# Aggregate Preamble Sequence Design and Detection for Massive IoT with Deep Learning

Ahmed Elhamy Mostafa, *Student Member, IEEE*, Vincent W.S. Wong, *Fellow, IEEE*, Yong Zhou, *Member, IEEE*, Robert Schober, *Fellow, IEEE*, Zhe Luo, Shuri Liao, and Mengying Ding

***Abstract*—Massive Internet of Things (mIoT) is a major use case of the fifth generation (5G) wireless systems. mIoT aims to support a large number of connection requests from IoT devices. However, the conventional Long Term Evolution (LTE) random access procedure hinders the support of mIoT due to the limited number of available preambles. In this paper, we propose to aggregate two Zadoff-Chu preamble sequences from two different roots to obtain a larger set of preambles by considering all possible combinations of preamble sequence pairs. Decoding the aggregate preambles is challenging because the receiver needs to decode two preamble sequences where each one is allocated half of the transmit power. We propose two receiver architectures for preamble decoding. The first one is a threshold-based receiver which only requires minor changes to the LTE preamble receiver architecture. The second proposed preamble decoder architecture exploits a deep neural network. Simulations show that the proposed aggregate preamble design results in a lower service time for backlogged IoT devices compared to existing collision avoidance techniques. Moreover, the proposed receiver architectures can decode the aggregate preambles with low probabilities of misdetection and false alarm (less than $11\%$), especially in the high signal-to-noise ratio (SNR) regime.**

***Index Terms*—Massive Internet of Things (mIoT), preamble sequence design, random access, deep neural networks.**

## I. Introduction

The number of connected Internet of Things (IoT) devices is expected to reach 14.7 billion by 2023 [2]. The fifth generation (5G) wireless technologies, such as New Radio [3], aim at supporting the massive IoT (mIoT) use case [4]. For mIoT, connectivity for a large number of low-cost low-power IoT devices has to be provided within a given area. IoT devices communicate with their servers and among each other with minimal human intervention [5]. High connection density and low energy consumption are the main requirements of the mIoT use case, where typical applications include smart home, smart cities, wearables, and environmental sensing. The number of simultaneous connections can reach one million per km$^2$ [6]. In addition, the battery lifetime of IoT devices is required to be at least $10$ years due to potential difficulties in battery replacement in some IoT applications.

Enhancing the random access design of 5G networks is a major challenge towards supporting mIoT [7]. An IoT device initiates the random access procedure to establish a cellular network connection by sending a randomly selected preamble sequence [8]. This procedure takes place before data packets are sent. In Long Term Evolution (LTE), only $64$ orthogonal preambles generated from Zadoff-Chu sequences are available as specified in [9]. Preamble collision occurs when multiple devices select the same preamble within a single random access opportunity. Given the limited number of preambles and the massive number of concurrent random access requests, there is a high probability of preamble collision. In case of preamble collision, the IoT devices have to backoff and attempt to retransmit the preambles later. When preamble collisions occur frequently, the random access procedure becomes inefficient as the IoT devices have to wait for a longer time to start data transmission, especially when multiple preamble retransmissions are required. In addition, retransmissions cause the power consumption of the battery-powered IoT devices to increase. Thus, it is desirable to improve the random access design of 5G networks to meet the requirements of mIoT.

Several random access schemes have been proposed to alleviate the random access contention. Access class barring (ACB) assigns a higher access probability to high priority devices in order to reduce the random access contention [10]. This is achieved by assigning an ACB factor to each quality of service (QoS) class. Then, each IoT device sends a preamble with a probability that is equal to the ACB factor associated with its QoS class. This reduces the number of contending devices in the random access procedure during each random access opportunity. In [11], the random access contention is reduced by optimizing the ACB factor based on the knowledge of the number of backlogged IoT devices. In [12], an algorithm is proposed to adjust the ACB factor so as to maximize the number of successfully served IoT devices using timing advance information. Primary and secondary ACB factors are used in [13] to reduce the random access delay by allowing the secondary devices to bypass the preamble transmission step in the random access procedure based on information

Manuscript received on Apr. 5, 2020; revised on Oct. 26, 2020; accepted on Feb. 21, 2021. This paper has been published in part in the *Proceedings of IEEE Global Communications Conference (Globecom)*, Abu Dhabi, UAE, Dec. 2018 [1]. The editor coordinating the review of this paper and approving it for publication was Sunwoo Kim.

A. E. Mostafa and V. W.S. Wong are with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC, V6T 1Z4, Canada (e-mail: {ahmedem, vincentw}@ece.ubc.ca).

Y. Zhou is with the School of Information Science and Technology, ShanghaiTech University, Shanghai, China (e-mail: zhouyong@shanghaitech.edu.cn).

