Improving Medium Access Efficiency With Intelligent Spectrum Learning

BO YANG\textsuperscript{1}, (Member, IEEE), XUELIN CAO\textsuperscript{2}, (Student Member, IEEE), OLUWASEYI OMOTERE\textsuperscript{1}, XIANGFANG LI\textsuperscript{1}, (Senior Member, IEEE), ZHU HAN\textsuperscript{2,3}, (Fellow, IEEE), AND LIJUN QIAN\textsuperscript{1}, (Senior Member, IEEE)

\textsuperscript{1}Department of Electrical and Computer Engineering and CREDIT Center, Prairie View A&M University, Texas A&M University System, Prairie View, TX 77446, USA
\textsuperscript{2}Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77004, USA
\textsuperscript{3}Department of Computer Science and Engineering, Kyung Hee University, Seoul 446-701, South Korea

Corresponding author: Lijun Qian (liqian@pvamu.edu)

This work was supported in part by the U.S. Office of the Under Secretary of Defense for Research and Engineering (OUSD(R&E)) under Agreement FA8750-15-2-0119, and in part by the U.S. Multidisciplinary University Research Initiative (MURI) under Grant MURI 18RT0073, Grant NSF EARS-1839818, Grant CNS1717454, Grant CNS-1731424, and Grant CNS-1702850.

\textbf{ABSTRACT} Through machine learning, this paper changes the fundamental assumption of the traditional medium access control (MAC) layer design. It obtains the capability of retrieving the information even the packets collide by training a deep neural network offline with the historical radio frequency (RF) traces and inferring the STAs involved collisions online in near-real-time. Specifically, we propose a MAC protocol based on intelligent spectrum learning for the future wireless local area networks (WLANs), called SL-MAC. In the proposed MAC, an access point (AP) is installed with a pre-trained convolutional neural network (CNN) model to identify the stations (STAs) involved in the collisions. In contrast to the conventional contention-based random medium access methods, e.g., IEEE 802.11 distributed coordination function (DCF), the proposed SL-MAC protocol seeks to schedule data transmissions from the STAs suffering from the collisions. To achieve this goal, we develop a two-step offline training algorithm enabling the AP to sense the spectrum with the aid of the CNN. In particular, on receiving the overlapped signal(s), the AP firstly predicts the number of STAs involving collisions and then further identifies the STAs’ ID. Furthermore, we analyze the upper bound of throughput gain brought by the CNN predictor and investigate the impact of the inference error on the achieved throughput. Extensive simulations show the superiority of the proposed SL-MAC and allow us to gain insights on the trade-off between performance gain and the inference accuracy.

\textbf{INDEX TERMS} Medium access control (MAC), spectrum sensing, deep learning, convolutional neural network (CNN).

I. INTRODUCTION

In the past few years, IEEE 802.11 based wireless local area networks (WLANs), commonly known as WiFi networks, have experienced a noticeable growth and keep pace with an ever-increasing number of mobile devices with low cost [1]. In such a case, how the mobile devices proficiently achieve channel coordination to improve the spectrum efficiency in the dense deployment WLANs scenarios (e.g., ultra-dense 5G networks [2]) are attracting significant devotion from both industry and academia [3]. Conventionally, the IEEE 802.11 distributed coordination function (DCF) uses a binary exponential backoff (BEB) scheme as a carrier sense multiple access with collision avoidance (CSMA/CA) mechanism to decrease collisions. However, this scheme severely degrades the network performance when there exists a large number of devices contending the channel [4].

Recently, the field of machine learning (ML), especially deep learning (DL), has emerged as a promising technique to enable intelligent spectrum sensing and management capabilities in the future wireless communications [5]. By training deep neural networks, the features of signals can be extracted, which goes beyond simple waveform classification, e.g., amplitude and phase. Furthermore, this newly achieved
popularity of DL for wireless communications is because of its potential driving the machine intelligence capability into some relevant applications of wireless communications. This in turn poses many challenges for the DL for WLAN’s physical (PHY) layer [6] (e.g., wireless channel modeling [7], [8], modulation scheme recognition [9]) as well as MAC layer design [10]. The spectrum at the unlicensed ISM frequency bands bears more and more traffic from the heterogeneous networks. Since some interests in wireless networks and machine learning has been quite visible in recent years, how to integrate deep learning with spectrum sensing to achieve the intelligent medium access on the unlicensed ISM band is still ongoing research. In this context, artificial intelligence (AI) functioning on the MAC will be a general trend, and how to improve channel access efficiency with the aid of DL achieving backward compatibility with conventional IEEE 802.11 DCF is the focus of this work. The above reasons motivate this work, and thus it is desirable to design a MAC protocol powered by DL to improve the network capacity for WLANs.

This paper aims to improve channel access efficiency of IEEE 802.11 DCF on the unlicensed ISM band. Instead of the traditional spectrum sensing methods that can only know whether the channel is occupied or not, we hope to find a way to explore more about the channel usage information, e.g., how many devices are sharing the spectrum and who are they. In this case, an additional deeper level of information is required to make the MAC protocol design more intelligently and efficiently. Therefore, by integrating MAC design with deep learning, better channel coordination and spectrum sharing can be achieved based on the inference results from the convolutional neural network (CNN) model. Compared with the conventional CSMA/CA based 802.11 DCF scheme, the unique advantage of the proposed solution is “Intelligent” yet “Undemanding”. Different from the conventional IEEE 802.11 DCF, the proposed SL-MAC protocol becomes more intelligent with the aid of deep learning and more efficient without requiring additional information. To the best of the authors’ knowledge, this is the first attempt to integrate MAC layer design with deep neural networks predictor in a conventional IEEE 802.11 DCF setup.

The main contributions of this paper are summarized as follows.

- Spectrum learning-based MAC framework. We introduce a comprehensive MAC framework integrating with SL to improve MAC efficiency and bandwidth utilization for the future WLANs. We formulate the users’ identification as a multi-class classification problem, which is resolved with the pre-trained CNN models.

- Design and implementation of the CNN structure. We first present the design of the CNN structure, including the master-CNN model and slave-CNN models. Then we detail the implementation of the CNN models, including data collection, two-step offline training and online inference. Based on the implementation of the pre-trained CNN models, the Access Point (AP) can identify the users from the overlapped signals and dynamically schedule the conflicted users’ data transmissions.

- Performance analysis of the proposed MAC framework. The upper bound of the throughput gain of the proposed SL-MAC protocol is analyzed, and the impact of the inference error on the achieved throughput is also investigated. Extensive simulations demonstrate the superiority of the proposed SL-MAC protocol.

The remainder of this paper is organized as follows. In Section II, we introduce related works. We present the proposed SL-MAC protocol in detail in Section III. In Section IV, the upper bound of the throughput gain brought by deep learning is first analyzed and then the impact of the inference error on the achieved throughput is investigated. The CNN framework including data collection, offline training, and online inference, is presented in Section V. In Section VI, simulations are conducted to demonstrate the superiority of the proposed SL-MAC protocol. Conclusions and future work are finally discussed in Section VII.

II. RELATED WORK

As aforementioned, ML techniques are playing a more and more critical part in the MAC layer design for WLANs. In general, ML can be roughly categorized as supervised learning and unsupervised learning. In this section, we elaborate on the role of the two categories of learning techniques in the MAC scheme design in wireless communication networks.

