol. 23, no. 4, pp. 652–655, Apr. 2019.

[170] J.-M. Kang, C.-J. Chun, and I.-M. Kim, "Deep-learning-based channel estimation for wireless energy transfer," *IEEE Commun. Lett.*, vol. 22, no. 11, pp. 2310–2313, Nov. 2018.

[171] H. Huang, J. Yang, H. Huang, Y. Song, and G. Gui, "Deep learning for super-resolution channel estimation and DOA estimation based massive MIMO system," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8549–8560, Sep. 2018.

[172] W. Ma, C. Qi, Z. Zhang, and J. Cheng, "Sparse channel estimation and hybrid precoding using deep learning for millimeter wave massive MIMO," *IEEE Trans. Commun.*, vol. 68, no. 5, pp. 2838–2849, May 2020.

[173] Y. Jin, J. Zhang, S. Jin, and B. Ai, "Channel estimation for cell-free mmWave massive MIMO through deep learning," *IEEE Trans. Veh. Technol.*, vol. 68, no. 10, pp. 10325–10329, Oct. 2019.

[174] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "Deep learning-based channel estimation for beamspace mmWave massive MIMO systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 5, pp. 852–855, Oct. 2018.

[175] C.-K. Wen, W.-T. Shih, and S. Jin, "Deep learning for massive MIMO CSI feedback," *IEEE Wireless Commun. Lett.*, vol. 7, no. 5, pp. 748–751, Oct. 2018.

[176] E. Balevi, A. Doshi, and J. G. Andrews, "Massive MIMO channel estimation with an untrained deep neural network," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 2079–2090, Mar. 2020.

[177] D. Neumann, T. Wiese, and W. Utschick, "Learning the MMSE channel estimator," *IEEE Trans. Signal Process.*, vol. 66, no. 11, pp. 2905–2917, Jun. 2018.

[178] Y. Yang, F. Gao, X. Ma, and S. Zhang, "Deep learning-based channel estimation for doubly selective fading channels," *IEEE Access*, vol. 7, pp. 36579–36589, 2019.

[179] N. Kato, B. Mao, F. Tang, Y. Kawamoto, and J. Liu, "Ten challenges in advancing machine learning technologies toward 6G," *IEEE Wireless Commun.*, vol. 27, no. 3, pp. 96–103, Jun. 2020.

[180] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y.-J.-A. Zhang, "The roadmap to 6G: AI empowered wireless networks," *IEEE Commun. Mag.*, vol. 57, no. 8, pp. 84–90, Aug. 2019.

[181] S. M. Zafaruddin, I. Bistritz, A. Leshem, and D. Niyato, "Distributed learning for channel allocation over a shared spectrum," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2337–2349, Oct. 2019.

[182] Y. Cheng, D. Wang, P. Zhou, and T. Zhang, "Model compression and acceleration for deep neural networks: The principles, progress, and challenges," *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 126–136, Jan. 2018.

[183] B. L. Deng, G. Li, S. Han, L. Shi, and Y. Xie, "Model compression and hardware acceleration for neural networks: A comprehensive survey," *Proc. IEEE*, vol. 108, no. 4, pp. 485–532, Apr. 2020.

[184] K. Yang, Y. Shi, W. Yu, and Z. Ding, "Energy-efficient processing and robust wireless cooperative transmission for edge inference," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 9456–9470, Oct. 2020.

[185] S. Hua, Y. Zhou, K. Yang, Y. Shi, and K. Wang, "Reconfigurable intelligent surface for green edge inference," *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 2, pp. 964–979, Jun. 2021.

[186] (2020). *ViWi: A Deep Learning Dataset Framework for Vision-Aided Wireless Communications*. [Online]. Available: https://www.viwi-dataset.net/

[187] (2020). *IEEE CTW 2020 Data Competition: Self-Supervised Learning for User Localization*. [Online]. Available: https://ctw2020.ieee-ctw.org/data-competition/

[188] K. Bu, Y. He, X. Jing, and J. Han, "Adversarial transfer learning for deep learning based automatic modulation classification," *IEEE Signal Process. Lett.*, vol. 27, pp. 880–884, 2020.

[189] S.-E. Chiu, N. Ronquillo, and T. Javidi, "Active learning and CSI acquisition for mmWave initial alignment," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 11, pp. 2474–2489, Nov. 2019.

[190] M. K. Abdel-Aziz, S. Samarakoon, M. Bennis, and W. Saad, "Ultra-reliable and low-latency vehicular communication: An active learning approach," *IEEE Commun. Lett.*, vol. 24, no. 2, pp. 367–370, Feb. 2020.

[191] ITU-T, *Unified Architecture for Machine Learning in 5G and Future Networks*, document TS ITU-T FG-ML5G-ARC5G, 2019.

