This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2021.3085303, IEEE Transactions on Wireless Communications

Deep Learning Based Cellular Random Access Framework

Han Seung Jang, *Member, IEEE*, Hoon Lee, *Member, IEEE*, Tony Q. S. Quek, *Fellow, IEEE*, and Hyundong Shin, *Fellow, IEEE*

***Abstract*—Random access (RA) or preamble collision is one of the crucial problems in massive internet-of-things (IoT) at the network entry stage. Since a massive number of IoT nodes simultaneously attempt RAs on the same physical random access channel (PRACH), preambles may be selected by multiple nodes, incurring preamble collisions at the first step of the RA procedure. However, conventional RA models are limited to binary preamble detections which poses severe RA performance loss in the massive IoT environment. In this paper, we propose a deep learning (DL)-based end-to-end RA framework which has detection and resolution abilities for the collided preambles. In particular, advanced preamble classification and timing advance (TA) classifications are performed using deep neural networks (DNNs) for improving the probability of RA success while reducing the delay of the entire RA procedure. The effectiveness of the proposed DNN-based preamble and TA classifiers are demonstrated through extensive simulations. We further evaluate the system-level performance of the proposed DL-based RA model. It shows a significantly higher probability of instant RA success, which makes every node succeed in RA with very limited reattempts, and also maintains a significantly lower RA delay in massive IoT environment.**

***Index Terms*—Deep learning, internet-of-things, random access, preamble, collision detection, collision resolution.**

I. Introduction

MASSIVE connectivity is one of major requirements in 5G cellular networks. It has been actively studied under the name of massive machine-type communication (mMTC) or massive Internet-of-Things (IoT) targeting to provide broad wireless connectivity to tens of billions of machine devices with low-complexity and low-power characteristics [1]. A cellular network connection is initiated by random access (RA) on physical random access channel (PRACH) [2]. Through the initial RA, each device is synchronized with the network and transmits a scheduling request message for data transmission. In particular, IoT devices mostly stay in radio resource control (RRC) idle mode in order to reduce energy consumption, and they switch to RRC connected mode for data transmission via RA procedure. If a massive number of IoT devices simultaneously attempt RA for network connections, the network may experience access overload, and during the access overload period, most devices fail in network connections. To mitigate RA overload, several RA schemes have been proposed in massive IoT networks [3]–[6].

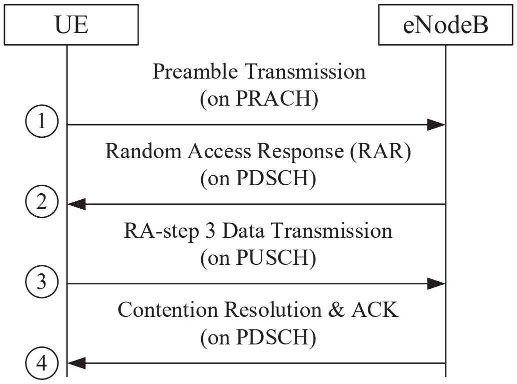

UE
eNodeB
Preamble Transmission
(on PRACH)
1
Random Access Response (RAR)
(on PDSCH)
2
RA-step 3 Data Transmission
(on PUSCH)
3
Contention Resolution & ACK
(on PDSCH)
4

Fig. 1. Conventional four-step RA procedure.

RA overload problems are induced by limitations of RA resources such as PRACHs and physical uplink shared channels (PUSCHs). Fig. 1 shows the conventional four-step RA procedure. In the conventional RA, a preamble collision occurs if two or more nodes randomly pick the identical preamble and transmit them on the same PRACH slot at the first step. If so, they receive the same uplink resource grant in the RA response (RAR) message at the second step. Then, the nodes transmit their data on the same PUSCH resource at the third step, and thereby, a PUSCH resource collision occurs. Consequently, the data decoding may fail, and this enables the eNodeB to recognize a preamble collision at the third step and notify the preamble collision to the corresponding nodes at the fourth step. This typical preamble collision detection method is very inefficient in terms of RA delay and PUSCH resource wastes. Thus, advanced preamble collision detection and notification

This work was supported in part by the NRF grant funded by the Korea government Ministry of Science and ICT (No. 2019R1F1A1061023, 2019R1F1A1060648), in part by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funced by the Korea government (MSIT) (No. 2021-0-00467, Intelligent 6G Wireless Access System), and in part by the National Research Foundation, Singapore and Infocomm Media Development Authority under its Future Communications Research & Development Programme, and MOE ARF Tier 2 under Grant T2EP20120-0006. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore and Infocomm Media Development Authority. *(Corresponding Author: Hoon Lee.)*

H. S. Jang is with the School of Electrical, Electronic Communication, and Computer Engineering, Chonnam National University, Yeousu, 59626, Republic of Korea. (e-mail: hsjang@jnu.ac.kr).

H. Lee is with the Department of Smart Robot Convergence and Application Engineering and the Department of Information and Communications Engineering, Pukyong National University, Busan, 48513, Republic of Korea. (e-mail: hlee@pknu.ac.kr).

T. Q. S. Quek is with the Information Systems Technology and Design Pillar, Singapore University of Technology and Design (SUTD), Singapore 487372, and also with the Department of Electronic Engineering, Kyung Hee University, Yongin 17104, South Korea. (e-mail: tonyquek@sutd.edu.sg).

H. Shin is with the Department of Electronic Engineering, Kyung Hee University, Yongin 17104, South Korea. (e-mail: hshin@khu.ac.kr).

© 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

methods are necessary.

It is ideal to quickly detect preamble collisions upon reception of preambles. However, it is a quite challenging problem in practical scenarios since multipath signals and delay spread of preambles cause significant interference to each other. Zhang *et al.* [7] proposed a preamble collision detection scheme by exploiting guard bands on PRACH. In this scheme, preamble collisions are detected with an extra ID on the guard bands. If a single ID is not decoded for a specific preamble, the eNodeB regards it as a preamble collision. However, the performance of preamble collision detection is limited by the bandwidth of the guard bands. Jang *et al.* [8] proposed a tagged preamble based preamble collision detection scheme. By superposing a tag sequence on to a preamble sequence, the extra tag information is delivered with the preamble, and if multiple tags are detected on a specific preamble, the eNodeB regards this preamble as a collisional preamble. However, such schemes [7], [8] require additional procedures to decode ID or detect tag during preamble detection in order to detect preamble collisions.

The conventional RA is executed in an orthogonal way, e.g., an orthogonal preamble transmission at the first step and an orthogonal data transmission at the third step. Non-orthogonal RA (NORA) techniques have been proposed [9]–[14]. The main idea of NORA techniques is a non-orthogonal resource scheduling at the second step of RA procedure and a non-orthogonal access to PUSCH resources at the third step of RA procedure. Liang *et al.* [9] proposed a NORA scheme based on successive interference cancellation (SIC). It exploits the difference in time of arrival to identify multiple nodes for multiplexing data in power domain. Jang *et al.* [14] provided a learning based power control scheme for NORA based on timing advance (TA) information. However, to make fully NORA system, non-orthogonal preamble transmission techniques with preamble collision resolution schemes are required. As one of solutions, we can utilize a preamble collision resolution scheme for non-orthogonal preamble transmission, which effectively detects a collision and captures multiple TA values at the first step of RA procedure [15]. In [16], using a fixed location feature of nodes, a collision avoidance is possible for nodes not to transmit data at the third step of RA procedure by identifying its TA value in the RAR messages. In [17], the authors proposed a massive MIMO-based grant free RA scheme with a collision resolution capability. An energy detection strategy is employed where the BS categorizes the status of the received preambles according to the conventional threshold-based classification.

Recently, deep learning (DL) approaches to wireless communication networks have been proposed [18]–[23]. Wireless resource management problems are addressed by the DL techniques for interference channels [18], cognitive radio networks [20], and device-to-device communications [19], [21]. In [22], modulation schemes are classified based on convolutional neural network (CNN) models such as AlexNet and GoogLeNet. These methods suggested new research opportunities of DL-based wireless network design to replace existing communication modules with deep neural networks (DNNs). A data-driven design is possible for the DL-based wireless communication systems by training DNNs with numerous samples. It has been reported that the DL methods can improve the performance of existing optimization algorithms with reduced real-time computation burden [18]–[21]. Ninkovic *et al.* [23] proposed an one-dimensional CNN based Wi-Fi packet detection method utilizing preamble in IEEE 802.11 frame. It can reduce long processing steps by data-based deep learning approaches, which simply insert the received signal sample into the trained model for a packet detection. The blind detection ability of the DL techniques is examined in the modulation classification [22] without the prior knowledge of the wireless channels. Such successes motivate us to investigate the feasibility of DNNs for handling technical limitations of the conventional RA model. However, it still remains unaddressed how to adapt the DL methods to the cellular RA systems.

DL techniques have been recently applied to design RA procedures in various configurations [24]–[26]. Ding *et al.* [24] considered a grant-free RA setup with multiple access points (APs) distributed in the network. Users first transmit the preamble signals, and the APs estimate the preamble multiplicity based on a well-trained DNN. Subsequently, the APs decode the data signals conveyed by the users. A joint decoding strategy at multiple APs is optimized by a K-means clustering algorithm. Based on the large scale antenna approximation, the multi-user interference is assumed to be canceled in the decoding process. Such an assumption is not viable for a moderate number of antenna cases in the LTE systems. Also, the grant-free configuration would not be straightforwardly employed for the current four-step RA procedure in the LTE standards. Sun *et al.* [25] proposed preamble detection and time of arrival estimation schemes based on CNNs. It is restricted to determine whether the received preamble is detected or not, thereby lacking the ability as a high-level collision resolution method. Jespersen *et al.* [26] presented a DL-based joint synchronization and channel estimation scheme for narrowband IoT (NB-IoT) system. The performance gain was achieved in estimating time-of-arrival, carrier-frequency offset, channel gain, and collision multiplicity. However, it cannot be straightforwardly applied to the LTE and 5G networks since the preamble format in these mobile systems is different from that in the NB-IoT systems.

In this paper, we introduce DL techniques to develop a novel end-to-end cellular RA framework. To the best of our knowledge, it is the first proposal of a fully non-orthogonal cellular RA framework including non-orthogonal preamble transmission at the first step and non-orthogonal data transmission at the third step based on DNNs. Most of existing works have dealt with only a non-orthogonal data transmission at the third step [9]–[14]. The proposed DL solution consists of two cascade DNN units, a preamble classifier and a TA classifier, which respectively perform detection and resolution of preamble collisions at the first step of the RA procedure. The preamble classifier is realized by a fully-connected neural network (FNN). It predicts the number of the nodes that utilize a specific preamble. The TA classifier, which is implemented with multiple CNNs, is further applied to the collided preambles for extracting the TA values of the collided nodes. The resulting TA values are exploited to resolve the preamble collision by properly assigning exclusive wireless resources or an identical wireless resource to the collided nodes. Consequently, the proposed DL-based RA model can be viewed as a general framework which includes existing collision detection (CD) [8] and collision resolution (CR) [15] models as special cases. Compared to the conventional techniques [8], [15] where a threshold-based binary decision is performed during the preamble detection, the proposed DL approach can improve the RA performance by learning statistical properties of fading channels from numerous training samples. Even though [24] proposed a preamble multiplicity estimator which has a similar concept to our proposed preamble classifier using deep learning techniques, the approach to preamble collision resolution is totally different from our proposed approach. In [24], preamble collisions can be resolved by multiple APs based on the K-means AP clustering algorithm while we focus on capturing multiple TA values based on the proposed TA classifier, and by using obtained TA values in the second step of the RA procedure, the BS allocates an exclusive PUSCH resource to each of collided device.