R. Schober is with the Institute for Digital Communications, Friedrich-Alexander-University, Erlangen-Nuremberg, Erlangen 91058, Germany (e-mail: robert.schober@fau.de).

Zhe Luo, Shuri Liao, and Mengying Ding are with Huawei Technologies, Co. Ltd., Shanghai, China (e-mail: {luozhe2, liaoshuri, dingmengying}@huawei.com).

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

from the base station regarding the preambles occupied by those primary devices which transmitted preambles earlier. The schemes in [11]–[13] require prior knowledge of the number of backlogged IoT devices, which may not be available in all scenarios. To address this issue, pseudo-Bayesian ACB is proposed in [14], where the ACB factor depends on the estimated number of IoT devices given the number of unused preambles. However, ACB requires the IoT devices to have different priorities, which may not be the case in mIoT. Another random access contention alleviation approach is to divide the IoT devices into clusters based on their geographical locations. Each cluster has a cluster head that is responsible for transmitting one preamble for all devices in its cluster [15]. Clustering reduces preamble transmissions but entails the additional complexities of cluster formation and cluster head selection. The authors in [16] propose a scheme that combines transmit power ramping (i.e., increasing the transmit power for each failed random access attempt) and backoff, and is shown to outperform ACB in heavy traffic scenarios. However, power ramping and multiple random access attempts may not be energy efficient for mIoT devices.

The authors in [17] propose a non-orthogonal random access (NORA) scheme which can detect the colliding preambles of multiple IoT devices based on the differences in their time of arrival. In particular, NORA resolves the preamble collision problem by allowing power-domain multiplexing of access request messages from multiple IoT devices. On the other hand, code-domain multiplexing is employed in [18]–[20], where sparse code multiple access (SCMA) and repeated preamble transmissions over multiple time slots are used to resolve preamble collisions. Other works, such as [21]–[23], use the concept of grant-free multiple access, which eliminates the random access procedure and allows devices to send data directly without waiting for the granting of the radio resources by the base station. For example, successive decoding is employed in [21] to decode superimposed messages in collision time slots by using the decoded messages from singleton time slots (i.e., slots without collisions). In addition, some unlicensed band technologies for enabling IoT, such as SigFox and long-range wide access network (LoRaWAN), use unslotted Aloha-like data transmission [24]. However, grant-free transmission is more suitable for ultra-reliable and low latency communications (URLLC) applications with stringent deadlines. On the other hand, the majority of mIoT applications are delay-tolerant, and hence, grant-based transmission may be better suited.

Furthermore, another random access enhancement option is to modify the preamble sequence design so that a larger set of preambles is available to facilitate random access and accommodate the expected large number of IoT devices. The preambles generated from Zadoff-Chu sequences are cyclic shifted versions of a basic sequence referred to as the root sequence [9]. A naive approach to generate more preambles is to reduce the cyclic shift value, but this may cause ambiguity at the receiver when the distance between the device and the base station is large. In general, a smaller cyclic shift leads to a smaller coverage area for the base station, which makes it an inefficient solution for macrocells. Hence, to obtain a larger set of preambles, the design of the Zadoff-Chu-based random access preambles has to be modified. In [25], multiple preamble transmissions are performed, where each IoT device can either transmit the same preamble multiple times or transmit a subset of different preambles consecutively. However, this scheme introduces extra overhead to the physical random access channel (PRACH). In [26], the addition of Zadoff-Chu sequences to cover sequences is proposed in order to increase the number of preambles and reduce random access collisions. The concept of virtual preambles is introduced in [27], where an IoT device transmits a PRACH preamble over a specific narrowband channel to reduce the probability of preamble collision.

To enable efficient random access for mIoT, new preamble sequence designs, which can provide a larger number of preamble sequences, are a promising approach. Although some existing techniques, such as ACB and backoff, can mitigate random access contention and collisions, they incur a long random access delay. Consequently, the IoT devices need to stay in the active mode for a longer time, which results in a higher energy consumption. On the other hand, savings in data transmission latency with grant-free multiple access may not be essential for delay-tolerant mIoT applications. For seamless integration with 5G standards, it is desirable that new preamble sequence designs can utilize similar building blocks at the transmitter side as the conventional Zadoff-Chu preamble sequences in LTE. In [28], an extended set of preambles is generated via element-by-element multiplication of Zadoff-Chu sequences and cover sequences. An enhanced preamble detection technique is developed in [29], where additional iterations are used to exclude preambles that are detected incorrectly due to channel impairments (e.g., frequency offset). In [30], a preamble detection technique is proposed for detection of non-orthogonal preambles in order to enable the use of a larger set of preambles in each cell.