A. SUPERVISED LEARNING-BASED MAC DESIGN

In supervised learning, the learning agent learns from a labeled training dataset supervised. The objective is to find the mapping from the input feature space to the label so that reliable prediction can be achieved for the new input data. Due to the features of the supervised ML, the supervised learning-based MAC design is not suitable for the scenario where the device learns the environment without the help of a supervisor and the labeled training dataset. Specifically, a deep CNN model was used to perform the classification directly from spectrograms to identify PU behavioral patterns in a cognitive radio context [11]. Ruan et al. [12] proposed an ML-based predictive dynamic bandwidth allocation algorithm to address the uplink bandwidth contention and latency bottleneck of such networks. Rajendran et al. [13] achieved the automatic modulation classification based on the LSTM model, which learns from the time domain amplitude and phase information of the modulation schemes present in the training data for a distributed wireless spectrum sensing network. In [14], Liu et al. used a deep neural network (DNN) to explore the data-driven test statistic intelligently and proposed a covariance matrix-aware CNN-based spectrum sensing algorithm to improve the detection performance further. In [15], the machine learning algorithms were demonstrated that appreciably outperform classical signal detection methods in the 3.5-GHz band. Furthermore, Gao et al. [16]...
proposed a deep learning-based signal detector that exploits the underlying structural information of the modulated signals. Peng et al. [17] explored the transfer learning to address robustness in DL-based spectrum sensing to improve the robustness.

In particular, to achieve intelligent spectrum sensing, our previous works [18], [19], proposed a distributed MAC framework assisted by the deep learning, where a DNN model was trained offline to help to coordinate channel access by exploring the features from the overlapped signals. Kim et al. [20] proposed a deep learning-aided sparse code multiple access scheme. They used the autoencoder structure of a DNN to learn the codebook and decoding strategy for SCMA to minimize the bit error rate (BER). In [21], the medium access control protocol identification was investigated for applications in cognitive radio networks, where the second (CR) users detect the MAC schemes used by the primary users with the help of SVM. Thereby the CR users can be aware of the time and frequency of the spectrum hole.

B. UNSUPERVISED LEARNING-BASED MAC DESIGN

In unsupervised learning, the learner is provided only with unlabeled data, while learning is performed by finding an efficient representation of the data samples without any labeling information. As such, the unsupervised learning-based MAC design is suitable for the practical wireless network scenarios where no prior knowledge about the outcomes exists. Specifically, recent years have witnessed the wide study on deep reinforcement learning (DRL) in the field of dynamic spectrum access problems in wireless networks [22]–[26]. In particular, Nguyen et al. [22] proposed to use the deep Q-learning method to learn a state-action value function that determines an access policy from the observed states of all channels. In [23], the authors applied the DRL for SUs to learn appropriate spectrum access strategies in a distributed fashion, assuming no knowledge of the underlying system statistics. In [24], a multi-agent deep reinforcement learning method was adopted by the secondary users to learn sensing strategy from the sensing results of some of the selected spectra to avoid interference to the primary users and to coordinate with other secondary users in cognitive radio networks. Furthermore, Yu et al. [25] investigated a DRL-based MAC protocol to learn an optimal channel access strategy to achieve a certain pre-specified global objective for heterogeneous wireless networking. Cao et al. [26] proposed a DRL-based MAC protocol to assist the backscatter communications for Internet-of-Things (IoT) networks, where the DRL was introduced to learn the reserved information and make decisions accordingly.

Nevertheless, most of the ML-based MAC works listed in the literature above have introduced significant overhead for the existing wireless communication systems. They are not applicable for the conventional IEEE 802.11 WLANs networks, where the stations (STAs) are usually with low complexity, and it is difficult for them to “learn and predict” the dynamic changes in the networks. Our proposal MAC framework falls within the first category of ML approaches (i.e., supervised learning-based MAC), aiming to enable AP to figure out the STAs involved the collisions during the channel contention and then schedule them to re-transmit without collisions.

III. THE PROPOSED MAC FRAMEWORK

A. PRELIMINARY AND BASIC IDEA

This paper considers a typical WLAN scenario where there exists a total of \( N \) STAs which are associated with the AP and trying to transmit data packets to the AP. In the context of the typical application scenarios of IEEE 802.11 WLANs (e.g., the office building where the associated WiFi users belong to the company employees in general), it is reasonably assumed that the user base keeps stable within a future period in this paper. All the STAs contend to access the channel following the BEB scheme introduced by the conventional IEEE 802.11 DCF. Generally, the conventional IEEE 802.11 DCF scheme (e.g., four-way handshake) is used in WLANs to coordinate channel access among users. Once more than one user transmits the request-to-send (RTS) packets at the same time, the collisions occur and these users need to increase the backoff counter and contend to reaccess the channel. According to the analysis in [29], the collision probability is proportional to the number of STAs. In this case, the channel is usually with low utilization due to the severe collisions, especially in the dense deployment scenarios. In this paper, through machine learning, we change the fundamental assumption of the traditional MAC layer design. We obtain the capability of retrieving the information even the packets collide by training a CNN model offline with the historical RF traces and inferring the STAs involved collisions online in near-real-time.

Without loss of generality, we assume that each STA contends the channel using the traditional four-way handshake (i.e., RTS-CTS-DATA-ACK), and some of the STAs may choose the same backoff counter during their backoff procedures. As a result, the collisions usually occur at the AP side due to receiving multiple RTS packets at the same time, e.g., Alice and Bob in Fig. 1(a). In this paper, a novel MAC protocol, called spectrum learning-powered MAC (SL-MAC), is proposed for future dense WLANs, as highlighted by Fig. 1(b). In the proposed SL-MAC, suppose that a pre-trained CNN model is deployed at the AP in advance, which enables the AP to identify the STAs from the overlapped RTS signals. On receiving the overlapped signals, the AP can detect the number of users involved collisions and identify who they are with the aid of the pre-trained CNN model. This process can be considered as a multi-class classification problem. According to the inference results, the AP replies a CTS packet, which includes the scheduling information of the users’ data transmissions. The other users set their network allocation vector (NAV) accordingly and keep silent within this period. After the NAV expires, all the users can contend to reaccess the channel following the conventional IEEE 802.11 DCF.
B. PROPOSED MAC PROTOCOL

In the proposed SL-MAC protocol, the pre-trained CNN model files are implemented at the AP. On receiving the RTS signal(s), the AP not only can detect the number of STAs (denoted as \( n, n \in \mathcal{N} \), \( \mathcal{N} \) is the set of all the STAs) but also identify who are they. An example of the proposed protocol with three STAs contending to access the channel is illustrated in Fig. 2. The proposed SL-MAC protocol includes three operation steps, which are channel contention step, collision detection and identification step, and scheduling transmission step, as detailed in the following three subsections.

1) CHANNEL CONTENTION

In this step, all the STAs with data traffic accumulated in the MAC queue first contend to access the channel based on the BEB scheme. According to the conventional IEEE 802.11 DCF scheme, denote the minimal and the maximum contention window (CW) size as \( CW_{\text{min}} \) and \( CW_{\text{max}} \), respectively. In the beginning, all the STAs (e.g., the STA\(_A\), STA\(_B\), and STA\(_C\) in Fig. 2) first randomly chooses their backoff counters’ value as \( B_i \in [0, CW_{\text{min}}], i \in \mathcal{N} \), then the STAs start to perform backoff via the BEB scheme. Note that the backoff counter is hanged-up once the channel becomes busy. When the backoff counter becomes zero, i.e., \( B_i = 0 \), the STAs transmit the RTS packets to AP. For example, STA\(_A\) and STA\(_B\) finish their backoff at the same time to transmit the RTS packets simultaneously. In this case, the collision occurs at the AP. Otherwise, the RTS packet can be received successfully by the AP.