The effectiveness of the proposed DL-based RA models are examined through intensive numerical results with training samples generated by statistical and accurate LTE channel models. We validate the proposed DNN-based preamble and TA classifiers over baseline machine learning (ML) schemes. The system level performance of the DL-based end-to-end cellular RA framework is evaluated with various performance metrics. It is verified that the proposed DL-based cellular RA model shows a significantly higher probability of instant RA success, which makes every node can succeed in RA with very limited RA reattempts, and also maintains a significantly lower RA delay in massive IoT environment. As a result, it can support 10^5 nodes with 100% of RA success with maximum reattempts of 10 when we assume that each IoT device initiates RA every 30 seconds on average. The major contributions of this paper are summarized as follows:

- We first propose a novel DL approach for a fully non-orthogonal end-to-end cellular RA task including detection and resolution of the preamble collision as well as orthogonal or non-orthogonal resource scheduling strategies.
- We design a FNN structure for preamble classification to precisely predict the number of collided IoT nodes utilizing the identical preamble.
- We design a CNN structure for TA classification to identify multiple TA values from the received preamble in order to resolve preamble collisions.
- We provide extensive performance evaluations from classification accuracy to system level RA performance. Then, we show the significant performance enhancements of the proposed DL-based RA model, compared to those of conventional RA model.

The rest of this paper is organized as follows. In Section II, we briefly review the conventional RA model and its limitations, and Zadoff-Chu (ZC) preamble sequence. Section III presents overall procedures of the proposed DL-based RA framework, and the proposed DNN architectures and the corresponding training strategies are provided in Section IV. Numerical results are provided in Section V. In particular, the preamble and TA classification accuracies are evaluated via extensive simulations in Section V-A and V-B, respectively. In addition, system level RA performance is evaluated in Section V-C. Finally, conclusions are drawn in Section VI.

II. Conventional random access model

A. Procedure and critical limitation of conventional random access model

Fig. 1 shows the conventional four-step RA procedure:
(***Step 1***) **preamble transmission and detection**: A node randomly selects a preamble among available preambles, and sends it to the eNodeB on the PRACH slot. In practice, more than one node may occupy the same preamble due to the random preamble selection strategy [2].
(***Step 2***) **random access response (RAR)**: After detecting the preambles, the eNodeB sends the RAR messages which consist of the identifier of the detected preamble, the timing advance (TA) information, and an initial uplink PUSCH resource grant. These data will be utilized for uplink data transmission at the third step.
(***Step 3***) **uplink data transmission**: Using the PUSCH resource indicated via the RAR message at the second step, the node transmits an uplink data packet, a radio resource control (RRC) connection request, a tracking area update, or a scheduling request.
(***Step 4***) **ACK message transmission**: If the eNodeB successfully decodes the data transmitted by a single node, it sends back the ACK message including the node identity (ID) which is obtained from the decoded data. Otherwise, no message is conveyed from the eNodeB in order to notify a preamble collision to the nodes.

As discussed, the conventional RA works in an *orthogonal* manner since each node should utilize an exclusive preamble at the first step and exclusive PUSCH resources at the third step. Thus, limitations of the conventional RA model occur when two or more node select the identical preamble at the first step of RA procedure. During the preamble detection of the conventional RA model at the first step, the eNodeB only determines whether a specific preamble is *idle* or *active*, i.e., whether it is chosen by nodes or unused. For this reason, it is difficult for the conventional RA model to predict that an active preamble is chosen by a single node or multiple nodes.[1] If multiple nodes use the identical preamble and the corresponding RAR message, all relevant nodes convey their own data on the same uplink resource at the third step. In this case, it would not be easy for the eNodeB to decode the uplink data correctly. Such an event is called *'PUSCH resource collision'*. However, from this unsuccessful decoding (PUSCH resource collision), the eNodeB can recognize a preamble collision at the third step of the RA procedure.

[1] In a large cell, the receiver could differentiate the preamble transmissions of two nodes if they appear distinctly apart. However, in a small cell, a collision detection is very difficult since signals are overlapped within the maximum delay spread.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2021.3085303, IEEE Transactions on Wireless Communications

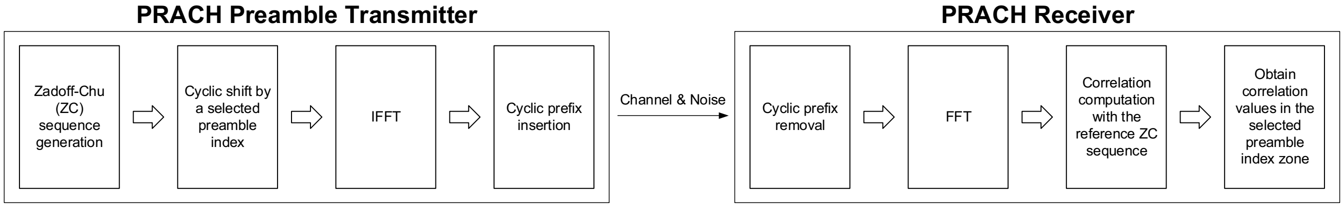

PRACH Preamble Transmitter
Zadoff-Chu (ZC) sequence generation
Cyclic shift by a selected preamble index
IFFT
Cyclic prefix insertion
Channel & Noise
PRACH Receiver
Cyclic prefix removal
FFT
Correlation computation with the reference ZC sequence
Obtain correlation values in the selected preamble index zone

Fig. 2. Procedure of obtaining the preamble correlation values

B. Zadoff-Chu sequence preamble

In LTE/LTE-A and 5G systems, preambles are generated based on the ZC sequence defined as

$$z_r[n] \triangleq \exp\left[\frac{-j\pi r n(n+1)}{N_{\mathsf{ZC}}}\right] \tag{1}$$

for $n = 0, \dots, N_{\mathsf{ZC}} - 1$, where N_{ZC} denotes the sequence length and $r \in \{1, \dots, (N_{\mathsf{ZC}} - 1)\}$ is the root number [27]. The ZC sequence has a *cyclic auto-correlation property* as

$$|c_{rr}[\tau]| \triangleq \left|\frac{1}{\sqrt{N_{\mathsf{ZC}}}} \sum_{n=0}^{N_{\mathsf{ZC}}-1} z_r[n] z_r^*[n+\tau]\right| = \sqrt{N_{\mathsf{ZC}}}\delta[\tau], \tag{2}$$

where $c_{rr}[\tau]$ represents the discrete cyclic auto-correlation function of $z_r[n]$ at lag τ, $\delta[\tau]$ denotes the Dirac delta function, and $(\cdot)^*$ denotes the complex conjugate. From (2), we can observe how much the received sequence $z_r[n+\tau]$ is shifted from the reference one $z_r[n]$. Another property is a *cyclic cross-correlation property*, where the magnitude of correlation $|c_{rk}[\tau]|$ between any two ZC sequences $z_r[n]$ and $z_k[n]$ with exclusive root numbers r and k, $\forall r \neq k$, is constant, i.e.,

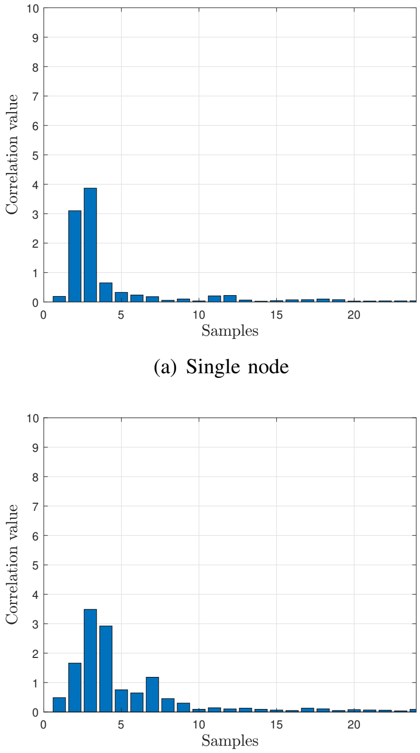
$$|c_{rk}[\tau]| = \left|\frac{1}{\sqrt{N_{\mathsf{ZC}}}} \sum_{n=0}^{N_{\mathsf{ZC}}-1} z_r[n] z_k^*[n+\tau]\right| = 1. \tag{3}$$

Basically, the i-th preamble $z_{r,i}[n]$ is given by the shifted version of the reference ZC sequence $z_r[n]$ by iN_{CS}, where $i \in \{0, \dots, (\lfloor N_{\mathsf{ZC}}/N_{\mathsf{CS}} \rfloor - 1)\}$ represents the preamble index and N_{CS} denotes the cyclic shift size. Hence, the i-th preamble $z_{r,i}[n]$ is expressed as

$$z_{r,i}[n] = z_r[(n + iN_{\mathsf{CS}}) \bmod N_{\mathsf{ZC}}], \tag{4}$$

where $A \bmod B$ represents the modulo operation. the eNodeB in the LTE systems utilizes fixed 64 prambles, and the number of available root numbers is $(N_{\mathsf{ZC}} - 1) = 838$ in a single cell [2].

Fig. 2 shows the PRACH preamble transmitter and the PRACH receiver. At the PRACH preamble transmitter, first, ZC sequence with the root number r, $z_r[n]$, is generated. To obtain the i-th preamble $z_{r,i}[n]$ of length 839, we cyclically shift $z_r[n]$ by iN_{CS}. It is then followed by 32768-point inverse fast fourier transform (IFFT) and two-step down samplings. After adding the cyclic prefix (CP), the preamble is transmitted in the form of 1920-length sequence. Through the PRACH, the noisy preamble sequence is observed by the receiver with J antennas. Discarding the CP, the receiver performs 1536-point FFT on the received signals and obtains the frequency-domain sequence $Y_j[k]$ at each antenna j for $j = 1, \dots, J$. Then, the receiver computes correlation values $|c_j[n]|^2$ between $Y_j[k]$ and the frequency-domain reference ZC sequence with the root number r, $Z_r[k]$. From the IFFT, we can calculate $|c_j[n]|^2$ in the frequency domain as



(a) Single node

(b) Two nearby nodes

Fig. 3. Comparison between collision-free and collisional preambles.

$$|c_j[n]|^2 = |\mathsf{IFFT}\{Y_j[k] Z_r^*[k]\}_n|^2 \tag{5}$$

for $n = 0, \dots, \hat{N}_{\mathsf{ZC}} - 1$, where $\hat{N}_{\mathsf{ZC}}$ denotes the length of the received preamble. By averaging correlation values from all J antennas, the averaged correlation value $\chi[n]$ is obtained as

$$\chi[n] = \frac{1}{J} \sum_{j=1}^{J} |c_j[n]|^2. \tag{6}$$

In the conventional RA model, to determine the i-th preamble is *idle* or *active*, a simple binary detection method is adapted at the the eNodeB which checks whether the averaged correlation values $\chi[n]$ exceed a threshold value within the i-th preamble detection zone $D_i \triangleq \{\chi[n] | i\hat{N}_{\mathsf{CS}} \leq n < (i+1)\hat{N}_{\mathsf{CS}}\}$ of size $\hat{N}_{\mathsf{CS}}$. Such a threshold-based binary detection fails to identify the collision event where a specific preamble is occupied by multiple nodes since it simply categorizes the

© 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

collided preamble into an *active* one. As a result, the same PUSCH resource is allocated to multiple nodes, resulting in the PUSCH resource collision at the third step of RA procedrue. An example is illustrated in Fig. 3 which presents the correlation values $\chi[n]$ in the 0-th preamble detection zone D_0 for (a) collision-free preamble utilized by a single node and (b) collisional preamble utilized by two nearby nodes. A collision-free preamble in Fig. 3 (a) has delayed multipath signals, and it looks similar to the collisional preamble in Fig. 3 (b). When the conventional RA model is applied to this example, the receiver may decide both the collision-free and the collisional preambles as *active* ones, and thus the collisional preamble in Fig. 3 (b) causes a PUSCH resource collision at the third step of RA procedure.