In our previous work [1], we propose a preamble design where two preamble Zadoff-Chu sequences are aggregated to generate up to $\binom{64}{2} = 2{,}016$ preambles instead of the $64$ preambles for one Zadoff-Chu root sequence. This scheme enlarges the set of preamble sequences. It only requires minor changes to the conventional PRACH transmitter and receiver, which facilitate the integration of the proposed scheme in the 5G standard. However, the design in [1] imposes several challenges. As the IoT devices divide their transmit power over two preambles, the probability of preamble misdetection increases, especially when the received signal-to-noise ratio (SNR) is low. In addition, reducing the received power threshold for preamble detection in the low SNR regime results in a higher probability of false alarm (i.e., a higher probability of detecting preambles that have not been actually transmitted). In addition, the false alarm probability is expected to increase further as the number of preamble transmissions by the IoT devices increases.

In this paper, we propose to aggregate two preamble Zadoff-Chu sequences with two different roots to generate up to $64^2 = 4{,}096$ preambles. Similar to [1], the proposed aggregate preamble design reduces the probability of preamble collision and enables 5G systems to support the mIoT use case. The new preamble sequences can be generated and decoded by applying minor changes to the conventional LTE PRACH transmitter and receiver, respectively. We also propose to select a subset of the aggregate preambles with low peak-to-average power ratio (PAPR) to reduce the energy consumption of the IoT devices. Moreover, generating the aggregate preambles using Zadoff-Chu sequences with two different roots enables preamble detection with a low probability of false alarm (less than $11\%$). We enhance the aggregate preamble detection performance by proposing a novel receiver architecture based on training a deep neural network (DNN) [31]. DNN is a powerful supervised machine learning tool that can learn the input-output relation of a system to predict the output for a given input in an autonomous manner [32]. Recently, machine learning techniques, and DNNs in particular, have been used to solve various problems in communication systems, including interference management [33], power control for interfering links [34], user clustering in millimeter wave non-orthogonal multiple access networks [35], joint transmitter and receiver design optimization [36], [37], and data sequence detection in optical and molecular communications [38]. In this paper, we utilize the correlation vector obtained from the first stage of the PRACH receiver, which represents the correlation between the received signal and the Zadoff-Chu preamble sequences, as the input of a DNN for aggregate preamble detection. We generate labelled datasets via simulations to train the DNNs (i.e., obtain the DNN parameters). Each sample in the dataset has a vector of correlation values as input features and a vector of binary labels that indicate the correct aggregate preambles to be detected for this sample. The same approach can also be used for training DNNs to detect the conventional single preambles. Moreover, the resulting DNNs are found to correctly detect the aggregate preambles with a high probability in the high SNR regime. The main contributions of this paper can be summarized as follows:

- We propose a new preamble sequence design for mIoT by aggregating two different Zadoff-Chu sequence preambles generated from two different roots to increase the number of available random access preambles.
- We present a threshold-based receiver architecture, which only requires minor changes to the conventional PRACH receiver.
- We propose a novel PRACH receiver architecture by training a DNN, where the decoding of the aggregate Zadoff-Chu preambles is a multi-label classification problem. We train the DNN (i.e., obtain the DNN parameters) by generating labelled datasets via simulations.
- Simulation results show that the proposed preamble design reduces the probability of preamble collision to less than $10^{-4}$, which is lower than that of the conventional LTE PRACH. The proposed preamble design also reduces the average total service time compared to ACB. In addition, both the threshold-based and the DNN-based receiver architectures can decode the aggregate preambles with low probabilities of misdetection and false alarm, especially when the SNR is high.

The remainder of this paper is organized as follows. In Section II, we present an overview of the random access procedure, the new preamble sequence design and the corresponding transmitter structure, and the threshold-based receiver structure. In Section III, we propose a novel PRACH receiver architecture for preamble detection which is based on a set of DNNs. In Section IV, we first analyze the outage probability and verify it by simulations. Then, based on simulations, we evaluate the PAPR, the probability of preamble collision, the total service time, and the probability of preamble detection at the receiver. Section V concludes the paper.

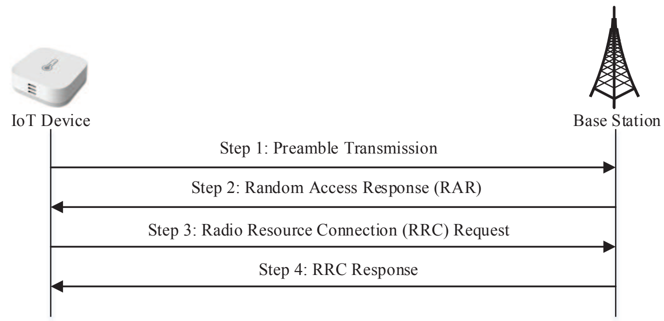


Fig. 1. Contention-based random access procedure in LTE systems [8].