2) COLLISION DETECTION AND IDENTIFICATION

In this step, on receiving the RTS packet(s), the AP replies the CTS packet based on the inference results given by the pre-trained CNN model. Suppose that the pre-trained CNN model files have already been implemented at the AP. On receiving the RTS signal(s), the AP achieves the inference results by performing the feedforward calculation. Denote the time cost achieving the inference as \( \theta \), then the AP replies the CTS packet after the time \( t_{\text{SIFS}} + \theta \). \( t_{\text{SIFS}} \) denotes the duration of the short inter-frame space (SIFS), as illustrated in Fig. 2. To realize collision avoidance and maintain compatibility with legacy IEEE 802.11, a “Scheduling Info” field
including the conflicting STAs’ ID and their time instants to transmit data packets is introduced to the traditional IEEE 802.11 CTS packet, as illustrated in Fig. 3. Based on the inferred number the STAs involving collisions (denoted as \( \tilde{n} \)), the AP sets the “Duration” field of the CTS packet as \( \text{NAV}_{CTS} = t_{SIFS} + t_{TXOP} \) and broadcast the CTS packet to all the STAs, where \( t_{TXOP} = \tilde{n}(t_{DATA} + t_{SIFS} + t_{ACK}) \) denotes the scheduled period of the transmission opportunity (TXOP). On receiving the CTS packet, all the STAs decode the “Scheduling Info” field. If the STAs are not scheduled to transmit data packets, they set their network allocation vector (NAV) according to the “Duration” field of the CTS (i.e., \( \text{NAV}_{CTS} \)) and keep silent during this period. Otherwise, the STAs transmit data packets according to the scheduled time instants.

The reason why the number of STAs can be predictable and the STAs are identifiable is that the spectrum usage identification can be considered as a multi-class classification problem, which can be well solved by training a deep neural network [27], [28]. In this paper, the AP only needs to identify the number and ID of the STAs involving collisions. As a result, the scalability of the proposed CNN based MAC protocol mainly depends on the average number of STAs suffering collisions. It is known that in the practical WLAN scenarios, the data traffic of the STAs usually follows the Poisson distribution, which indicates that the MAC buffer of the STAs is not always full and the number of STAs transmitting at the same time is far less than \( N \). Therefore, high classification accuracy can still be achieved by the well-trained CNN model.

To better understand this, an intuitional example is presented with two STAs, denoted as \( S_1 \) and \( S_2 \). In this particular example, since we have a controlled environment by setting up various transmission scenarios, we can collect the RF traces, which includes all the combinations of STAs in the network, and we label the data using the ground truth. In this example, a dataset including 4 different coexisting transmission scenarios (i.e., ‘Idle channel’, ‘STA 1 waveform only’, ‘STA 2 waveform only’, and the ‘Combined waveforms’) are collected from the testbed. We consider that the RF traces collected in our USRP2 testbed consists of Inphase (I) and Quadrature (Q) signals in a matrix form, which includes sophisticated features of the wireless signals. Similar to the structure of images consisting of pixels in a matrix form, the deep neural networks (especially CNN) are generally the preferred methods extracting and learning the high-level information hidden in the RF traces. Based on this, the CNN classifier is trained offline with the historical RF traces collected from the four scenarios above until it is able to learn the features from the RF traces and make the reasonable inference. Therefore, this results in a 4-class classification problem, where the four classes are detailed as below.

- **Class-1:** ‘Idle’. None of the STAs is transmitting, i.e., the collected RF dataset is only from the noise floor measurements.
- **Class-2:** \( S_1 \). Only \( S_1 \) is transmitting, i.e., the collected RF dataset is from the noise floor measurements plus the signal from \( S_1 \).
- **Class-3:** \( S_2 \). Only \( S_2 \) is transmitting, i.e., the collected RF dataset is from the noise floor measurements plus the signal from \( S_2 \).
- **Class-4:** \( S_1 + S_2 \). Both of the two STAs are transmitting simultaneously, i.e., the collected RF dataset is from the noise floor measurements plus the overlapped signal from \( S_1 \) and \( S_2 \).

### III. PERFORMANCE ANALYSIS

The SL-MAC protocol design presented in Section III-B can ensure a high overall throughput by identifying the STAs involved collisions with the trained CNN model. However, the achieved throughput of SL-MAC tends to be degraded due to the inference errors introduced by the trained CNN model, especially when the number of STAs involving collision is large. In this section, we first analyze the upper bound of throughput gain, where the inference errors do not occur. After, we investigate the impact of inference errors on the achieved throughput of SL-MAC protocol.

#### A. UPPER BOUND OF THROUGHPUT GAIN

Compared to the conventional IEEE 802.11 DCF scheme, the throughput performance gain introduced by the deep learning is analyzed, where the inference error of the pre-trained CNN is assumed to be neglected. In this case, the analyzed gain becomes an upper bound.

According to Bianchi’s Markov model [29], the saturation throughput of the conventional IEEE 802.11 DCF is

\[
\phi_{DCF} = \frac{P_s P_{tr} E[P]}{(1 - P_{tr}) \sigma + P_{tr} P_s T_s + P_{tr} (1 - P_s) T_{DCF}} \geq \frac{P_s P_{tr} E[P]}{T_{DCF}},
\]

(1)
where $E[P]$ denotes the average size of the data packet payload, $\sigma$ denotes the duration of empty slot time, $P_r$ denotes the probability that there is at least one transmission in the considered slot time and $P_s$ represents the conditional probability that a transmission occurring on the channel is successful. Besides, $T_s = t_{\text{DIFS}} + t_{\text{RTS}} + t_{\text{CTS}} + t_{\text{DATA}} + t_{\text{ACK}} + 3t_{\text{SIFS}}$ is the average time the channel is sensed busy due to a successful transmission, and $T_{s}^{\text{DCF}} = t_{\text{DIFS}} + t_{\text{RTS}}$ is the average time the channel is sensed busy by each device during a collision following the IEEE 802.11 DCF scheme.

Even though the collision occurs in our proposed SL-MAC protocol, the channel can be still utilized by scheduling the STAs’ data transmissions within a TXOP, as shown in Fig. 2. Therefore, the saturation throughput of SL-MAC (abbreviated as $\phi_{\text{DM}}$) can be obtained as

$$\phi_{\text{DM}} = \frac{P_r P_{tr} E[P] + n P_{tr} (1 - P_s) E[P]}{(1 - P_{tr}) \sigma + P_{tr} P_s T_s + P_{tr} (1 - P_s) T_{c}^{\text{DM}}},$$

(2)

where $n$ denotes the average number of STAs involved in RTS collisions. $T_{c}^{\text{DM}}$ denotes the average time the channel is utilized by scheduling STAs’ transmissions when a collision is detected for the proposed SL-MAC protocol.