In general, detection of the preamble collision is a challenging task especially in small cell networks where multipath fading and delay spread of preambles may cause severe interference to each other. To tackle this difficulty, we propose a DL-based end-to-end RA framework in which the preamble are categorized into multiple classes according to the number of nodes occupying it. When the preamble is chosen by the multiple nodes, an additional DL processor is further employed to find the TA value of each node. The obtained TA information is then exploited to assign an exclusive PUSCH resource to avoid the PUSCH resource collision. The details of the proposed DL-based RA model are discussed in the preceding sections.

III. Proposed DL-Based RA Framework

In this section, we describe an overall process of the proposed DL-based RA framework as shown in Fig. 4. The proposed RA framework consists of two consecutive DNN units: *preamble classifier* and *TA classifier*. The preamble classifier first separates the preamble into several classes. The preamble classifier is commonly applied to all preamble zone indices i as shown in Fig. 4. It is then followed by the corresponding TA classifiers, which are implemented with multiple CNNs, to capture a single TA value of the node with a collision-free preamble and multiple TA values of the collided nodes that choose the same preamble. For a given parameter K, we realize $(K-1)$ different TA classifiers for each CLASS k, $\forall k = 1, 2, \ldots, K-1$, categorized by the identical preamble classifier. Especially, Fig. 4 shows that different TA classifiers are applied to Class $\mathrm{LASS1}, \ldots, (K-1)$ preambles, respectively.

Based on the captured TA values, for the third step of the RA procedure, we schedule exclusive PUSCH resources to the collided nodes. As will be discussed later, the proposed DL-based RA model can be viewed as a generalization of the existing techniques such as the CD model [8] and the CR model [15].

A. Preamble Classifier

The preamble classifier accepts $\chi[n]$ as an input and is trained to predict the number of the nodes using the received preamble. To this end, we consider the following $K+1$ different classes:

- **CLASS 0**: The preamble is idle (unused).
- **CLASS 1**: The preamble is selected by a single node (collision-free).
- **CLASS** k: The preamble is collided by k nodes for $k = 2, \ldots, K-1$.
- **CLASS** K: The preamble is collided by K or more nodes.

When the preamble belongs to CLASS 0, it is determined as an unused one, where no PUSCH resource will be scheduled not to waste the PUSCH resources. CLASS 1 represents the collision-free preamble occupied only by a single node, and thus no collision occurs among the nodes. In addition, we denote CLASS k for $k = 2, \ldots, K-1$ as the preamble collision case collided by k different nodes. In this scenario, the PUSCH resource collision event at the third step of the RA procedure can be efficiently avoided by allocating orthogonal PUSCH resources to the nodes. Therefore, compared to the conventional binary thresholding RA model, the proposed DL-based RA model can proactively improve RA performance by resolving preamble collisions through orthogonal PUSCH resource allocation to multiple nodes experiencing the preamble collision. Finally, the preamble categorized into CLASS K is assumed to be conflicted by K or more nodes. In this case, the eNode does not allocate PUSCH resources to these preambles. Hence, unlike the conventional RA model, the nodes experiencing preamble collisions can directly identify their preamble states, i.e., if the collision occurs at the first step of the RA procedure, based on the fact that they do not receive any RAR messages including their preamble indices.

It is notable that K is a design parameter and determines the total number of classes for the preamble classification task. The choice of K depends on the target of the RA model. For $K = 2$, the proposed RA framework can be viewed as the CD model [8] where the preamble collision is detected for CLASS 2 preamble, i.e., when two or more nodes utilize the same preamble. Also, for $K > 2$, the proposed RA framework includes the CR model [15] which resolves the preamble collisions for CLASS 2 to CLASS $(K-1)$ preambles.

In the proposed DL-based CR model, K should be carefully determined since it directly affects the overall RA performance. When K is a large number, the resolution capability of the preamble classifier can be enhanced, leading to the improved end-to-end RA performance. However, for a large K, CLASS K preamble rarely occurs in practice. In this configuration, the DL-based preamble classifier would overfit to this unusual case, resulting in a severe degradation of the generalization capability [28]. Furthermore, it is obvious that the training difficulty of the preamble classifier increases as K grows. These pose challenges in an efficient training of the preamble classifier, and the overall RA performance would not be enhanced as desired. We thus set K to a moderate number to construct a suitable DL model for the practical RA scenarios.

In order to evaluate the impact of the design parameter K, we need to calculate the probability that a single preamble is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2021.3085303, IEEE Transactions on Wireless Communications

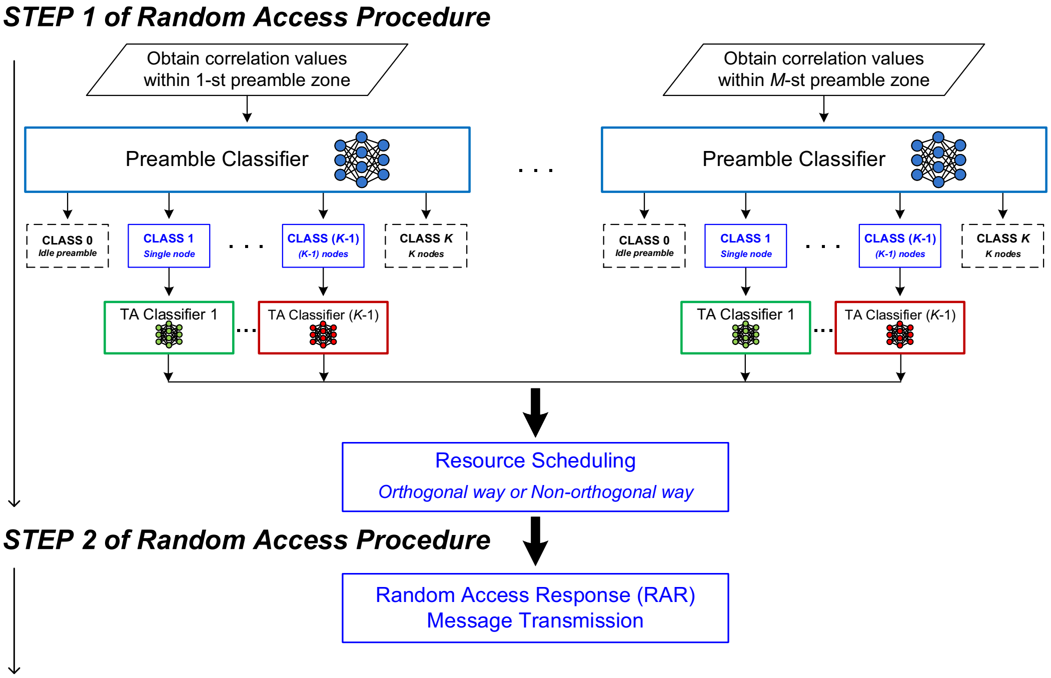


Fig. 4. Flow chart of the proposed DL-based RA model.

utilized by k nodes as follows:

$$P[k] = \sum_{n=0}^{\infty} \frac{\lambda^n}{n!} e^{-\lambda} \binom{n}{k} \left(\frac{1}{M}\right)^k \left(1 - \frac{1}{M}\right)^{(n-k)}, \qquad (7)$$

for $k = 0, \dots, n$, where n and M denote the number of nodes simultaneously attempting RA on the same PRACH slot and the number of available preambles, respectively, and λ represents the average number of nodes attempting access per PRACH slot. Fig. 5 shows $P[k]$ as λ varies 10 to 50 when $M = 50$. In particular, $\lambda = 50$ corresponds to the same number of available preambles $M = 50$, and this case is considered as a significantly heavy RA load case. Even in this case, we observed that $P[k = 3] < 0.1$, which implies that events for $k \geq 3$ are very rarely occurred. As a result, we decided to focus on the collision resolution for the case of $k = 2$, and it corresponds to $K = 3$. It is worth noting that the proposed CR model with $K = 3$ still has an ability to detect preamble collisions for $k \geq 3$.

B. TA Classifier

As shown in Fig. 4, the TA classifier operates only when the output of the preamble classifier is given by CLASS k for $k = 1, \dots, K-1$, i.e., if the input preamble is verified as the collision-free or the collided by k different nodes. The goal is to perform, namely, *k-node TA classification*, by predicting the TA value of k nodes for all possible $k = 1, \dots, K-1$. The resulting TA is then employed for PUSCH resource scheduling at the second step of RA procedure.

Without loss of the generality, we assume that L TA values exist in a cell. Then, there are total $\binom{L}{k}$ TA set candidates for the preambles identified as CLASS $k \in \{1, \dots, K-1\}$ by the preamble classifier. The TA resolution L is not that large in practical small cell environment, e.g., we have $L = 10$ for the cellular system with the cell radius 780 m [14]. Thus, the number of categories $\binom{L}{k}$ to be handled by the TA classifier is a moderate number in practice. Since the number of collided nodes k varies as $k = 1, \dots, K-1$, the TA classifier should be designed to be adaptive for any arbitrary given k. Also, the statistical properties of the received preamble at the eNodeB are affected by the number of collided nodes k. This implies that in practice, the TA estimation rules for each CLASS k are different from each other. To this end, we construct an individual neural network for each CLASS k to effectively capture the characteristics of the collision-free ($k = 1$) and collided preambles ($k > 1$). Each neural network k is independently trained to perform k-node(s) TA classification task with $\binom{L}{k}$ TA set classes. As a result, the

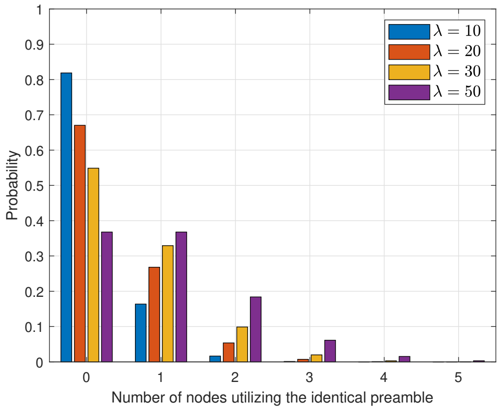


Fig. 5. Probability mass function for the number of nodes utilizing the identical preamble.