## II. New Preamble Sequence Design

In this section, we first review the random access procedure in LTE. We then present the proposed new preamble sequence design as well as the transmitter and receiver structures that generate and decode the proposed preamble sequences, respectively.

### A. Random Access Procedure in LTE

In LTE, each IoT device is required to establish a connection with the base station before transmitting data packets by initiating a contention-based random access procedure, as shown in Fig. 1 [8]. In particular, the IoT device first sends a randomly selected preamble during a random access opportunity (Step 1) and waits for a random access response (RAR) message from the base station (Step 2). After receiving the RAR message, the IoT device extracts the control information (e.g., the preamble identifier, the timing advance information, and an initial uplink resource grant) and transmits a radio resource connection (RRC) request (Step 3). Finally, the IoT device waits for an uplink radio resource grant for data transmission from the base station (Step 4). There are $64$ orthogonal preambles generated from the Zadoff-Chu sequences as specified in [9]. Due to the orthogonality, different preambles transmitted by multiple IoT devices can be decoded by the base station at the same time. However, a collision occurs when two or more IoT devices select the same preamble and transmit simultaneously (i.e., in the same random access opportunity), which is very probable in mIoT scenarios.

In LTE, orthogonal preamble sequences of length $N_{zc}$ are generated using Zadoff-Chu sequences for a given cyclic shift $N_{cs}$ and root index $l$ that is selected from a set of available roots $\mathcal{L} = \{1, \ldots, N_{zc} - 1\}$ [9]. Given a set of preamble indices $\mathcal{M} = \{1, \ldots, M\}$, each preamble has a unique index $m \in \mathcal{M}$. We denote the preamble with index $m$ that is generated using root index $l \in \mathcal{L}$ as $s_m^l$. The first preamble $s_1^l$ is the root sequence. The other preambles are generated by cyclically shifting the root sequence by multiples of $N_{cs}$. Hence, the number of orthogonal preambles per root $M$ is equal to $\lfloor N_{zc}/N_{cs} \rfloor$, where $\lfloor \cdot \rfloor$ denotes the floor function. Taking the LTE preamble formats $0-3$ [9] as an example, when $N_{zc}$ is equal to 839 and $N_{cs}$ is equal to 13, there are 64 preambles per root. Note that the preambles can take different formats that differ in the length of the preamble sequence and in the length of cyclic prefix. For channels with deep fading, longer sequences and longer cyclic prefixes (e.g., format 3) can be used to enhance the reliability of preamble transmission and combat channel fading.

### *B. Preamble Aggregation*

We propose to enlarge the set of preambles by adding two preamble sequences, e.g., $s_a^l$ and $s_b^{l'}$, that have the same length $N_{zc}$ for given root indices $l,\ l' \in \mathcal{L}$, where $a,\ b \in \mathcal{M}$, and $l \neq l'$. A one-to-one mapping function $g$ is used for selecting $l'$ for given $a$, i.e., $l' = g(a)$. This means that each possible value of the preamble index $a$ corresponds to a unique root index $l'$ to generate the second preamble $s_b^{l'}$. Here, $l$ is referred to as the *primary root* and all possible values of $l'$ are referred to as *secondary roots*. The rationale behind this design is to facilitate the association of pairs of preambles in a given aggregate preamble. For example, when preamble $s_b^{l'}$ is detected, it can be easily associated with $s_a^l$ since $l' = g(a)$ and $l$ is fixed for a given base station. Further details are provided in Section II.D. An example mapping function is given by $l' = (l + Aa)$, where $A$ is a positive integer that satisfies inequality $AM \leq N_{zc} - 1$. The resulting aggregate preamble, denoted by $q_{a,b}^{l,l'}$, is given by

$$q_{a,b}^{l,l'}[n] = \alpha_a s_a^l[n] + \alpha_b s_b^{l'}[n],\ \forall\, l, l' \in \mathcal{L},\ l' = g(a),$$
$$a, b \in \mathcal{M},\ 0 \leq n < N_{zc}, \quad (1)$$

where $n$ is the discrete time index. $\alpha_a$ and $\alpha_b$ denote the power scaling coefficients of preambles $s_a^l$ and $s_b^{l'}$, respectively, such that $\alpha_a^2 + \alpha_b^2 = 1$. Hence, the number of preambles generated can be increased from $M = 64$ to $M^2 = 64^2 = 4{,}096$. With a larger number of preambles, the probability of preamble collision in the random access procedure can be reduced significantly. Moreover, the proposed aggregate preamble design can easily be implemented and be considered for adoption in future standards as only minimal upgrades of the conventional LTE transmitter and receiver are required. Specifically, the transmitter only needs to average two preamble sequences, while the receiver only needs to handle the detection of two preamble sequences to decode the aggregate preamble. Throughout this paper, we refer to the conventional Zadoff-Chu preamble sequences (e.g., $s_a^l$) as *single preambles* and to the preamble sequences resulting from aggregating two Zadoff-Chu preamble sequences (e.g., $q_{a,b}^{l,l'}$) as *aggregate preambles*.