According to Fig. 2, $T_{c}^{DM}$ can be calculated as

$$T_{c}^{DM} = t_{\text{DIFS}} + t_{\text{RTS}} + t_{\text{CTS}} + 3t_{\text{SIFS}} + \theta + t_{\text{TXOP}}.$$  

(3)

where $\theta$ denotes the inference delay with the pre-trained CNN. $t_{\text{TXOP}}$ is the time period of the TXOP including scheduled $n$ data transmissions, i.e., $t_{\text{TXOP}} = n(t_{\text{DATA}} + t_{\text{SIFS}} + t_{\text{ACK}})$.

Substituting $T_{c}^{\text{DCF}} = t_{\text{DIFS}} + t_{\text{RTS}}$ into (3), we have

$$T_{c}^{DM} = T_{c}^{\text{DCF}} + \omega,$$

(4)

where $\omega = t_{\text{CTS}} + 2t_{\text{SIFS}} + \theta + t_{\text{TXOP}}$.

As a result, we can obtain

$$T_{DM} = T_{c}^{\text{DCF}} + P_{tr} (1 - P_s) \omega.$$  

(5)

According to (1), (2) and (5), the upper bound gain brought by SL-MAC can be calculated as

$$\tilde{\eta} = \frac{\phi_{\text{DM}}}{\phi_{\text{DCF}}} = \frac{n (1 - P_s) + P_s}{P_s} \frac{T_{\text{DCF}}}{T_{c}^{\text{DCF}} + P_{tr} (1 - P_s) \omega}.$$  

(6)

Remark 1: Suppose that the inference error of the pre-trained CNN is not considered, it can be seen from (6) that the gain brought by the pre-trained CNN is proportional to the number of STAs involved in the RTS collisions (i.e., $n$). In practice, the inference error of the multi-class classification problem increases with an increase of $n$. Therefore, there exists a trade-off between performance gain ($\eta$) brought by deep learning and the inference accuracy.

**B. ACHIEVED THROUGHPUT WITH INFERENCE ERRORS**

To characterize the impact of the inference error on the achieved throughput of SL-MAC, the definition of the inference error rate is introduced as follows.

**Definition 1 (Inference Error Rate):** Inference error rate introduced by trained CNN model is defined as the ratio of the number of correct inference to the total number of inference, i.e., $\gamma = \frac{N_{\text{correct}}}{N_{\text{total}}}$, and $\gamma \in [0,1]$.

Based on Definition 1, the throughput of SL-MAC is calculated as

$$\phi_{\text{DM}} = (1 - \gamma) \phi_{\text{DM}}^{w/e} + \gamma \phi_{\text{DM}}^{e},$$

(7)

where $\phi_{\text{DM}}^{w/e}$ denotes the throughput of SL-MAC without inference error, i.e., $\phi_{\text{DM}}^{w/e} = P_{tr} (n(1-P_s)+P_s) E[P]$. $\phi_{\text{DM}}^{e}$ denotes the throughput of SL-MAC with inference error rate $\gamma$.

To calculate $\phi_{\text{DM}}^{e}$, two typical cases are considered: 1) over-estimation, and 2) under-estimation. Specifically, if over-estimation happens, the inference results include not only the STAs involved collisions but also the other STAs not being collided. Denote the number of users inferred by the CNN as $\tilde{n}$, then $\tilde{n} > n$ holds and $\tilde{n}$ transmissions are scheduled, which leads to the channel resources waste in this case. If under-estimation occurs, the inference results may miss one or more users involved collisions, i.e., $\tilde{n} < n$. As a result, just a portion of the users involving collisions can be scheduled to transmit, which, however, degrades the fairness of each device.

Denote the achieved throughput in two cases as $\psi_{\text{over}}$ and $\psi_{\text{under}}$, respectively. Then we have

$$\phi_{\text{DM}}^{e} = \alpha \psi_{\text{over}}^{DM} + (1 - \alpha) \psi_{\text{under}}^{DM},$$

(8)

where $\alpha \in [0,1]$ is the coefficient representing the probability that over-estimation occurs. Similarly, $1 - \alpha$ is the probability that under-estimation occurs. In the following, we analyze the achieved throughput under the two cases.

1) CASE 1: OVER-ESTIMATION

In this case, the number of data transmission opportunities scheduled is larger than the true value, i.e., $\tilde{n}_{\text{over}} > n$. $\tilde{n}_{\text{over}}$ is the number of users suffering collisions inferred by the CNN with over-estimation. According to (2), the achieved throughput in case 1 can be obtained as

$$\phi_{\text{over}}^{DM} = \frac{P_{tr} P_{r} E[P] + n P_{tr} (1 - P_{s}) E[P]}{(1 - P_{tr}) \sigma + P_{tr} P_{s} T_{s} + P_{tr} (1 - P_{s}) T_{c}^{\text{over}}}.\frac{T_{\text{DCF}}}{T_{c}^{\text{over}} - \omega},$$

(9)

where $T_{c}^{\text{over}}$ denotes the average length of a slot time when over-estimation happens, $T_{c}^{\text{DM}}$ is the average time the channel is utilized by scheduling STAs’ transmissions when a collision is detected for the proposed SL-MAC protocol under the case 1.

In this case, $T_{c}^{\text{DM}}$ is calculated as

$$T_{c}^{\text{DM}} = t_{\text{DIFS}} + t_{\text{RTS}} + t_{\text{CTS}} + 2t_{\text{SIFS}} + \theta + t_{\text{TXOP}}.$$  

(10)
where $\nu_{\text{over}}$ is the time period of the TXOP including scheduled data transmissions, i.e., $\nu_{\text{over}} = \tilde{n}_{\text{over}}(t_{\text{DATA}} + t_{\text{SIFS}} + t_{\text{ACK}})$.

**Proposition 1:** Compared to the perfect inference case where the inference errors do not consider, the performance loss introduced by over-estimation can be calculated as

$$\theta_{\text{over}} = \frac{\nu_{\text{over}} - \nu_{\text{DM}}}{\nu_{\text{DM}}} = \frac{\nu_{\text{DM}} - \nu_{\text{DM}}}{\nu_{\text{DM}}}.$$ 

Therefore, the over-estimation deteriorates the system throughput of the SL-MAC protocol. On the other hand, when under-estimation occurs, we can infer from (14) that the impact of inference errors on system throughput performance of SL-MAC is not decisive, i.e., the total throughput may decrease or remain the same. However, since only a portion of STAs suffering collisions can be scheduled to transmit data within the TXOP, the remaining STAs not being scheduled will double the contention window size following the traditional CSMA/CA scheme. In such a case, it becomes more difficult for them to access the channel to lead to fairness problem.

Substituting (9) and (12) into (8), the achieved throughput with inference errors is calculated as (15), as shown at the bottom of this page. Then, by substituting (15) into (7), the system throughput can be achieved as (16), as shown at the bottom of this page, where $\gamma$ denotes the inference error rate and $\alpha$ is the probability that over-estimation occurs.

**V. CNN FRAMEWORK DESIGN**

**A. OVERVIEW OF THE CNN ARCHITECTURE**

To identify the collisions in the proposed MAC protocol, a CNN framework is proposed to predict the number and ID of the STAs involving collisions by offline training on a large set of labeled dataset. As illustrated in Fig. 4, the CNN framework includes a master-CNN model inferring the number of STAs and $N - 2$ slave-CNN models identifying the ID of the STAs accordingly. Specifically, in the trained CNN models, considering that CNN is typically suitable for processing grid-like data, e.g., 1-D grid time-series data and 2-D grid image data [30], so the collected RF traces can be fed into the convolutional layer with data preprocessing. The collected I and Q samples are reshaped into 4-dimensional tensor suitable for Keras convolutional layer.