© 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

TA classifier unit consists of $(K-1)$ paralleled DNNs as illustrated in Fig. 4.

C. Resource Scheduling Strategy

In this subsection, we explain the resource scheduling strategy of the proposed DL-based NORA model. First, when the preamble is categorized into CLASS 0 or CLASS K by the preamble classifier, no uplink resources are allocated as mentioned before. For a CLASS 1 preamble, a single uplink resource is assigned. Next, for CLASS $k \in \{2, \ldots, K-1\}$, the preamble collision is resolved by the following procedure:
(1) RAR message generation: Through the TA classification for each CLASS k, the eNodeB obtains k TA values and generates k RAR messages for the i-th preamble, each of which includes the i-th preamble index, one of TA values, and an uplink resource grant (URG) for the third step of the RA procedure. Theses are conveyed to k nodes using the same preamble index i.
(2) Verifying RAR messages: Since each node is aware of its TA value from the previous successful RA[2], the node can obtain its own RAR message by verifying both its preamble index and TA value at the second step of the RA procedure [16], [29] in case that multiple RAR messages including its own preamble index are received. If TA values in multiple RAR messages are different from the node's TA value, the node can directly reattempt RA without proceeding the remaining steps. If it successfully obtains the RAR message, the node sends data on an exclusive PUSCH resource in orthogonal way or on the identical PUSCH resource in non-orthogonal way at the third step of the RA procedure. As a result, unlike the conventional RA model, although multiple nodes choose the identical preamble, the proposed DL-based RA model can succeed in transmitting data at the third step, which implies that the proposed DL-based RA model can work in a fully non-orthogonal way.

Fig. 6 shows an illustrative example with three nodes. Node x utilizes the preamble 2, whereas the preamble 9 on the same PRACH is selected by nodes y and z. For the preambles 2 and 9, the preamble classifier for the CR model with $K=3$ produces the classification result as CLASS 1 and CLASS 2, respectively. Once the eNodeB verifies that preambles 2 and 9 are occupied by a single node and two nodes, respectively, they are processed to the single-node and the two-node TA classifiers, respectively. Based on the TA classification results, three RAR messages are generated at the eNodeB. In particular, for node x, the preamble 2 has TA value of TA_x, and for the remaining nodes, the preamble 9 has two different TA values TA_y and TA_z. As a result, nodes y and z can successfully transmit their data at the third step, for instance, by assigning orthogonal PUSCH resources 2 and 3 to nodes y and z, respectively in Fig. 6 (a), and by assigning non-orthogonal PUSCH resource 2 to both nodes y and z in Fig. 6 (b). For the non-orthogonal scheduling, the eNodeB performs a SIC based on TA values [14].

IV. Proposed Neural Network Architecture

Constructing efficient DNNs is a nontrivial task since it highly relies on the target applications and the types of input features. In our RA applications, DNNs should be designed to learn the statistical features of the IoT nodes and the fading channels from the correlation values, which has not been studied yet in the literature. Thus, it is not straightforward to apply existing neural network architectures to the RA scenario. In this section, we propose DNN structures for the preamble classifier and the TA classifier units in Section III. The performance of the proposed neural networks will be verified in Section V via numerical results.

A. Data collection

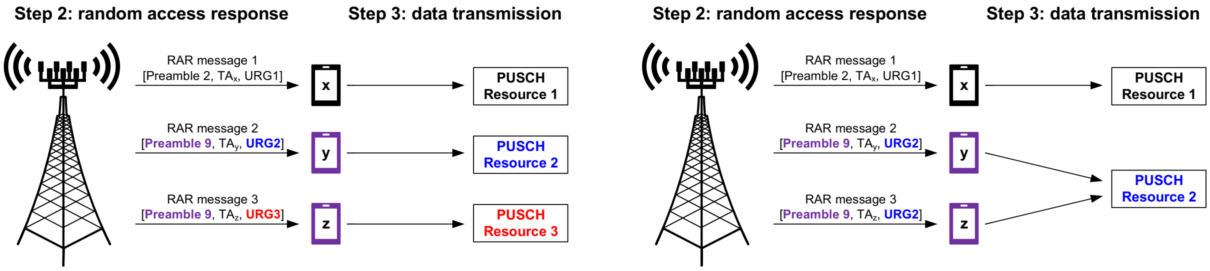
We generated all data using the MATLAB LTE Toolbox. For the preamble classification, a single set of input data consists of $\hat{N}_{\mathsf{CS}} = 24$ correlation values within a single preamble detection zone, which are calculated by the PRACH receiver at the end. Each set of input data has a label corresponding to $k \in \{0, \ldots, K\}$, which is output data in our preamble classification problem.

We consider five different signal-to-noise ratio (SNR) levels as $\mathrm{SNR} \in \{-14, -13, -12, -11, -10\}$ dB, and we call these a target SNR range for a model training. The reason for selecting this target SNR range is as follows. In general, the probability of successful preamble detection (classification for class 1 to class 1) should be exceeded 0.95. Thus, the target SNR for training should be guaranteed for this. In fact, the SNR of below -14 dB yields the successful preamble detection probability of smaller than the practical requirement 0.95. Also, the false alarm probability should not exceed 0.001 both for the conventional RA and the proposed framework. Observing lower SNR values can improve the performance of the proposed DL method, but it no longer satisfies the false alarm probability constraint. Thus, the minimum target SNR value is set to -14 dB for fair comparison. In addition, since employing a wide range of SNR values can improve the generalization ability of the DNNs, we thus consider 1 dB resolution for the training SNR. Lastly, our choice for the maximum SNR value of -10 dB is verified through the numerical results.

At each SNR, we generate 10^4 data samples for each CLASS $k \in \{0, \ldots, K\}$, resulting in total $T = 5(K+1) \times 10^4$ data samples. We utilize 70 % and 30 % of data sets for training and test, respectively. In addition, depending on the channel environments, we collected different set of input and output data for the preamble classification problem.

In the same way to the preamble classifier, for TA classification, a single set of input data consists of $\hat{N}_{\mathsf{CS}} = 24$ correlation values within a single preamble detection zone, which are calculated by the PRACH receiver at the end. However, different from the preamble classification, each set of input data has a label corresponding to one of 10 and 45 TA set classes, which are determined by nodes' locations, in the single-node TA classifier and the two-nodes TA classifier, respectively, which is output data in our TA classification problem. By varying nodes' locations uniformly in a cell, we

[2] A successful RA is achieved when a single RAR message including a node's preamble index and a TA value is received, and the node transmits a step-3 message successfully. The TA value obtained by such a successful RA is the node's current TA value.


Step 2: random access response
Step 3: data transmission
RAR message 1
[Preamble 2, TA$_x$, URG1]
x
PUSCH Resource 1
RAR message 2
[Preamble 9, TA$_y$, URG2]
y
PUSCH Resource 2
RAR message 3
[Preamble 9, TA$_z$, URG3]
z
PUSCH Resource 3
Step 2: random access response
Step 3: data transmission
RAR message 1
[Preamble 2, TA$_x$, URG1]
x
PUSCH Resource 1
RAR message 2
[Preamble 9, TA$_y$, URG2]
y
PUSCH Resource 2
RAR message 3
[Preamble 9, TA$_z$, URG2]
z

(a) Orthogonal resource scheduliung

(b) Non-orthogonal resource scheduliung

Fig. 6. An example of collision resolution with resource scheduling: (a) orthogonal way and (b) non-orthogonal way.

can generate data samples for each TA set. To be specific, in the CD and the CR models, CLASS 1 preambles proceed to the single-node TA classifier, and it gives a single TA value among 10 candidates. In addition, the two-nodes TA classifier is only applied to the CR model and provides two TA values with total $\binom{10}{2} = 45$ TA set classes for each of two nodes. We also consider five different signal-to-noise ratio (SNR) levels as SNR $\in \{-14, -13, -12, -11, -10\}$ dB. At each SNR, we generate 10^4 data samples for each TA set class, resulting in total 5×10^5 and 2.25×10^6 data samples for the single-node and two-nodes TA classifiers, respectively. We utilize 70 % and 30 % of data sets for training and test, respectively. In addition, depending on the channel environments, we collected different set of input and output data for the TA classification problem.

B. Preamble classifier

For designing the preamble classifier, it requires to check the number of the collided nodes from the correlation values. To this end, one can examine the amplitude of the correlation values $\chi[n]$ in a specific (i-th) preamble detection zone D_i. In other words, all the input correlation values in the i-th preamble detection zone would evenly affect the preamble classification task. Different preamble index only affects the shift of ZC sequence, and thus, we utilize the 0-th preamble detection zone for obtaining the correlation values. For this reason, we employ a FNN as shown in Fig. 7 for the preamble classifier with five fully-connected (FC) layers[3] where all input elements in the 0-th detection zone contribute to the output. Each hidden layer (layers 1-4 in Fig. 7) consists of 240 neurons and the hyperbolic tangent function $\tanh(x) \triangleq \frac{e^x - e^{-x}}{e^x + e^{-x}}$ is applied as activations.[4] In contrast, for the output layer (layer 5 in Fig. 7), we adapt the softmax activation with $(K+1)$ neurons for the classification task of $(K+1)$ different classes. As a result, the output of the preamble classifier becomes the probability vector $\mathbf{p} \triangleq [p_0, p_1 \dots, p_K]^T$ of length $(K+1)$ where each element p_k for $k = 0, 1, \dots, K$ represents the probability of the preamble being predicted as CLASS k. The input feature to the preamble classifier is the correlation vector $\boldsymbol{\chi} \triangleq [\chi[0], \dots, \chi[\hat{N}_{\mathsf{CS}} - 1]]^T$ in the 0-th preamble detection zone D_i of length $\hat{N}_{\mathsf{CS}} = 24$ in a cell with a radius of 780 m. Then, the maximum index of the probability vector $\mathbf{p}$, i.e., $k = \arg\max_{0 \le k \le K} p_k$, is selected as the class of the input correlation vector.

The supervised learning concept is applied to train the preamble classifier so that the FNN can learn an unknown input-output relationship between the correlation vector $\boldsymbol{\chi}$ and its ground-true label $k_{\mathsf{true}} \in \{0, 1, \dots, K\}$. A training set $\mathcal{T}$ of size T comprises pairs of the correlation vector and the label as $\mathcal{T} = \{(\boldsymbol{\chi}^{(t)}, k_{\mathsf{true}}^{(t)}) | t = 1, \dots, T\}$, where the superscript t stands for the index of a training sample. For the training cost, we employ the categorical cross-entropy function $f(k_{\mathsf{true}}, \mathbf{p}) \triangleq -\log(p_{k_{\mathsf{true}}})$ which measures the affinity between the ground-true label k_{true} and the $(k_{\mathsf{true}}+1)$-the element of the softmax output $\mathbf{p}$, $p_{k_{\mathsf{true}}}$ characterizing the probability of CLASS k_{true}. The overall cost F of the FNN evaluated over the training set $\mathcal{T}$ is written as

$$F = \frac{1}{T} \sum_{t=1}^{T} f\left(k_{\mathsf{true}}^{(t)}, \mathbf{p}^{(t)}\right), \tag{8}$$

where $\mathbf{p}^{(t)}$ is the output of the FNN corresponding to the t-th training sample $\boldsymbol{\chi}^{(t)}$.