### *C. Transmitter Architecture*

The transmitted signal is generated as follows. First, two preamble sequences $s_a^l$ and $s_b^{l'}$ are randomly selected and

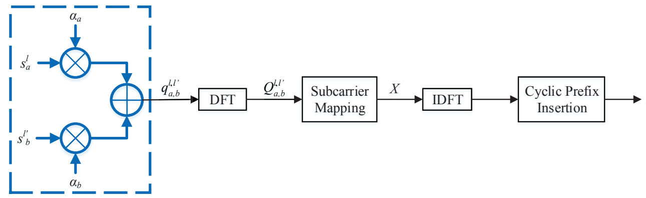



Fig. 2. Illustration of the transmitter architecture that generates the aggregate preambles. Preambles $s_a^l$ and $s_b^{l'}$ are scaled by $\alpha_a$ and $\alpha_b$, respectively, and aggregated. The resulting preamble sequence $q_{a,b}^{l,l'}$ is processed by the stages of DFT, subcarrier mapping, IDFT, and cyclic prefix insertion.

aggregated with power scaling coefficients $\alpha_a$ and $\alpha_b$, respectively, to obtain $q_{a,b}^{l,l'}$, as in (1). Then, the aggregate preamble $q_{a,b}^{l,l'}$ is input to the processing of a conventional LTE PRACH transmitter, as shown in Fig. 2. The required changes with respect to the conventional PRACH transmitter are confined to the blue dashed box. Next, a discrete Fourier transform (DFT) of size $N_{zc}$ is applied to the aggregate preamble $q_{a,b}^{l,l'}$ to obtain the frequency domain representation $Q_{a,b}^{l,l'}$, which is given by

$$Q_{a,b}^{l,l'}[k] = \sum_{n=0}^{N_{zc}-1} q_{a,b}^{l,l'}[n] \exp\left(\frac{-j2\pi nk}{N_{zc}}\right),\ 0 \leq k < N_{zc}, \quad (2)$$

where $k$ is the index in the discrete frequency domain. Then, after subcarrier mapping, we obtain the transmitted signal in the frequency domain, denoted by $X$ in Fig. 2. This step is followed by an inverse discrete Fourier transform (IDFT) of size $(1/(\Delta f_{\text{RA}} T_s))$, where $\Delta f_{\text{RA}}$ is the preamble subcarrier spacing and $T_s$ is the LTE basic time unit. In LTE, $\Delta f_{\text{RA}} = 1.25$ kHz and $1/T_s = 30.72$ MHz for preamble formats $0-3$, which results in an IDFT size of 24,576 [9]. After adding the cyclic prefix, the transmitted signal in the time domain $x[t]$ is given by

$$x[t] = \beta_{\text{PRACH}} \sum_{k=0}^{N_{zc}-1} Q_{a,b}^{l}[k]$$
$$\times \exp\left(-j2\pi(k + \varphi + K(k_0 + 0.5))\Delta f_{\text{RA}}(t - T_{\text{CP}})\right),$$
$$0 \leq t < T_{\text{CP}} + T_{\text{SEQ}}, \quad (3)$$

where $\beta_{\text{PRACH}}$ is the PRACH amplitude scaling factor [9]. $T_{\text{CP}}$ and $T_{\text{SEQ}}$ are the cyclic prefix length and sequence length, respectively. $\varphi$, $K$, and $k_0$ are constant PRACH parameters that are used to map the transmitted signal to the proper subcarriers based on the preamble format as specified in [9]. The received signal $y[t]$ at the base station, assuming a frequency-selective fading channel with impulse response $h[j]$ with $J+1$ taps, can be expressed as

$$y[t] = \sum_{j=0}^{J} h[j]x[t-j] + z[t], \quad 0 \leq t < T_{\text{CP}} + T_{\text{SEQ}}, \quad (4)$$

where $z[t]$ is the complex additive white Gaussian noise (AWGN).