The proposed CNN structure is illustrated in Fig. 4, where each Convolutional layer is followed by a Rectified Linear Units layer (denoted as “Conv + ReLU”). For the feature extraction, the pattern of the master-CNN and slave-CNN models is based on $[\text{Conv} + \text{ReLU}] \times K^1_M$ and $[\text{Conv} + \text{ReLU}] \times K^1_M$, respectively. After this, total of $K^2_M$ and $K^2_S$ fully-connected (FC) layers are used for the master-CNN model and slave-CNN model to process the flattened matrix and then classify the signals using Softmax activation.

1It is worth noting that several methods have been proposed to scale up deep neural network training across graphics processing unit (GPU) clusters [33], which helps to reduce the runtime of the offline training.
on the output layer. The master-CNN model infers the number of STAs involving collisions (denote as $\tilde{n}$), and the output is a $(N - 2) \times 1$ vector indicating the probability of each class. The slave-CNN model infers the ID of the STAs involving collisions, and the output is a $C_{\tilde{n}} \times 1$ vector indicating the probability of each class. Finally, Adam optimizer [31] is used to optimize the CNN models, and the CNN based predictor is trained offline until it can learn the features from the RF traces and make the reasonable inference from the overlapped signals. The CNN framework in the proposed MAC protocol consists of three aspects: data collection, offline training, and online inference, as illustrated in Fig. 5. It worth noting that the data collection and offline training are performed only once. After the CNN models are trained offline, we can achieve inference online once given an I/Q dataset.

**B. DATA COLLECTION**

During the data collection, we collect the RF traces with a constant SNR using our USRP2 testbed, which is wired connected to a host PC (e.g., a laptop) with an implementation of the GNU Radio [32], as shown in Fig. 6(a). The data collected with a constant SNR is valid because the closed-loop power control is generally used to obtain a constant received power at the AP. Specifically, for the device side, the laptop is mainly responsible for baseband processing while the universal software radio peripheral (USRP2) focuses on the up-conversion, digital-to-analog (D/A), and transmitting from the wireless radio. For the AP side, the USRP2 module first receives signals from the radio, then performs A/D and down-conversion. After that, the laptop receives the signals from the USRP2 via Ethernet and carries on the baseband processing. Finally, the I/Q sequences can be stored as a file at the laptop. The collected I and Q samples in size of $N_{\text{examples}} \times \text{Dimension}$ are reshaped into 4-dimensional tensor suitable for Keras convolutional layer. This takes the form $N_{\text{examples}} \times N_{\text{channels}} \times \text{Dimension}_1 \times \text{Dimension}_2$. We have $N_{\text{examples}}$ examples from I and Q samples after reshape, each consisting of $w$ ($w$ is the window size) time-series samples. Besides, $N_{\text{channels}} = 1$ similar to RGB values in imagery, $\text{Dimension}_1 = 2$ holding for our I and Q channels, and $\text{Dimension}_2 = w$. In this paper, the window size ($w$) is set to be 32, 128, and 512, respectively. With the collected data traces (i.e., I and Q samples), we train on 80% of collected RF data set (training set). It contains about 790 million I and Q samples, validate on 10% of data set (validation set) and test on 10% of data set (testing set) each corresponding to about 100 million of the I and Q samples.

**C. OFFLINE TRAINING AND ONLINE INFERENCE**

The CNN based predictor is offline trained based on the historical radio frequency (RF) traces. Then the pre-trained CNN predictor can be deployed at the AP and used to indicate the STAs from the overlapped signals, as highlighted in Fig. 5. Specifically, we first collect the RF traces with our USRP2 testbed within a relatively large window size. Suppose that the window size is $w$, the dimension of each
RF sample is 2 (i.e., In-phase (I) and Quadrature (Q) signals), then the dimensionality of the input space equals to $2w$. After, the CNN predictor is trained and tested offline based on the collected historical RF traces. During the offline training, back-propagation is performed to train the CNN model. It is worth noting that we use the Graphics Processing Unit (GPU) cluster (i.e., NVIDIA DGX-1) with TensorFlow installed to accelerate the offline training process [33], which only needs to be executed once, as highlighted in Fig. 6(b). Considering that the ground truth is known to us, the identification of STAs can be considered as a multi-class classification problem which belongs to the supervised learning. After the offline training, the online RF traces can be fed into the pre-trained CNN predictor deployed at the AP to identify the STAs from the overlapped signals in near-real-time.

Specifically, for the multi-class classification problem, the probability of each class is predicted using the Softmax function. The predicted probability for the $i$-th class is given as

$$\sigma_i(z) = \frac{e^{z_i}}{\sum_{j=1}^{M} e^{z_j}}, \quad i = 1, \ldots, M,$$

where $M$ is the total number of classes, $z$ is the output of the last fully connected layer.

Then, we conventionally set the loss function of the classification (denoted as $L$) as cross-entropy [34], which is given as

$$L = -\frac{1}{M} \sum_{i=1}^{M} y_i \ln(f(x_i)),$$

where $x_i$ is the $i$-th input data sample, $y_i$ denotes the corresponding groundtruth and $f(x_i)$ is the actual output of neurons.

The two-step offline training and online inference are illustrated with an example with a total of four STAs existing in the network, as shown in Fig. 7. During the first-step training, the master-CNN model detecting the total number of STAs involved collisions is trained offline based on the whole RF traces, and the inference result falls into one of the three classes. Note that the class-3 has a fixed number of STAs (i.e., all the STAs in class-3). Therefore, we only need to indicate the STAs’ ID in the remaining two classes, i.e., class-1 and class-2. In the second-step training, three slave-CNN models are trained separately offline based on different RF traces. Compared to the conventional CNN training that includes 16 classes, the accuracy of the proposed two-step CNN training can be improved due to the decrease in the number of classes.

D. IMPLEMENTATIONS

The implementation of the proposed CNN predictor that includes one master-CNN model file and two slave-CNN model files is illustrated in Fig. 8. In the CNN predictor, the AP first infers the number of STAs involved collisions according to the new RF traces via the pre-trained master-CNN model. After this, based on the previous reference result (i.e., the number of STAs), the AP selects a slave-CNN model accordingly and then performs the inference to identify the STAs’ ID. For the implementation of the proposed CNN-based MAC, the complexity and generalizability are discussed as below.
PAC learning, generalization can be achieved by designing models even for parameter settings not included in the training. The trained CNN will enable the generalization property of machine learning models to avoid over-fitting by dividing the dataset into training, validation, and testing data. Therefore, the proposed SL-MAC with pre-trained CNN model can generalize well to a new environment or at different positions because of the nonstationary nature of the wireless environment has been learned well.