The preamble classifier is trained to minimize the cost function in (8). This can be achieved by the stochastic gradient descent (SGD) algorithm, which is a powerful stochastic optimization tool in the DL [28]. The training step is an offline procedure and can be performed in advance before the real-time RA services, and thus the training complexity is independent of the online realization of the trained preamble classifier. After the training, the online computation of the FNN is processed by simple linear matrix multiplications and additions.

C. TA classifier

For the preamble classification task, we have focused on all the elements of the correlation vector $\boldsymbol{\chi}$ via the FNN structure with fully-connected layers. On the contrary, the TA of the node would affect both the amplitude and the delay of the received preamble obtained through multi-path fading channels. Thus, an efficient TA classifier should capture temporal changes in the correlation value $\boldsymbol{\chi}$. To this end, we use a CNN structure in Fig. 8 for the TA classification

[3] In general, as the number of layers in the FNN model increases, the accuracy of the FNN model can be enhanced. However, with too many layers, the model may yield an overfitted classification result. Based on this fact, we found five FC layers with 240 neurons in the preamble classifier yields the best result for the test data.

[4] We have found from simulations that the hyperbolic tangent function works better than a rectified linear unit (ReLU).

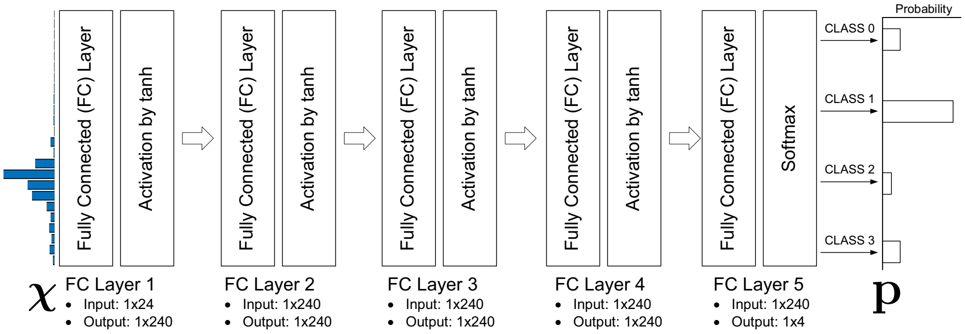


Fig. 7. FNN for preamble classification

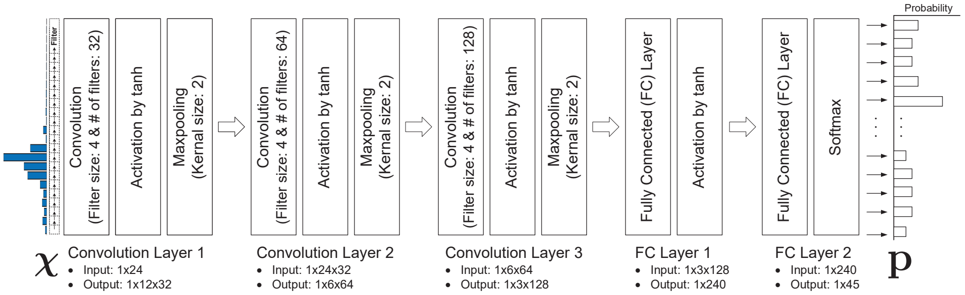


Fig. 8. CNN for TA classification

task. The CNN has been well known as an efficient feature extractor for visualization patterns, i.e., images in the computer vision field [28]. However, the applications of the CNN are not simply limited to the image processing. The windowing nature of the CNN can reveal any stochastic correlations in input vectors. For this reason, the CNN has been also employed in one-dimensional (1D) signal processing applications such as modulation classification [30] and signal reconstruction [31]. These results imply that the basic design philosophy of the CNN, which has been originally developed for efficiently capturing the spatial correlation of two-dimensional images, is valid regardless of the dimension and the types of the inputs. In our scenario, there exists temporal correlations for the received ZC sequences since the impairment of the multipath fading is modeled as the temporal 1D convolution operation. This can be observed from Fig. 3 where the correlation values of the collision-free and collision cases show different amplitude-delay aspects. This motivates us to utilize the CNN architecture for the design of the TA classifier whose target is to identify the number of the collided nodes by observing the correlation values.

The CNN architecture for the TA classifier[5] is depicted in Fig. 8 where the first three layers are implemented with convolutional layers followed by maxpooling operations, while the two FC layers are added for the multi-class classification. Unlike FC layers in the preamble classifier where all input elements contribute to an output of hidden layers, convolutional layers extract spatially-correlated features by applying a windowing filter to an input. Each convolutional layer of the TA classifier is followed by pooling units which reduce the output dimension of the CNN. Among various pooling operations, the maxpooling layer has been widely utilized in a typical CNN [28]. The maxpooling operation makes the CNN be invariance to small changes in the input vector and become robust to the noise in the training samples [28]. As a result, the CNN can focus on the important parts by capturing the higher-level features of input patterns.

The input to the TA classifier is 1D correlation vector $\boldsymbol{\chi}$ of length $\hat{N}_{\mathsf{CS}}$. Thus, the convolutional filters of the proposed TA classifier are set to 1D vectors to carry out the 1D signal convolution. To be specific, in each convolution layer, the size of the convolution filter and the stride are given by 4 and 1, respectively, and we fix the size of the maxpooling kernel and the stride as 2 and 2, respectively. We increase the number of the filters at convolutional layers 1, 2, and 3 as 32, 64, and 128, respectively. Thus, output dimensions of convolutional layers 1, 2, and 3 become $(12, 32)$, $(6, 64)$, and $(3, 128)$, respectively. As illustrated in Fig. 8, the output of convolutional layer 3 is reshaped to a 1D vector of length $3 \times 128 = 384$ so that it can be processed by two consecutive FC layers each of which has the output dimension 240 and $\binom{L}{k}$ for the k-node TA classification procedure. For the activations, we employ the hyperbolic tangent function at the hidden layers, i.e., convolutional layers 1-3 and FC layer 1, whereas the softmax function is utilized at FC layer 2 for the TA classification.

Similar to the preamble classifier, the training cost function of the TA classifier is set to the categorical cross entropy between the output probability vector of the CNN and the ground-true TA label. Therefore, the training set contains pairs of the correlation vector and its TA label for k nodes. The TA classification typically has a larger number of classes compared to the preamble classifier. For instance, in a small

[5] By considering a generalization capability, we found this CNN architecture showing the best classification accuracy for the test data.

cell with the radius of 780 m, $L = 10$ different TA values exist. When the preamble classifier outputs CLASS $k = 2$ for the received preamble, then we need to examine total $\binom{10}{2} = 45$ TA set candidates for the TA classification task. For this reason, training of the TA classifier would be more difficult than that of the preamble classifier. In order to further improve the TA classification accuracy, we utilize ensemble learning methods [32]. In the ensemble learning approach, multiple independent CNNs are created for the k-node TA classification, and multiple output results are combined in order to form a better hypothesis. We expect the ensemble CNN model to achieve a better predictive performance than a single CNN.

The training tasks of the preamble and TA classifiers can be performed at the eNodeB in an offline manner by collecting sufficiently large number of training samples in advance. The training data can be generated through computer simulations, or artificial field experiments as in [30], [33]. To be specific, we randomly choose transmitting nodes among the IoT devices deployed in the test network. We force them to use the identical preamble to generate a single training sample, i.e., the correlation values $\chi[n]$ in (6). Then, the training label of the correlation values, which is the number of the transmitting nodes, are readily obtained. Such an experiment can be repeatedly performed until sufficient number of samples are obtained. Notice that this process is only for collecting the training data. After the training step, the trained classifiers are implemented at the eNodeB by means of memory units storing the neural networks for the real-time RA service. In this online RA service, training samples are no longer required. Thus, the communication overhead for the training data collection does not affect the RA performance.

The input to the trained classifiers is the correlation vector $\boldsymbol{\chi}$ in the 0-th preamble detection zone whose length is fixed as $\hat{N}_{\text{CS}} = 24$, which is independent of the number of the nodes. Also, the number of hidden layers and neurons are fixed and do not vary for the number of the nodes. As discussed, the eNodeB is responsible for the prediction of the preamble collision events by using the trained neural networks, whereas the nodes have no responsibility for the DL calculations. Therefore, the proposed DL approaches lays no additional computational burdens on the IoT devices. Consequently, the computation complexity as well as the corresponding energy consumption of the DL classifiers do not depend on the number of the nodes. As will be shown in the simulation results, the proposed DL approaches can tackle massive IoT scenarios with 10^5 devices.

V. Numerical Results

In this section, we evaluate the performance of the proposed DL-based RA framework. Two different RA model is considered: the CD model with $K = 2$ and the CR model with $K = 3$. We first examine the test accuracy of the preamble classifier and the TA classifier units, respectively. Then, the system level performance of the end-to-end RA procedure for the proposed DL-based CD and CR models is further evaluated in the cellular environment. Detailed metrics of system level performance are explained in Section V-C. Table I shows specific simulation parameters for preamble classification and TA classification. Specifically, we evaluate the performance of cellular random access in a small cell with the radius of 780 m. According to 78 m of the resolution of TA value, 780 m cell classifies $L = 10$ different TA values.

TABLE I
Simulation parameters of preamble and TA classifications

Parameters	Values
Cell radius	780 m
Resolution of TA value	78 m
Total number of TA values	10
Cyclic shift size (N_{CS})	13
Size of preamble detection zone ($\hat{N}_{\text{CS}}$)	24
Number of receive antennas (J)	2, 4, and 8
Channel environments	EPA and ETU

Two LTE channel models, i.e., extended pedestrian A (EPA) and extended typical urban (ETU) [34], are considered to reflect statistical features of practical wireless environment such as shadowing, multipath effects, and the fast fading characterizing the mobility of the nodes during the transmission of preambles. The classifiers are trained with numerous preambles received through wireless fading generated based on these practical channel models. This enables the classifiers to extract useful statistics, e.g., delay spread and amplitudes of the correlation values in Fig. 3, from the received preambles for improving the CR and the CD performance. Consequently, the RA system can be successfully optimized without the perfect knowledge of the channels.

A. *Performance of preamble classifier*

In this subsection, we demonstrate the effectiveness of the FNN-based preamble classifier accuracy for the CD model and the CR model, respectively. For comparison, the classification performance of two shallow ML schemes are also investigated. In particular, we examine ensemble bagging tree (EBT) [35] and support vector machine (SVM) [36] as baseline learning techniques, which have been widely employed in classification applications.