### *D. Threshold-based Receiver Architecture*

Given the received signal $y$ at the base station, the estimates of the transmitted preamble pairs of the different devices

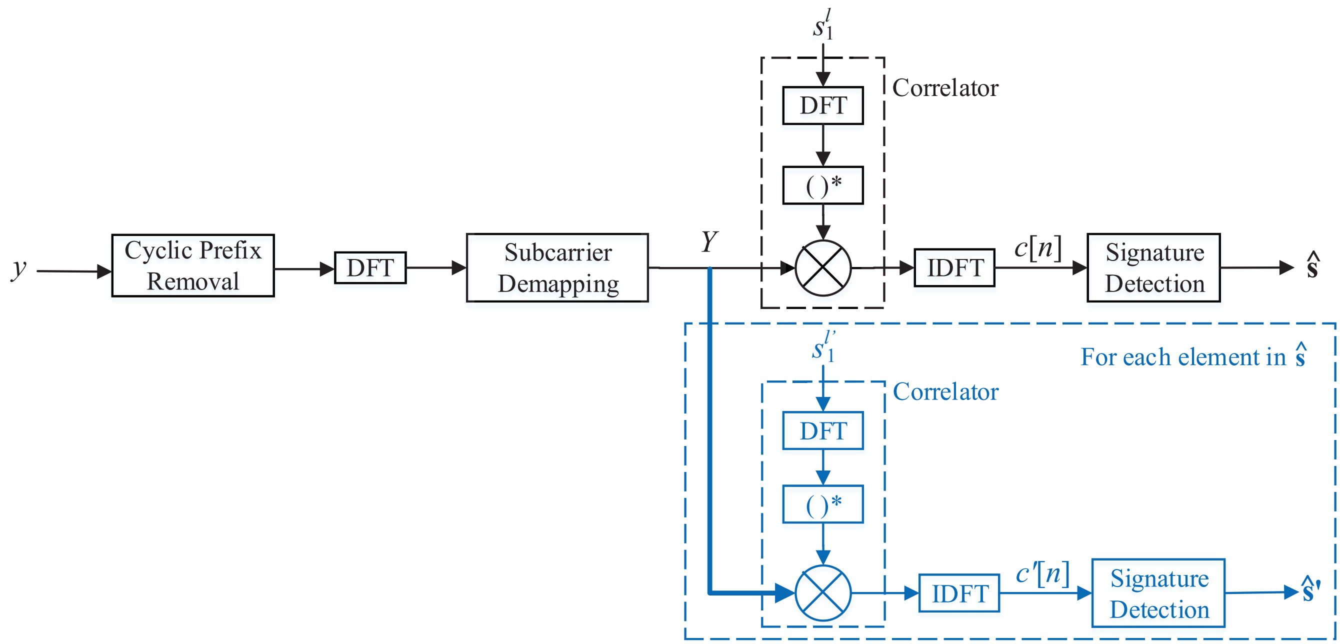



Fig. 3. Illustration of the threshold-based receiver architecture. The received signal $y$ is correlated with the primary root sequence $s_1^l$ to obtain the time correlation $c[n]$ that is used to estimate the transmitted preambles using the primary root $l$, i.e., set $\hat{\mathbf{s}}$. For each preamble $\hat{s}_m^l$ in $\hat{\mathbf{s}}$, the received signal $y$ is correlated with the secondary root sequence $s_1^{l'}$ ($l' = g(m)$) to obtain the preamble(s) paired with $\hat{s}_m^l$, i.e., $\hat{\mathbf{s}}'$.

can be obtained by directly correlating the received signal with all possible preamble pairs $q_{a,b}^{l,l'}$, $a, b \in \mathcal{M}$. However, this requires the calculation of the correlation with all $4{,}096$ possible preamble pairs, which entails a high complexity.

To circumvent this, we propose an alternate receiver architecture that requires only one additional stage compared to the conventional PRACH receiver, which is contained in the blue dashed box in Fig. 3. This additional stage correlates the received signal with the root sequence of the candidate secondary roots to identify the paired preamble. In particular, the received signal $y$ is processed by the stages of cyclic prefix removal, DFT, and subcarrier demapping. The frequency domain representation of the received signal, denoted as $Y[k]$, is multiplied by the complex conjugate of the frequency domain representation of the root sequence of the primary root $l$, denoted as $S_1^l$. The resulting time correlation $c[n]$ can be expressed as

$$c[n] = \frac{1}{N_{\text{IDFT}}} \sum_{k=0}^{N_{\text{IDFT}}-1} Y[k](S_1^l[k])^* \exp\left(\frac{j2\pi kn}{N_{\text{IDFT}}}\right), \quad 0 \le n < N_{\text{IDFT}}, \tag{5}$$