**TABLE 1. Inference accuracy and inference time cost of the CNN model, where \( N = 4 \).**

| Model                  | Inference Accuracy |
|------------------------|--------------------|
| Master-CNN model       | 90%                |
| Slave-CNN model-1      | 92%                |
| Slave-CNN model-2      | 95%                |

**VI. NUMERICAL RESULTS**

**A. TESTING RESULTS OF CNN**

The hyper-parameters of the pre-trained CNN model files are summarized as follows. For the master-CNN model detecting the number of STAs and the slave-CNN model-1, they contain three convolution layers and one fully connected (FC) layer, i.e., \( K_M = K_S = 3 \), \( K_M^2 = K_S^2 = 1 \). For the slave-CNN model-2, it contains four convolution layers and one fully connected (FC) layer, i.e., \( K_M^2 = 4 \), \( K_S^2 = 1 \). The average inference accuracy is presented in Table 1, where four STAs are taken as an example, and the window size is set to be \( w = 128 \). It can be seen from Table 1 that all of the pre-trained CNN models have achieved a relatively high inference accuracy (i.e., \( \geq 90\% \)). Therefore, the inference error of the pre-trained CNN is reasonably neglected in the following simulations.

**B. PERFORMANCE EVALUATION**

**1) SIMULATION SETTINGS**

The simulations are carried out using the ns-2 simulator [36] according to the PHY and MAC layer parameters that are presented in Table 2. To integrate with the experimental results of deep learning and the ns-2 network simulations, we consider the inference error rate \( \eta \) as an input to the ns-2 simulations. Specifically, we collect the RF data using our USRP2 testbeds and perform the testing via the pre-trained deep learning model. Since the deep learning experiments have been done in TensorFlow, we use a Log-Sigmoid function \( \eta = \frac{D}{1 + e^{\eta + CN}} \) to portray the relationship between \( \eta \) and \( N \) according to the inference results given by the CNN [19]. Therefore, if the total number of devices \( N \) is given, \( \eta \) can be obtained by mapping accordingly and then fed into the ns-2 simulations. We consider a general star topology...
TABLE 2. Simulation parameters.

| Parameter     | Value  | Parameter     | Value  |
|---------------|--------|---------------|--------|
| RTS size      | 20 bytes | CTS size     | 14 bytes |
|               |        | Ack size      | 14 bytes |
| Data packet rate | 11 Mbps | SIFS         | 10 µs  |
| Control packet rate | 2 Mbps | DIFS         | 50 µs  |

FIGURE 9. Normalized throughput versus number of devices (N), where data payload size is 32 bytes in (a) and 1024 bytes in (b).

FIGURE 10. Normalized throughput versus data payload size (E[P]).

network scenario with a totally N STAs uniformly distributed within the AP’s converge radius of 50 m. We assume that the inference time cost is ignored (i.e., \( \theta = 0 \)), and each STA is under the saturation traffic with the same data payload size. To demonstrate the superiority of the MAC efficiency improved by the deep learning, we implemented and compared the performance of SL-MAC with the conventional IEEE 802.11 DCF protocol using a four-way handshake, i.e., RTS-CTS-DATA-ACK. The simulation results are averaged from 100 runs.

2) PERFORMANCE EVALUATION

In Figs. 9-10, we compare the normalized throughput of the proposed SL-MAC protocol with IEEE 802.11DCF to evaluate the upper bound of the performance gain brought by deep learning, where the inference errors of SL-MAC are ignored. Fig. 9 presents the normalized throughput against the number of devices (N). First, it is observed from Fig. 9 that the analysis of SL-MAC matches well with the simulation. The throughput of the proposed SL-MAC protocol increases and then keeps stable as N increases while the throughput of the IEEE 802.11 DCF severely decreases. This is due to the severe collisions occur using CSMA/CA, but the collisions can be avoided with the proposed SL-MAC based on deep learning. Suppose that the inference error is neglected, it can be seen from Fig. 9 that a larger N can lead to more significant normalized throughput of SL-MAC. This demonstrates that the proposed SL-MAC protocol is suitable for dense WLAN scenarios.

Moreover, it is observed from Fig. 9 that when the data payload size is relatively large (e.g., 1024 bytes in Fig. 9(b)), RTS/CTS handshake scheme can help to reduce the time spent during a collision, with respect to the basic access mechanism. However, when the data payload size is relatively small (e.g., 32 bytes in Fig. 9(a)), RTS/CTS handshake degrades throughput performance due to the control overhead. Therefore, Fig. 9 demonstrates that the normalized throughput gain brought by deep learning is decreased when the data payload size increases. This phenomenon is also verified by Fig. 10. The main reason is that when the data payload size is relatively large, the time ratio occupied by the control handshake is reduced, and thus the total time with RTS collisions resolved by the deep learning is decreased. Therefore, the advantage of SL-MAC declines for the larger data payload size. Besides, given a data payload size, we find that the more devices existing in the network, the more improvement can be achieved by the proposed SL-MAC, which verifies the Remark 1. This is because when the number of devices increases, more collisions will occur, which is exactly what the proposed SL-MAC expected. Only when the collision occurs, the AP can get the opportunity to schedule the STAs involved in collisions to transmit their data packets within the TXOP.
TABLE 3. RTS collision probability under different user density.

| Number of STAs (N) | User density (ρ) | RTS collision |
|-------------------|------------------|---------------|
| 1                 | 0.13×10^{-2}     | 0             |
| 5                 | 0.64×10^{-2}     | 28%           |
| 10                | 1.27×10^{-2}     | 40%           |
| 15                | 1.90×10^{-2}     | 45%           |
| 20                | 2.55×10^{-2}     | 49%           |
| 25                | 3.18×10^{-2}     | 52%           |
| 30                | 3.82×10^{-2}     | 54%           |
| 35                | 4.45×10^{-2}     | 56%           |
| 40                | 5.10×10^{-2}     | 58%           |
| 45                | 5.73×10^{-2}     | 59%           |
| 50                | 6.36×10^{-2}     | 60%           |
| 80                | 0.010            | 66%           |
| 100               | 0.013            | 68%           |
| 150               | 0.019            | 73%           |
| 200               | 0.025            | 77%           |
| 250               | 0.032            | 79%           |
| 300               | 0.038            | 82%           |

To better understand the benefit of SL-MAC even in the absence of inference error, we define the user density as the ratio between the total number of STAs and the AP's coverage area, i.e., $ρ = \frac{N}{πχ}$, where $χ = πr^2$ and $r$ denotes the radius of the AP coverage area. $ρ$ denotes the number of STAs deployed per $m^2$. Taking a radius of 50 m as an example, we evaluate the probability of RTS collision, as shown in Table 3. It is observed that the RTS collision probability increases significantly with the increase of user density. In particular, when 25 STAs are deployed, i.e., the user density is only $3.18×10^{-3}$, the RTS collision probability has exceeded 50%. Furthermore, if scenarios with dense deployment are considered, e.g., stadium, train station, or conference room, the user density may usually reach up to 1 STA per $m^2$, i.e., at least one STA is deployed with $1 m^2$ [37]. In this case, the RTS collision probability reached almost 100%, which leads to the failure of the traditional 802.11 DCF scheme. Therefore, the proposed SL-MAC can explore promising advantages of deep learning in wireless networks, especially with dense deployment scenarios.

Figs. 11-12 evaluate the impact of the inference error rate ($γ$) of a trained CNN model on the normalized throughput of the proposed SL-MAC protocol, where the over-estimation probability ($α$) is set to be 0.5 and 0.1, respectively. It is observed from Fig. 11 and Fig. 12 that the normalized throughput of SL-MAC protocol decreases with an increase of $γ$. This is because that when $γ$ becomes larger, more inference errors occur, which degrade the achieved throughput. Moreover, we can observe that the decline of normalized throughput with $α = 0.5$ is more significant than that of $α = 0.1$. The reason for this outcome is apparent since the larger value of $α$ can cause a higher probability of over-estimation, which leads to throughput degradation, as presented in Remark 2.