Table II summarizes the preamble classification accuracies of the proposed FNN and the baseline schemes according to the channel models and the number of receiver antennas. Recall that the proposed FNN-based preamble classifier has 3 and 4 preamble classification outputs for the CD and the CR models, respectively. Regardless of the CD and the CR models, the proposed FNN outperforms the baseline EBT and the SVM methods, especially for the ETU channel model. In all schemes, the preamble classification of the CD model shows higher accuracies than those of the CR model due to the fact that the CR model has one more class. We can see that the classification accuracy increases as the number of receive antenna gets larger since the averaged correlation values for each preamble are more stable, and it helps to improve the classification performance. Especially, in the ETU models, the

TABLE II
PREAMBLE CLASSIFICATION ACCURACY

(a) EPA channel model

Schemes	2 antennas		4 antennas		8 antennas	
	CD	CR	CD	CR	CD	CR
Proposed FNN	0.968	0.931	0.996	0.984	0.999	0.996
EBT	0.960	0.892	0.991	0.957	0.997	0.982
SVM	0.950	0.860	0.990	0.938	0.998	0.978

(b) ETU channel model

Schemes	2 antennas		4 antennas		8 antennas	
	CD	CR	CD	CR	CD	CR
Proposed FNN	0.941	0.862	0.975	0.934	0.996	0.974
EBT	0.914	0.827	0.969	0.901	0.992	0.955
SVM	0.935	0.784	0.973	0.893	0.991	0.948

TABLE III
TA CLASSIFICATION ACCURACY

(a) EPA channel model

Schemes	2 antennas		4 antennas		8 antennas	
	1 TA	2 TAs	1 TA	2 TAs	1 TA	2 TAs
Proposed CNN	0.991	0.943	0.999	0.992	1.000	0.999
EBT	0.990	0.929	0.999	0.989	1.000	0.998
SVM	0.967	0.899	0.993	0.977	0.998	0.995

(b) ETU channel model

Schemes	2 antennas		4 antennas		8 antennas	
	1 TA	2 TAs	1 TA	2 TAs	1 TA	2 TAs
Proposed CNN	0.959	0.799	0.996	0.949	0.999	0.994
EBT	0.948	0.764	0.993	0.933	0.999	0.990
SVM	0.941	0.773	0.991	0.936	0.999	0.991

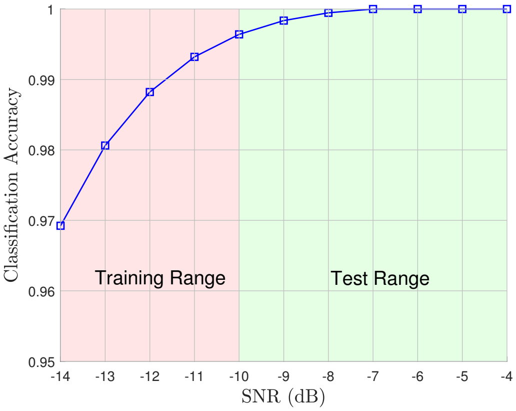


Fig. 9. Classification accuracy for in-range SNR and out-range SNR with two receive antennas.

TABLE IV
SYSTEM LEVEL SIMULATION PARAMETERS

Parameters	Values
Cell radius	780 m
Number of nodes, N	$10000 \sim 100000$
Average access arrival rate per node	2 per minute (Poisson)
PRACH configuration index	6
Period of PRACH time slot, T_{RACH}	10 msec
Number of available preambles, M	50
Number of schedulable PUSCH resources	50
Maximum number of reattempts, Q	10
RAR window size	5 msec
Backoff indicator	20 msec
mac-ContentionResolutionTimer	48 msec

preamble classification accuracy of the proposed CR model with 2 antennas is poor as 0.862, while it is significantly improved as 0.934 and 0.974 with 4 antennas and 8 antennas, respectively.

In practical scenario, the received SNR of preamble can be out of the training range. To see the generalization ability of the preamble classifier, we evaluate the classification accuracy with respect to the SNR. The preamble classifier is trained only at the predetermined SNR range SNR $\in \{-14, -13, -12, -11, -10\}$ dB, but its performance is examined in unseen high SNR regime above -10 dB by Fig. 9. It is interesting to see that the preamble classifier still performs well at unseen SNR values. This implies that the considered training SNR range is sufficient to achieve the generalization capability to a wide range of SNR values.

B. Performance of TA classification

In this subsection, we evaluate the TA classification accuracy for CLASS $k = 1$ and 2. Note that CLASS 1, i.e., the single-node TA classification, appears both in the CD and the CR models, whereas we only have the two-node TA classification of CLASS 2 preamble in the CR model. We provide the TA classification performance in Table III of the proposed CNN and the baseline methods for different channel models. First, the proposed CNN-based TA classifier outperforms the baseline ML methods regardless of the number of antennas and the number of TAs. In general, the single-node TA classification shows a significant improvement in the classification accuracy even with two receive antennas in EPA. In the ETU, the TA classification accuracy is a little bit degraded. However, with 8 receive antennas, the TA classification accuracy is almost unity based on the baseline methods both in the EPA and the ETU environments. It is observed that with a small number of receive antennas in the ETU environment, the two-node TA classification is more challenging compared to the single-node case. For example, the proposed CNN-based two-node TA classifier shows a poor accuracy of 0.799 with two receive antennas in the ETU.

C. System level performance

In this subsection, we evaluate the system level performance of the proposed end-to-end DL-based RA framework for the cellular setup specified in Table IV. The four-step RA performance is investigated by jointly considering the available preamble resources, the preamble classification accuracy, and TA classification accuracy at the first step of RA procedure as well as the schedulable PUSCH resources at the second step of RA procedure. We equal the number of schedulable PUSCH resources to the number of available preambles[6] in order to exclude the case that the eNodeB cannot allocate PUSCH resources due to lack of available PUSCH resources. We use the average delay time of each RA procedure specified in [37], [38] in simulations. We compare the performance of five RA models: 1) the conventional RA model, 2) early preamble collision detection (eCD) model [8], 3) the proposed CD model, 4) collision avoidance model [16], and 5) the proposed CR model in terms of following common performance metrics:

- ***Probability of instant RA success by a single attempt***: The ratio of the number of successful RAs to the number of total RA attempts.
- ***Probability of final RA success considering all reattempts***: The ratio of the number of successful RAs to the number of new RA initiations.
- ***Collision probability***: The ratio of the number of collisions to the number of total RA attempts.
- ***PUSCH utilization efficiency***: The ratio of the number of successful RAs to the number of allocated PUSCH resources.
- ***Average random access delay***: The average of the RA delay time for successful RAs

In the conventional RA, eCD, and collision avoidance models, we utilize preamble detection and collision detection probabilities within the test SNR range of $\{-14, -13, -12, -11, -10\}$, which are obtained from [10].

In addition to common metrics, specific performance metrics *only applicable to the proposed CD model* are as follows:

- ***Probability of successful collision detection***: The ratio of the number of successfully predicted CLASS 2 preambles to the number of transmitted CLASS 2 preambles.
- ***Probability of RAR reception failure***: The ratio of the number of erroneous preamble classification into CLASS 0 or CLASS 2 to the number of transmitted CLASS 1 preambles.
- ***Probability of erroneous TA reception (TA confusion)***: The ratio of the number of erroneous TA classifications to the number of transmitted CLASS 1 preambles.

The RA failure in the proposed DL-based CD model occurs 1) when *no RAR messages are provided* due to preamble classification errors of CLASS 1 preamble to CLASS 0 or CLASS 2 preambles and 2) when *a node is synchronized with an erroneous TA value* instead of its own TA value due to TA classification errors of CLASS 1 preambles. Please remind that the receiver at the eNodeB successfully performs the CD by predicting the CLASS 2 preamble as a CLASS 2 preamble. However, the CD fails due to the classification error of CLASS 2 preamble to CLASS 1 preamble, which is mainly caused by TA overlap of two nearby nodes with the same TA value utilizing the identical preamble. It is notable that in the proposed CD model, TA overlap results in PUSCH resource collisions at the third step of RA procedure.

Next, specific performance metrics *only applicable to the proposed DL-based CR model* are as follows:

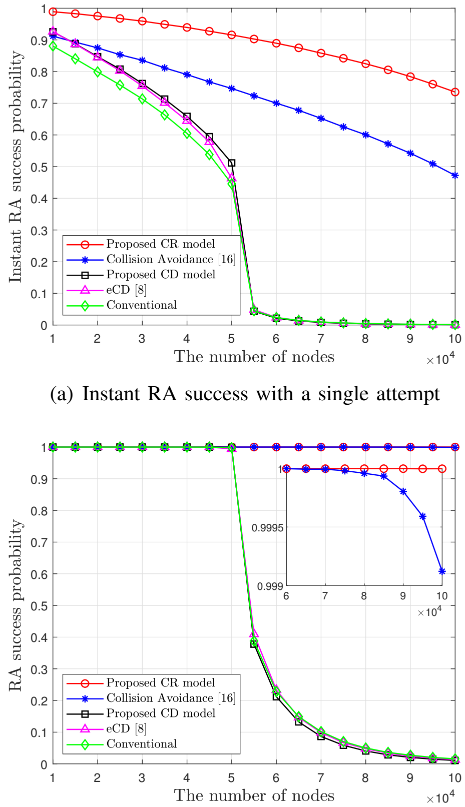
- ***Probability of successful collision resolution***: The ratio of the number of successfully predicted CLASS 2 preambles to the number of transmitted CLASS 2 preambles.
- ***Probability of successful collision detection***: The ratio of the number of successfully predicted CLASS 3 preambles to the number of transmitted CLASS 3 preambles.
- ***Probability of erroneous TA reception (TA confusion)***: The ratio of the number of erroneous TA classifications to the number of transmitted CLASS 1 and 2 preambles.

In the proposed CR model, the RA may fail since 1) *no RAR messages are provided* due to preamble classification errors of CLASS 1 and CLASS 2 preambles to CLASS 0 or CLASS 3 preambles and 2) *there is no TA value matched with that of a node in RAR messages* due to TA classification errors of CLASS 1 and CLASS 2 preambles. In practice, the CR fails mostly when two nearby nodes with the same TA value utilize the identical preamble. In this case, the classification error of CLASS 2 to CLASS 1 preambles occurs, and then both nodes transmit data on the same PUSCH resource at the third step of RA procedure since the received RAR message includes the utilized preamble index, the TA value, and a single PUSCH resource grant.

Figs. 10 (a) and (b) plot the instant RA success probability and the final RA success probability considering all reattempts, respectively, by varying the number of nodes in the cell. The proposed CR model outperforms other models due to the ability for resolving preamble collisions. Especially, the proposed CR model shows significantly higher instant RA success probabilities and no RA failure incurs even if the number of nodes increases up to 10^5. The degradation of instant RA success probability of the proposed CR model is mainly due to the increases in the number of CLASS 3 preambles which cannot be resolved, but only be detected. We observe that the classification errors of the preamble and TA predictions, which cause the RAR reception failure and the erroneous TA reception, marginally affect the instant RA success probability. In addition, the instant RA success probability of the collision avoidance model is much less than that of the proposed CR model, which causes higher RA delay, and lower PUSCH utilization efficiency. Due to multiple RA reattempts, the collision avoidance model can achieve the final RA success probability of one, but, from 7×10^4 nodes, this probability decreases, which may be problematic in ultra-reliable communications.

In contrast, the final RA success probability considering all reattempts drastically decreases after 5×10^4 nodes in both of the proposed CD, the eCD, and conventional models. Although the proposed CD model can detect preamble collisions at the first step, it cannot improve the final RA success probability considering all reattempts, but help to avoid PUSCH resource collisions at the third step of the RA procedure.