where $N_{\text{IDFT}}$ is the IDFT size, and $(\cdot)^*$ denotes complex conjugation. Since the other preamble sequences $s_2^l, \ldots, s_M^l$ are cyclically shifted versions of $s_1^l$, $c[n]$ represents the correlation with all preambles $s_m^l$, $m \in \mathcal{M}$. The range of $n$, i.e., $[0, N_{\text{IDFT}})$, is divided into $M$ partitions. Each partition corresponds to one preamble sequence $s_m^l$. Let $\mathbf{n}_m$ denote the set of consecutive values of $n$ in the partition associated with preamble $s_m^l$. For example, $\mathbf{n}_1 = \{n^{\text{st}}, \ldots, n^{\text{end}}\}$, $0 \le n^{\text{st}}, n^{\text{end}} < N_{\text{IDFT}}$, denotes the set of the values of $n$ in the partition associated with root sequence $s_1^l$. Then, $\mathbf{n}_2$ can be defined as $\{(n^{\text{end}}+1) \mod N_{\text{IDFT}}, \ldots, (n^{\text{end}} + \lfloor N_{\text{IDFT}}/M \rfloor) \mod N_{\text{IDFT}}\}$, and so on for all $m \in \mathcal{M}$. The maximum value of $c[n]$ in a given partition $m$, i.e., $\max_{n\in\mathbf{n}_m} c[n]$, is the correlation between the received signal and the corresponding preamble $s_m^l$. Hence, having an impulse at index $n$ that exceeds a certain threshold indicates that the corresponding preamble was transmitted. The signature detection stage in Fig. 3 determines the set of estimated transmitted preambles $\hat{\mathbf{s}}$ based on the following rule

$$\hat{\mathbf{s}} = \{\hat{s}_m^l \mid \max_{n\in\mathbf{n}_m} c[n] > C, \ m \in \mathcal{M}\}, \tag{6}$$

where $C$ is a threshold value that is chosen to meet a certain detection requirement. Note that we cannot employ equalization since the channel state information (CSI) is not available at the base station because the preamble transmission is the first communication between the IoT device and the base station before any data transmission or channel estimation takes place.

In Fig. 4, examples of correlation vectors $c[n]$ are shown for the case of two devices sending two aggregate preambles with primary root index $l = 129$ to the base station. In these examples, we focus on the upper branch of the receiver in Fig. 3, where we only detect preambles using the primary root. In Fig. 4 (a), the received preambles $\hat{\mathbf{s}} = \{\hat{s}_{53}^{129}, \hat{s}_{39}^{129}\}$ can easily be identified in the absence of delay, channel noise, and fading since $c[n]$ has impulses at values of $n$ that correspond to these two preambles. Fig. 4 (b) shows a case where false alarms may occur in the presence of AWGN and the extended typical urban (ETU) fading channel [39]. When the detection threshold $C$ is set to $0.06$, we have $\hat{\mathbf{s}} = \{\hat{s}_{53}^{129}, \hat{s}_{39}^{129}\}$. However, when $C = 0.04$, we have $\hat{\mathbf{s}} = \{\hat{s}_{53}^{129}, \hat{s}_{39}^{129}, \hat{s}_{a}^{129}, \hat{s}_{b}^{129}\}$, where $s_a^{129}$ and $s_b^{129}$ are detected although they were not transmitted by any device. This can be avoided by tuning the threshold $C$ to meet certain detection or false alarm criteria.

Furthermore, for each estimated preamble $\hat{s}_m^l$ using the

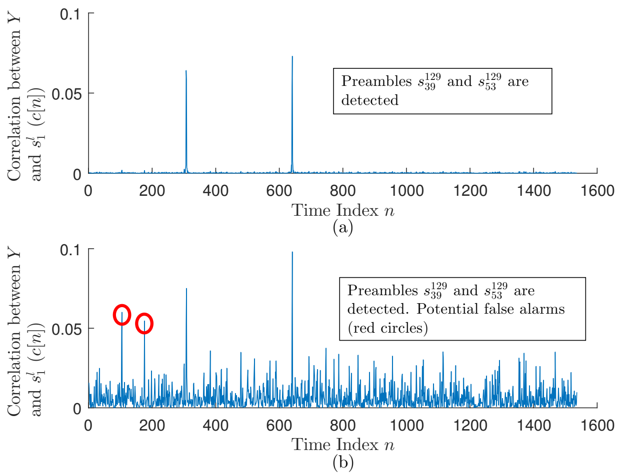



Fig. 4. Examples of correlation vectors $c[n]$. (a) In the absence of delay, noise, and channel fading, the transmitted preambles are detected successfully without any false alarms. (b) With AWGN and for the ETU fading channel, the receiver may suffer from false alarms (red circles).

root sequence of the primary root, we obtain the correlation between $Y$ and the frequency domain representation of the root sequence of the secondary root $l' = g(m)$, i.e., $S_1^{g(m)} = S_1^{l'}$, which can be expressed as

$$c'[n] = \frac{1}{N_{\text{IDFT}}} \sum_{k=0}^{N_{\text{IDFT}}-1} Y[k](S_1^{g(m)}[k])^* \exp\left(\frac{j2\pi kn}{N_{\text{IDFT}}}\right), \quad \hat{s}_m^l \in \hat{\mathbf{s}},\ 0 \le n < N_{\text{IDFT}}. \tag{7}$$