[192] "3GPP work item description: Study on traffic characteristics and performance requirements for AI/ML model transfer in 5GS," 3GPP, Tech. Rep., 2020. [Online]. Available: https://www.3gpp.org/DynaReport/WiCr–860009.htm and https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3721

[193] "3GPP work item description: Study on supporting of railway smart station services," 3GPP, Tech. Rep., 2020. [Online]. Available: https://www.3gpp.org/DynaReport/WiCr–850044.htm and https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3699

**QUOC-VIET PHAM** (Member, IEEE) received the B.S. degree in electronics and telecommunications engineering from the Hanoi University of Science and Technology, Vietnam, in 2013, and the Ph.D. degree in telecommunications engineering from Inje University, South Korea, in 2017.

From September 2017 to December 2019, he was with Kyung Hee University, Changwon National University, and Inje University, on various academic positions. He is currently a Research Professor with Pusan National University, South Korea. His research interests include convex optimization, game theory, and machine learning to analyze and optimize edge/cloud computing systems and future wireless systems. He received the Best Ph.D. Dissertation Award in engineering from Inje University, in 2017. He has been granted the Korea NRF Funding for outstanding young researchers for the term 2019–2023. He received the Top Reviewer Award from the IEEE Transactions on Vehicular Technology, in 2020. He is an Editor of the *Journal of Network and Computer Applications* (Elsevier) and a Lead Guest Editor of the IEEE Internet of Things Journal.

**NHAN THANH NGUYEN** (Member, IEEE) received the B.S. degree in electronics and communications engineering from the Hanoi University of Science and Technology, Hanoi, Vietnam, in 2014, and the M.S. and Ph.D. degrees in electrical and information engineering from the Seoul National University of Science and Technology, Seoul, South Korea, in 2017 and 2020, respectively. From October 2019 to March 2020, he was a Visiting Researcher with North Carolina State University, Raleigh, NC, USA. Since September 2020, he has been a Research Fellow with the Centre for Wireless Communication, University of Oulu, Oulu, Finland. His research interests include signal processing, optimization, and applied machine learning for wireless communication with a focus on massive MIMO systems. He was a recipient of the Outstanding M.S. and Ph.D. Thesis Awards from the Seoul National University of Science and Technology, in 2017 and 2020, respectively.

**THIEN HUYNH-THE** (Member, IEEE) received the B.S. degree in electronics and telecommunication engineering from the Ho Chi Minh City University of Technology and Education, Vietnam, in 2011, and the Ph.D. degree in computer science and engineering from Kyung Hee University (KHU), South Korea, in 2018. He is currently a Postdoctoral Research Fellow with the ICT Convergence Research Center, Kumoh National Institute of Technology, South Korea. His current research interests include radio signal processing, digital image processing, computer vision, machine learning, and deep learning. He was awarded with the Superior Thesis Prize from KHU.

**LONG BAO LE** (Senior Member, IEEE) received the B.Eng. degree in electrical engineering from the Ho Chi Minh City University of Technology, Vietnam, in 1999, the M.Eng. degree in telecommunications from the Asian Institute of Technology, Thailand, in 2002, and the Ph.D. degree in electrical engineering from the University of Manitoba, Canada, in 2007. He was a Postdoctoral Researcher with the Massachusetts Institute of Technology, from 2008 to 2010, and the University of Waterloo, from 2007 to 2008. Since 2010, he has been with the Institut National de la Recherche Scientifique (INRS), Université du Quebec, Montreal, QC, Canada, where he is currently an Associate Professor. He is the coauthor of the book *Radio Resource Management in Multi-Tier Cellular Wireless Networks* (Wiley, 2013) and *Radio Resource Management in Wireless Networks: An Engineering Approach* (Cambridge University Press, 2017). His current research interests include smart grids, cognitive radio, radio resource management, network control and optimization, and emerging enabling technologies for 5G wireless systems. He is currently a member of the Editorial Board of IEEE Transactions on Wireless Communications and IEEE Communications Surveys and Tutorials.

**KYUNGCHUN LEE** (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, in 2000, 2002, and 2007, respectively. From April 2007 to June 2008, he was a Postdoctoral Researcher with the University of Southampton, U.K. From July 2008 to August 2010, he was with Samsung Electronics, Suwon, South Korea. Since September 2010, he has been with the Seoul National University of Science and Technology, South Korea. In 2017, he was a Visiting Assistant Professor with North Carolina State University, Raleigh, NC, USA. His research interests include wireless communications and applied machine learning. He received the Best Paper Awards from the IEEE International Conference on Communications (ICC) and IEEE Wireless Communications and Networking Conference (WCNC), in 2009 and 2020.

**WON-JOO HWANG** (Senior Member, IEEE) received the B.S. and M.S. degrees in computer engineering from Pusan National University, Busan, South Korea, in 1998 and 2000, respectively, and the Ph.D. degree in information systems engineering from Osaka University, Osaka, Japan, in 2002. From 2012 to 2019, he was a Full Professor with Inje University, Gimhae, South Korea. Since January 2020, he has been a Full Professor with the School of Biomedical Convergence Engineering, Pusan National University, Yangsan, South Korea. His research interests include network optimization and cross layer design.

● ● ●