[6] The shortage of PUSCH resources causes significant performance degradation for the conventional model since it allocates PUSCH resources to all active preambles including collision-free and collisional preambles. On the other hands, the proposed CD model selectively allocates PUSCH resources only to collision-free preambles. Details can be found in [8].



(a) Instant RA success with a single attempt

(b) Final RA success considering all reattempts

Fig. 10. (a) Probability of instant RA success with a single attempt and (b) Probability of final RA success considering all reattempts.

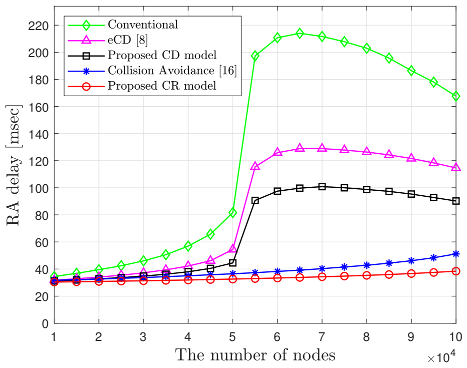


Fig. 11. Average of random access delay time

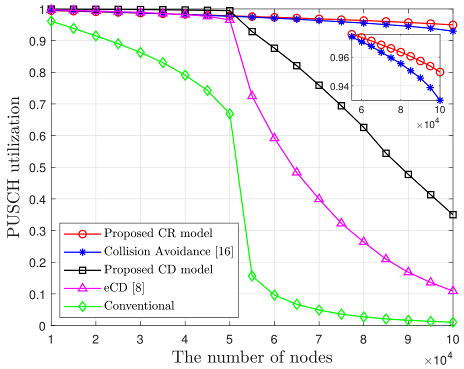


Fig. 12. PUSCH utilization efficiency.

In Fig. 11, we present the average RA delay time of successful nodes for each model as a function of the number of nodes. We first observe that the proposed CD model can reduce RA delay compared to the conventional and eCD models while achieving the similar RA success probability as shown in Fig. 10. This is because the proposed CD model can quickly detect preamble collisions with a higher probability than that of the eCD model at the first step and notify these at the second step. As a result, the nodes do not need to proceed to the remaining RA steps and directly reattempt RA. The proposed CR model maintains a significantly lower RA delay only including the time for a single four-step RA procedure even with a massive number of nodes. However, the collision avoidance model can avoid some PUSCH resource collisions, but unsuccessful nodes in RA still remain since they cannot receive their own PUSCH resource grants at the second step. In the proposed CD, eCD, and conventional models, the decrease in the RA delay after peaks is due to the fact that we only calculate the average RA delay time from RA success nodes.

Fig. 12 exhibits PUSCH utilization efficiency for each model by varying the number of nodes. The proposed CR and collision avoidance models can efficiently utilize the PUSCH resources with a small number of PUSCH resource wastes compared to the proposed CD, eCD, and conventional models. These PUSCH resource wastes in the proposed CR model are mainly caused by PUSCH resource collisions of two nearby nodes with the same preamble and TA value since the proposed CR model cannot detect and resolve preamble collisions of two nodes with the same TA value. However, the PUSCH utilization efficiency of the collision avoidance model becomes lesser than that of the proposed CR model from 6×10^4 nodes. Compared to the proposed CR model, we also observe that the PUSCH utilization efficiency of the proposed CD model drastically decreases after 5×10^4 nodes since successful RAs rarely occur and the collision detection failures from the erroneous classification for CLASS 2/3 to CLASS 1 preambles. In contrast, the conventional model shows a very poor utilization of PUSCH resources since it allocates PUSCH resources to all *active* preambles including collision-free and collided preambles.

In order to evaluate the proposed CD model in detail, we investigate the collision detection performance of the proposed CD model. In order to analyze the probability of successful collision detection, we first obtain collision probability. Let p_{c} denote the collision probability. The total RA arrival rate in the i-th PRACH slot is expressed as [39]: $\lambda_{\text{T}}[i] = \lambda \cdot N \cdot T_{\text{RACH}} + \eta \cdot p_{\text{c}} \cdot \lambda_{\text{T}}[i-1]$, where λ denotes the RA arrival rate of a device (sec^{-1}), N denotes the number of machine nodes in a single cell, T_{RACH} denotes the period of PRACH time slot, and $\eta \cdot p_{\text{c}} \cdot \lambda_{\text{T}}[i-1]$ represents the reattempted RA arrival rate from the previous slot due to collisions. η denotes the reducing factor expressed as $\eta = \frac{\sum_{i=1}^{Q-1} p_{\text{c}}^i}{\sum_{i=1}^{Q} p_{\text{c}}^i}$, where Q is the maximum number of RA

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TWC.2021.3085303, IEEE Transactions on Wireless Communications

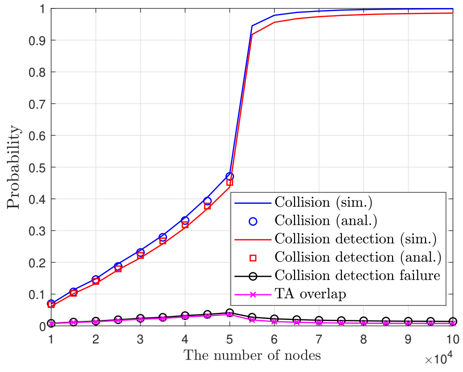


Fig. 13. Collision detection performance of the proposed CD model.

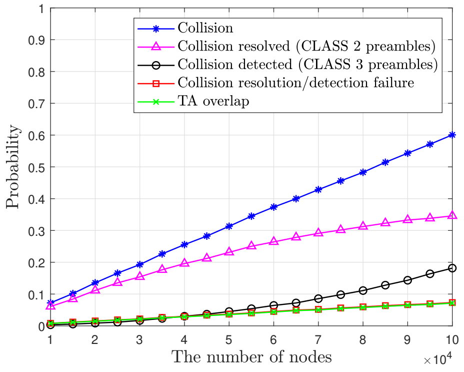


Fig. 14. Collision resolution and detection performance of the proposed CR model.

TABLE V

AVERAGE CPU RUNNING TIME FOR CLASSIFICATION TASKS [SEC]

	Proposed	Conventional
PA classifier	9.27×10^{-6}	1.71×10^{-6}
TA classifier	1.39×10^{-5}	2.13×10^{-6}

reattempts. In steady state, we can drop the slot index i, and then λ_{T} is expressed as $\lambda_{\mathrm{T}} = \frac{\lambda \cdot N \cdot T_{\mathrm{RACH}}}{1-\eta \cdot p_{\mathsf{c}}}$. As a result, the collision probability is expressed as $p_{\mathsf{c}} = 1 - (1-1/M)^{\lambda_{\mathrm{T}}}$ where M is the number of available preambles. Assuming $\eta \approx 1$, $p_{\mathsf{c}} = 1 - \exp\left\{W\left(\ln\left(1-\frac{1}{M}\right) \cdot \lambda \cdot N \cdot T_{\mathrm{RACH}}\right)\right\}$ where $W(x)$ denotes the Lambert W function [4]. However, in the heavy traffic, the derived collision probability does not work, and the collision probability goes unity. As a result, the probability of successful collision detection is calculated as $p_{\mathsf{c}}^{\mathsf{detect}} = p_{\mathsf{c}} \times A_{\mathsf{class2}}$, where A_{class2} denotes the preamble classification accuracy for $k = 2$ in the proposed CD model, and it is obtained as 0.968 in Table II (a). Fig.13 shows the collision detection performance of the proposed CD model. The analysis and the simulation results match well under $\eta \approx 1$, which means less than 50000 nodes in the simulation. As the number of nodes increases, the collision probability rapidly converges to 1. According to high preamble and TA classification accuracies of the proposed CD model, the most of preamble collisions are detected. However, we observe that a small number of collision detection failures would be still caused by TA overlap of two or more nodes, which is difficult to be solved in the proposed CD model.

Fig. 14 depicts the collision resolution and detection performance of the proposed CR model. As the number of nodes increases, the collision probability of the proposed CR model increases due to the proactive action to resolve collisions. According to high preamble and TA classification accuracies of the proposed CR model, the most of preamble collisions occurred by two nodes are resolved, and the most of preamble collisions occurred by three or more nodes are detected. Even though some of collision resolution and detection failures occur, it is not caused by preamble and TA classification problems, but the TA overlap problem of two or more nodes.

Finally, Table V compares the average CPU execution time of the proposed DL-based and conventional models. The testing procedures are implemented with a Matlab on a PC equipped with Intel i7 CPU 3.8 GHz processor and 32 GB RAM. The proposed DL approaches only require few microseconds, which is much shorter than a round-trip delay (RAR window size) of 5×10^{-3} seconds between the first and the second steps of the RA procedure [2], [40]. Although the DL-based preamble classifier needs more computational costs compared to the conventional RA model, the performance gain is meaningful at the expense of the increased time complexity.

VI. CONCLUSION

This paper proposed a DL-based non-orthogonal end-to-end RA framework where detection and resolution of preamble collisions are performed by DNNs. In the proposed RA framework, the preamble collision is first detected by a FNN preamble classifier, which estimates the number of the nodes using the identical preamble. For the preambles which are collided by multiple nodes, the TA classifier, which is realized by multiple CNNs, is applied to identify the TA value of each node. Based on the TA values, we can further resolve the preamble collision scenarios by assigning exclusive PUSCH resources or an identical PUSCH resource to collided nodes at the second step of the RA procedure. Numerical results verified the effectiveness of the proposed DL-based RA framework over baseline ML schemes such as the EBT and the SVM. We also provided the results of system level simulations to show the performance improvement obtained by the proposed DL-based CD and CR models. It was observed that almost of nodes succeed in the RA process by the proposed DL-based CR model while achieving the significantly reduced RA delay, even when there exist 10^5 nodes in a cell. This result can shed light on the viability of DL-based RA approaches for massive IoT scenarios. As a future research direction, the investigation on prototypes of the proposed DL solutions is worth pursuing.

REFERENCES

[1] C. Bockelmann, N. Pratas, H. Nikopour, K. Au, T. Svensson, C. Stefanovic, P. Popovski, and A. Dekorsy, "Massive machine-type communications in 5G: physical and MAC-layer solutions," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 59–65, Sept. 2016.

[2] S. Sesia, I. Toufik, and M. Baker, *LTE - The UMTS Long Term Evolution From Theory to Practice*. John Wiley & Sons Ltd., 2009.

© 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

[3] M.-Y. Cheng, G.-Y. Lin, H.-Y. Wei, and A.-C. Hsu, "Overload control for machine-type-communications in LTE-Advanced system," *IEEE Commun. Mag.*, vol. 50, no. 6, pp. 38–45, June 2012.

[4] H. S. Jang, S. M. Kim, K. S. Ko, J. Cha, and D. K. Sung, "Spatial group based random access for M2M communications," *IEEE Commun. Lett.*, vol. 18, no. 6, pp. 961–964, June 2014.

[5] T. P. de Andrade, C. A. Astudillo, and N. L. da Fonseca, "Allocation of control resources for machine-to-machine and human-to-human communications over LTE/LTE-A networks," *IEEE Internet Things J.*, vol. 3, no. 3, pp. 366–377, June 2016.