Figs. 13-14 evaluate the impact of the over-estimation probability ($α$) on the normalized throughput of the proposed SL-MAC protocol, where the inference error rate ($γ$) is set to be 0.5 and 0.1, respectively. It can be observed that when $γ$ is given, the normalized throughput of SL-MAC protocol decreases with an increase of $α$ because of the higher probability of over-estimation. Furthermore, compared to Fig. 13
with $\gamma = 0.5$, it is observed from Fig. 14 that the normalized throughput almost keeps unchanged as $\alpha$ increases when $\gamma$ is set as 0.1. This is because that a larger value of $\gamma$ leads to a larger inference error rate. In such a case, as $\alpha$ increases, more over-estimations occur when $\gamma = 0.5$, which can seriously degrade the throughput performance of SL-MAC. Moreover, we can see from Figs. 11-14 that with the increase of data payload size, the throughput gain (i.e., the normalized throughput gap) decreases because that the inference errors can introduce more damage to the SL-MAC with larger payload size.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a novel MAC protocol for future WLANs. Because of the severe collisions that occur using the traditional CSMA/CA scheme, spectrum learning-powered MAC protocol (SL-MAC) is proposed to schedule the STAs’ data transmissions who involved in the RTS collisions. An essential feature of the proposed SL-MAC protocol is the backward compatibility with the conventional IEEE 802.11 DCF mechanism. Then, both of the superiority and inferiority brought by the CNN predictor are analyzed, which demonstrates the necessity of the potential applications of deep learning to the MAC design. Extensive simulations demonstrate the advantages of the proposed SL-MAC protocol. This paper is the first attempt to integrate fundamental MAC layer design with deep neural networks in a conventional IEEE 802.11 DCF setup to avoid introducing additional hardware overhead and aim to lay a foundation for further related research.

Our potential future works are listed as follows. How to generate new RF training datasets in practical scenarios and retrain the CNN models more efficiently is regarded as one of our future work, known as the “scalability” issue. For example, when new STAs join to the WLAN, this impacts on the inferring results of the slave-CNN models (i.e., the ID of the STAs involving collisions). It would be difficult to collect enough labeled RF datasets to retrain the CNN models since there exist too many combinations in real environments.

To mitigate this issue, we can take advantage of the historical labeled data and combine it with some unlabeled data. In this context, to achieve high inference accuracy of the CNN models, fine-tuning the slave-CNN models with the collected combined RF dataset (some of the RF datasets is unlabeled) becomes promising. Since semi-supervised learning (SSL) makes use of unlabeled RF traces to facilitate the learning process and transfer learning (TL) can learn generalizable representations in source domain [38], the SSL and TL could be combined in our future work to yield significant inference improvements on the spectrum learning.

ACKNOWLEDGMENT

The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Office of the Under Secretary of Defense for Research and Engineering (OUSD(R&E)) or the U.S. Government.

REFERENCES

[1] C. V. N. Index, “Cisco visual networking index: Global mobile data traffic forecast update,” Cisco Syst., San Jose, CA, USA, White Paper 2016-2021, Mar. 2017.
[2] M. Carmo, S. Jardim, A. Neto, R. Aguiar, D. Corujo, and J. J. P. C. Rodrigues, “Slicing WiFi WLAN-sharing access infrastructures to enhance ultra-dense 5G networking,” in Proc. IEEE Int. Conf. Commun. (ICC), May 2018, pp. 1–6.
[3] R. Ali, N. Shahin, Y. B. Zikria, B. S. Kim, and S. W. Kim, “Deep reinforcement learning paradigm for performance optimization of channel observation-based MAC protocols in dense WLANs,” IEEE Access, vol. 7, pp. 3500–3511, 2019.
[4] X. Cai, B. Li, M. Yang, Z. Yan, B. Yang, Y. Jin, and X. Li, “Fractional backoff algorithm for the next generation WLAN,” in Proc. 11th EAI Int. Wireless Internet Conf. (WICON), Taipei, Taiwan: Springer, Oct. 2018, pp. 24–31.
[5] C. Jiang, H. Zhang, Y. Ren, Z. Han, K.-C. Chen, and L. Hanzo, “Machine learning paradigms for next-generation wireless networks,” IEEE Wireless Commun., vol. 24, no. 2, pp. 98–105, Apr. 2017.
[6] 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.
[7] J. Yuan, H. Q. Ngo, and M. Mathaiuvi, “Machine learning-based channel prediction in massive MIMO with channel aging,” IEEE Trans. Wireless Commun., vol. 19, no. 5, pp. 2960–2973, May 2020.
[8] 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.
[9] G. J. Mendis, J. Wei, and M. Madanayake, “Deep learning-based automated modulation classification for cognitive radio,” in Proc. IEEE Int. Conf. Commun. Syst. (ICCS), Dec. 2016, pp. 1–6.
[10] H. B. Pasandi and T. Nadeem, “Challenges and limitations in automating the design of MAC protocols using machine-learning,” in Proc. Int. Conf. Artif. Intell. Inf. Commun. (ICAIIC), Feb. 2019, pp. 107–112.
[11] F. Wunsch, F. Puisana, S. Rajendran, A. Selim, P. Alvarez, S. Müller, S. Kosloski, B. Van den Bergh, and S. Pollin, “DrySPAN spectrum challenge: Situational awareness and opportunistic spectrum access benchmarked,” IEEE Trans. Cognit. Commun. Netw., vol. 3, no. 3, pp. 550–562, Sep. 2017.

2Note that when new STAs join to the WLAN, the combined RF traces can easily be collected. However, the ground truth of all the combinations is still challenging to achieve.
[12] L. Ruan, M. P. I. Dias, and E. Wong, “Machine learning-based bandwidth prediction for low-latency H2M applications,” IEEE Internet Things J., vol. 6, no. 2, pp. 3743–3752, Apr. 2019.

[13] S. Rajendran, W. Meert, D. Giustinianno, V. Lenders, and S. Pollin, “Deep learning models for wireless signal classification with distributed low-cost spectrum sensors,” IEEE Trans. Cognit. Commun. Netw., vol. 4, no. 3, pp. 433–445, Sep. 2018.

[14] C. Liu, J. Wang, X. Liu, and Y.-C. Liang, “Deep CM-CNN for spectrum sensing in cognitive radio,” IEEE J. Sel. Areas Commun., vol. 37, no. 10, pp. 2306–2321, Oct. 2019.

[15] W. M. Lees, A. Wunderlich, P. J. Jeavons, P. D. Hale, and M. R. Souryal, “The Network Simulator NS-2,” Berlin, Germany: Eur. Conf. Mach. Learn. Knowl. Discovery Databases Joint, 2016, pp. 1–7.

[16] Y. Zhang, P. Cai, C. Pan, and S. Zhang, “Multi-agent deep reinforcement learning-based cooperative spectrum sensing with upper confidence bound exploration,” IEEE Access, vol. 7, pp. 118898–118906, 2019.

[17] Y. Yu, T. Wang, and S. C. Liew, “Deep-reinforcement learning multiple access for heterogeneous wireless networks,” IEEE J. Sel. Areas Commun., vol. 37, no. 6, pp. 1277–1290, Jun. 2019.