[6] H. Jin, W. Toor, B. C. Jung, and J. B. Seo, "Recursive Pseudo-Bayesian access class barring for M2M communications in LTE systems," *IEEE Trans. Veh. Technol.*, vol. 66, no. 9, pp. 8595–8599, Sept. 2017.

[7] N. Zhang, G. Kang, J. Wang, Y. Guo, and F. Labeau, "Resource allocation in a new random access for M2M communications," *IEEE Commun. Lett.*, vol. 19, no. 5, pp. 843–846, May 2015.

[8] H. S. Jang, S. M. Kim, H.-S. Park, and D. K. Sung, "An early preamble collision detection scheme for cellular M2M random access," *IEEE Trans. Veh. Technol.*, vol. 66, no. 7, pp. 5974–5984, July 2017.

[9] Y. Liang, X. Li, J. Zhang, and Z. Ding, "Non-orthogonal random access for 5G networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 7, pp. 4817–4831, July 2017.

[10] H. S. Jang, H.-S. Park, and D. K. Sung, "A non-orthogonal resource allocation scheme in spatial group based random access for cellular M2M communications," *IEEE Trans. Veh. Technol.*, vol. 66, no. 5, pp. 4496–5000, May 2017.

[11] J. Seo, B. C. Jung, and H. Jin, "Nonorthogonal random access for 5G mobile communication systems," *IEEE Trans. Veh. Technol.*, vol. 67, no. 8, pp. 7867–7871, Aug. 2018.

[12] J. Choi, "Layered non-orthogonal random access with SIC and transmit diversity for reliable transmissions," *IEEE Trans. Commun.*, vol. 66, no. 3, pp. 1262–1272, Mar. 2018.

[13] J. Seo, B. C. Jung, and H. Jin, "Performance analysis of NOMA random access," *IEEE Commun. Lett.*, vol. 22, no. 11, pp. 2242–2245, Nov. 2018.

[14] H. S. Jang, H. Lee, and T. Q. S. Quek, "Deep learning-based power control for non-orthogonal random access," *IEEE Commun. Lett.*, vol. 23, no. 11, pp. 2004–2007, Nov. 2019.

[15] H. S. Jang, S. M. Kim, H.-S. Park, and D. K. Sung, "A preamble collision resolution scheme via tagged preambles for cellular IoT/M2M communications," *IEEE Trans. Veh. Technol.*, vol. 67, no. 2, pp. 1825–1829, Feb. 2018.

[16] K. S. Ko, M. J. Kim, K. Y. Bae, D. K. Sung, J. H. Kim, and J. Y. Ahn, "A novel random access for fixed-location machine-to-machine communications in OFDMA based systems," *IEEE Commun. Lett.*, vol. 16, no. 9, pp. 1428–1431, Sept. 2012.

[17] L. Bai, J. Liu, Q. Yu, J. Choi, and W. Zhang, "A collision resolution protocol for random access in massive MIMO," *IEEE Journal on Selected Areas in Communications*, 2020 (to appear).

[18] W. Lee, D.-H. Cho, and M. Kim, "Deep power control: transmit power control scheme based on convolutional neural network," *IEEE Commun. Lett.*, vol. 22, no. 6, pp. 1276–1279, June 2018.

[19] W. Lee, M. Kim, and D.-H. Cho, "Deep learning based transmit power control in underlaid device-to-device communication," *IEEE Sys. J.*, vol. 13, no. 3, pp. 2551–2554, Sept. 2019.

[20] W. Lee, "Resource allocation for multi-channel underlay cognitive radio network based on deep neural network," *IEEE Commun. Lett.*, vol. 22, no. 9, pp. 1942–1945, Sept. 2018.

[21] W. Lee, M. Kim, and D.-H. Cho, "Transmit power control using deep neural network for underlay device-to-device communication," *IEEE Wireless Commun. Lett.*, vol. 8, no. 1, pp. 141–144, Feb. 2019.

[22] S. Peng, H. Jiang, H. Wang, H. Alwageed, Y. Zhou, M. M. Sebdani, and Y.-D. Yao, "Modulation classification based on signal constellation diagrams and deep learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 3, pp. 718–727, Mar. 2019.

[23] V. Ninkovic, D. Vukobratovic, A. Valka, and D. Dumic, "Preamble-based packet detection in Wi-Fi: A deep learning approach," *arXiv preprint:2009.05740*, 2020.

[24] J. Ding, D. Qu, P. Liu, and J. Choi, "Deep learning enabled preamble collision resolution in distributed massive mimo," *arXiv preprint:2006.04314*, 2020.

[25] H. Sun, A. O. Kaya, M. Macdonald, H. Viswanathan, and M. Hong, "Deep learning based preamble detection and toa estimation," in *Proc. IEEE GLOBECOM*. IEEE, 2019, pp. 1–6.

[26] M. H. Jespersen, M. Pajovic, T. Koike-Akino, Y. Wang, P. Popovski, and P. V. Orlik, "Deep learning for synchronization and channel estimation in NB-IoT random access channel," in *Proc. IEEE GLOBECOM*. IEEE, 2019, pp. 1–7.

[27] D. Chu, "Polyphase codes with good periodic correlation properties," *IEEE Trans. Inf. Theory*, vol. 18, no. 4, pp. 531–532, July 1972.

[28] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, May 2015.

[29] Z. Wang and V. W. Wong, "Optimal access class barring for stationary machine type communication devices with timing advance information," *IEEE Trans. Wireless Commun.*, vol. 14, no. 10, pp. 5374–5387, Oct. 2015.

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

[31] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Unsupervised representation learning of structured radio communication signals," *in Proc. IEEE Int. Workshop Sens. Process. Learn. Intell. Mach. (SPLINE)*, pp. 1–5, Aug. 2016.

[32] J. J. Rodriguez, L. I. Kuncheva, and C. J. Alonso, "Rotation forest: A new classifier ensemble method," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 10, pp. 1619–1630, Oct. 2006.

[33] S. Dörner *et al.*, "Deep learning-based communication over the air," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 132–143, Feb. 2018.

[34] *Base Station (BS) radio transmission and reception*, 3GPP TS 36.104 v15.4.0, Sept. 2018.

[35] T. G. Dietterich, "An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization," *Machine learning*, vol. 40, no. 2, pp. 139–157, 2000.

[36] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[37] *Study on RAN improvements for machine-type Communications*, 3GPP TR 37.868 V11.0.0, Sept. 2011.

[38] *Medium Access Control (MAC) protocol specification*, 3GPP TS 36.321 V15.3.0, Sept. 2018.

[39] Y.-J. Choi, S. Park, and S. Bahk, "Multichannel random access in OFDMA wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 3, pp. 603 – 613, Mar. 2006.

[40] *Study on RAN improvements for machine-type communications*, 3GPP TR 37.868 V11.0.0, Sept. 2011.

Han Seung Jang (S'14–M'18) received the B.S degree in Electronics and Computer Engineering from Chonnam National University, Gwangju, Korea, in 2012, and the M.S. and Ph.D. degrees in Electrical Engineering from Korea Advanced Institute for Science and Technology (KAIST), Daejeon, Korea in 2014 and 2017, respectively.

From May 2018 to February 2019, he was a Post-Doctoral Fellow with the Information Systems Technology and Design (ISTD) Pillar, Singapore University of Technology and Design (SUTD), Singapore. He was also a Post-Doctoral Fellow from September 2017 to April 2018 with Chungnam National University, Daejeon, Korea. He is currently an Assistant Professor with the School of Electrical, Electronic Communication, and Computer Engineering, Chonnam National University, Yeosu, Korea. His research interests include cellular Internet-of-things (IoT)/machine-to-machine (M2M) communications, machine learning, smart grid, and energy ICT.

Hoon Lee (S'14–M'18) received the B.S. and Ph.D. degrees in electrical engineering from Korea University, Seoul, Korea, in 2012 and 2017, respectively. In 2018, he was a Post-Doctoral Fellow with Singapore University of Technology and Design, Singapore. Since 2019, he has been with Pukyong National University, Busan, South Korea, where he is currently an Assistant Professor with the Department of Information and Communication Engineering. His research interests include optimization, machine learning, and signal processing for wireless networks.

Tony Q.S. Quek (S'98-M'08-SM'12-F'18) received the B.E. and M.E. degrees in electrical and electronics engineering from the Tokyo Institute of Technology in 1998 and 2000, respectively, and the Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology in 2008. Currently, he is the Cheng Tsang Man Chair Professor with Singapore University of Technology and Design (SUTD). He also serves as the Director of the Future Communications R&D Programme, the Head of ISTD Pillar, and the Deputy Director of the SUTD-ZJU IDEA. His current research topics include wireless communications and networking, network intelligence, internet-of-things, URLLC, and big data processing.

Dr. Quek has been actively involved in organizing and chairing sessions, and has served as a member of the Technical Program Committee as well as symposium chairs in a number of international conferences. He is currently serving as an Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS and an elected member of the IEEE Signal Processing Society SPCOM Technical Committee. He was an Executive Editorial Committee Member for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, an Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS, and an Editor for the IEEE WIRELESS COMMUNICATIONS LETTERS.

Dr. Quek was honored with the 2008 Philip Yeo Prize for Outstanding Achievement in Research, the 2012 IEEE William R. Bennett Prize, the 2015 SUTD Outstanding Education Awards – Excellence in Research, the 2016 IEEE Signal Processing Society Young Author Best Paper Award, the 2017 CTTC Early Achievement Award, the 2017 IEEE ComSoc AP Outstanding Paper Award, the 2020 IEEE Communications Society Young Author Best Paper Award, the 2020 IEEE Stephen O. Rice Prize, the 2020 Nokia Visiting Professor, and the 2016-2020 Clarivate Analytics Highly Cited Researcher. He is a Distinguished Lecturer of the IEEE Communications Society and a Fellow of IEEE.

Hyundong Shin (M'04-SM'11-F'21) received the B.S. degree in electronics engineering from Kyung Hee University (KHU), Yongin-si, Korea, in 1999, and the M.S. and Ph.D. degrees in electrical engineering from Seoul National University, Seoul, Korea, in 2001 and 2004, respectively. During his postdoctoral research at the Massachusetts Institute of Technology (MIT) from 2004 to 2006, he was with the Wireless Communication and Network Sciences Laboratory within the Laboratory for Information Decision Systems (LIDS).

In 2006, Dr. Shin joined the KHU, where he is now a Professor at the Department of Electronic Engineering and Department of Electronics and Information Convergence Engineering. His research interests include quantum information science, wireless communication, and machine intelligence.

Dr. Shin was honored with the Knowledge Creation Award in the field of Computer Science from Korean Ministry of Education, Science and Technology (2010). He received the IEEE Communications Society's Guglielmo Marconi Prize Paper Award (2008) and William R. Bennett Prize Paper Award (2012). He served as a Publicity co-chair for the IEEE PIMRC (2018) and a Technical Program co-chair for the IEEE WCNC (PHY Track 2009) and the IEEE Globecom (Communication Theory Symposium 2012, Cognitive Radio and Networks Symposium 2016). He was an Editor for IEEE Transactions on Wireless Communications (2007-2012) and IEEE Communications Letters (2013-2015).