[18] X. Cao, Z. Song, B. Yang, X. Du, L. Qian, and Z. Han, “Deep reinforcement learning MAC for backscatter communications relying on Wi-Fi architecture,” in Proc. IEEE Global Commun. Conf. (GLOBECOM), Dec. 2019, pp. 1–6.

[19] O. Omotore, L. Qian, R. Jantti, M. Pan, and Z. Han, “Big RF data assisted cognitive radio network coexistence in 3.5GHz band,” in Proc. 50th Int. Conf. Comput. Commun. Netw. (ICC-CN), Jul. 2017, pp. 1–8.

[20] O. Omotore, J. Fuller, L. Qian, and Z. Han, “Spectrum occupancy prediction in coexisting wireless systems using deep learning,” in Proc. IEEE 88th Veh. Technol. Conf. (VTC-Fall), Aug. 2018, pp. 1–7.

[21] G. Bianchi, “Performance analysis of the IEEE 802.11 distributed coordination function,” IEEE J. Sel. Areas Commun., vol. 18, no. 3, pp. 535–547, Mar. 2000.

[22] D. P. Kingma and J. B. Adam, “A method for stochastic optimization,” in Proc. Int. Conf. Learn. Represent. (ICLR), San Diego, CA, USA, May 2015.

[23] Eric Blossom. GNU Radio. Accessed: 2020. [Online]. Available: http://www.gnuradio.org/software/gnuradio/.

[24] F. N. Iandola, M. W. Moskewicz, K. Ashraf, and K. Krutzter, “FireCaffe: Near-linear acceleration of deep neural network training on compute clusters,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 2592–2600.

[25] J. Nam, J. Kim, E. L. Mencia, I. Gureycz, and J. Furrkranz, “Large-scale multi-label text classification revisiting neural networks,” in Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases, Berlin, Germany: Springer, Sep. 2014, pp. 437–452.

[26] The Network Simulator NS-2. Accessed: 2011. [Online]. Available: http://www.isi.edu/nsnam/ns

[27] Y. Lin, B. Li, M. Yang, Q. Qu, Z. Yan, X. Zuo, and B. Yang, “Integrated link-system level simulation platform for the next generation WLAN—IEEE 802.11ax,” in Proc. IEEE Global Commun. Conf. (GLOBECOM), Dec. 2016, pp. 1–7.

[28] H.-Y. Zhou, A. Oliver, J. Wu, and Y. Zheng, “When semi-supervised learning meets transfer learning: Training strategies, models and datasets,” 2018, arXiv:1812.05313. [Online]. Available: http://arxiv.org/abs/1812.05313

BO YANG (Member, IEEE) received the B.E., M.S., and Ph.D. degrees from Northwestern Polytechnical University, Xi’an, China, in 2007, 2010, and 2016, respectively. From October 2017 to January 2020, he was a Postdoctoral Fellow with the Department of Electrical and Computer Engineering, Prairie View A&M University (PVAMU), and a member of Texas A&M University System, Prairie View, TX, USA. He is currently a Postdoctoral Fellow with the Engineering Product and Design Pillar, Singapore University of Technology and Design (SUTD), Singapore. He has authored or coauthored over 20 articles published and presented in prestigious journals and conferences. He holds over ten granted and pending international and Chinese patents. His research interests include machine learning, 5G/6G, and application of artificial intelligence for wireless networks. He was the Chair of the Session entitled Machine Learning in the IoT, M2M, Sensor Networks, and Ad-Hoc Networking on IEEE VTC-Fall 2018.

XUELIN CAO (Student Member, IEEE) received the M.S. and Ph.D. degrees from Northwestern Polytechnical University, Xi’an, China, in 2010 and 2019, respectively. From 2010 to 2012, she worked as an Engineer with the CDMA BTS Laboratory, Nokia Siemens Networks (NSN), Hangzhou, China. From September 2017 to January 2020, she was a Visiting Scholar with the Department of ECE, University of Houston, Houston, TX, USA. She is currently a Postdoctoral Fellow with the Engineering Product and Design Pillar, Singapore University of Technology and Design (SUTD), Singapore. Her research interests include medium access control protocol design and modeling, 5G/6G heterogeneous networks, reconfigurable intelligent surface (RIS), and backscatter communications.

OLUWASEYI OMOTERE is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering and CREDIT Center, Prairie View A&M University, a member of the Texas A&M University System, Prairie View, TX, USA. His research interests are in big data analytics, wireless communications, and machine learning.
XIANGFANG (LINDSEY) LI (Senior Member, IEEE) received the M.S. and Ph.D. degrees in computer engineering from Rutgers University, in 2003 and 2007, respectively. She was a TEES Associate Research Scientist with Texas A&M University, College Station, TX, USA. She is currently an Associate Professor with the Department of Electrical and Computer Engineering, Prairie View A&M University (PVAMU). She has actively participated in the Bioinformatics Training Program sponsored by NIH. Her research interests are in computational and systems biology, systems pharmacology, computer networking and communication, and artificial intelligence. She was a recipient of the Outstanding Researcher of the Year Award from the Roy G. Perry College of Engineering, PVAMU, in 2017.

ZHU HAN (Fellow, IEEE) received the B.S. degree in electronic engineering from Tsinghua University, in 1997, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Maryland, College Park, MD, USA, in 1999 and 2003, respectively. From 2000 to 2002, he was a Research and Development Engineer of JDSU, Germantown, MD. From 2003 to 2006, he was a Research Associate with the University of Maryland. From 2006 to 2008, he was an Assistant Professor with Boise State University, Boise, ID, USA. He is currently a John and Rebecca Moores Professor with the Electrical and Computer Engineering Department as well as with the Computer Science Department, University of Houston, Houston, TX, USA. His research interests include wireless resource allocation and management, wireless communications and networking, game theory, big data analysis, security, and smart grid. Since 2019, he has been an AAAS Fellow and ACM Distinguished Member. He received the NSF Career Award, in 2010, the Fred W. Ellersick Prize of the IEEE Communication Society, in 2011, the EURASIP Best Paper Award for the Journal on Advances in Signal Processing, in 2015, the IEEE Leonard G. Abraham Prize in the field of Communications Systems (Best Paper Award in IEEE JSAC), in 2016, and several best paper awards in IEEE conferences. From 2015 to 2018, he was the IEEE Communications Society Distinguished Lecturer. Since 2017, he has been 1% Highly Cited Researcher according to Web of Science.

LIJUN QIAN (Senior Member, IEEE) received the B.S. degree from Tsinghua University, the M.S. degree from the Technion-Israel Institute of Technology, and the Ph.D. degree from Rutgers University. He was a member of Technical Staff with Bell-Labs Research, Murray Hill, NJ, USA. He is currently a Regents Professor, where he holds the AT&T Endowment with the Department of Electrical and Computer Engineering, Prairie View A&M University (PVAMU), a member of the Texas A&M University System, Prairie View, TX, USA. He is also the Director of the U.S. DOD Center of Excellence in Research and Education for Big Military Data Intelligence (CREDIT Center). He is also a Visiting Professor with Aalto University, Finland. His research interests are in the areas of big data processing, artificial intelligence, wireless communications and mobile networks, network security and intrusion detection, and computational and systems biology.