Similarly, the preamble transmitted using the secondary root is detected according to a correlation threshold $C'$

$$\hat{\mathbf{s}}' = \{\hat{s}'^{l'}_{m'} \mid \max_{n \in \mathbf{n}_{m'}} c'[n] > C',\ m' \in \mathcal{M}\}. \tag{8}$$

Hence, given an estimated preamble $s_m^l$ using the root sequence of the primary root $s_1^l$ and an estimated preamble $s_{m'}^{l'}$ using the root sequence of the secondary root $s_1^{l'}$, $l' = g(m)$, we estimate that the aggregate preamble $q_{m,m'}^{l,l'}$ is received by the base station. We define $\mathbf{q}_e$ as the set of all received aggregate preambles. The algorithm for detecting the aggregate preamble sequences is given in Algorithm 1. In Steps $3-4$, the preambles generated using the primary root are estimated. In Steps $5-11$, the set of received aggregate preambles are estimated by determining the preambles transmitted using the secondary roots that correspond to the estimated preambles in Steps $3-4$.

The proposed transmitter and receiver architectures require only minor changes to the conventional PRACH transmitter and receiver. This makes the proposed preamble sequence design a promising candidate to support mIoT without significant changes to the 5G standard.

## III. DNN-based PRACH Receiver Architecture

In this section, we present the proposed DNN-based PRACH receiver architecture, where we replace the signature detection stage in Fig. 3 with a DNN-based detection module, as shown in Fig. 5. The DNN-based detection stage takes the correlation vector $c[n]$ (or $c'[n]$) as input to determine the sets

**Algorithm 1:** Aggregate Preamble Sequence Detection Algorithm

1 **Input**: $Y$, $C$, $C'$, $s_1^l$
2 $\mathbf{q}_e := \emptyset$
3 Evaluate $c[n]$ according to (5)
4 Determine the set of estimated preambles $\hat{\mathbf{s}}$, which are generated using the primary root $s_1^l$ according to (6)
5 **for** $s_m^l \in \hat{\mathbf{s}}$ **do**
6 Evaluate $c'[n]$ according to (7)
7 Determine the set of estimated preambles $\hat{\mathbf{s}}'$, which are generated using the secondary root $s_1^{l'}$, where $l' = g(m)$, according to (8)
8 **for** $s_{m'}^{l'} \in \hat{\mathbf{s}}'$ **do**
9 $\mathbf{q}_e := \mathbf{q}_e \cup \{q_{m,m'}^{l,l'}\}$
10 **end**
11 **end**
12 **Output**: $\mathbf{q}_e$

of estimated preambles $\hat{\mathbf{s}}$ and $\hat{\mathbf{s}}'$. Note that the DNN-based detection stage can also be used for detecting the conventional single preambles by suitably modifying the DNN parameters.

### A. DNN-based Detection

*1) Problem Modelling:* The detection of the received preambles at the base station can be modelled as a multi-label classification problem [40] since multiple classes (i.e., preambles) can be detected in a single received PRACH signal. Multi-label classification problems can be solved by using one-vs-all classification [40], where one classifier is trained for each class (i.e., preamble) to decide whether this class is present in a given sample (i.e., received PRACH signal at the base station) or not. We need one classifier per single preamble since multiple classes can be detected simultaneously as multiple preambles can be detected within a single received signal instance (i.e., random access opportunity). Hence, the DNN-based detection stage consists of a number of DNN classifiers as shown in Fig. 5. Each classifier takes the values in the correlation vector $c[n], 0 \le n < N_{\text{IDFT}}$ (or $c'[n]$) as input features and generates a binary output $\theta_m^l \in \{0, 1\}$, $m \in \mathcal{M}$, for a given root index $l \in \mathcal{L}$. The output $\theta_m^l$ is set to 1 if the aggregate preamble $s_m^l$ is detected, and is set to 0 otherwise. The DNN classifier that detects $s_m^l$ can be employed to detect the same preamble generated using other root sequences as well (i.e., for $s_m^{l'}$, where $l' \neq l$) with the same weights since the root index does not impact the form of the correlation vector $c[n]$ (or $c'[n]$). Hence, for the proposed DNN-based receiver architecture, we need to train $M$ DNN classifiers (e.g., $M = 64$).

An advantage of the proposed preamble sequence design is that we only need $M$ DNN classifiers although there are $M^2$ aggregate preambles in total. In addition, training $M^2$ one-vs-all classifiers is expected to result in a very high false alarm rate even if each DNN module produce a very small number of false alarms as the false alarms will accumulate from $M^2$ modules. Similar to the threshold-based receiver in Section II.D, if preamble $s_m^l$ is detected, then we look for the paired preamble by correlating the frequency domain representation of the received signal $Y$ with the root sequence $s_1^{l'}$, $l' = g(m)$. We follow the same steps as in Algorithm 1 but we replace

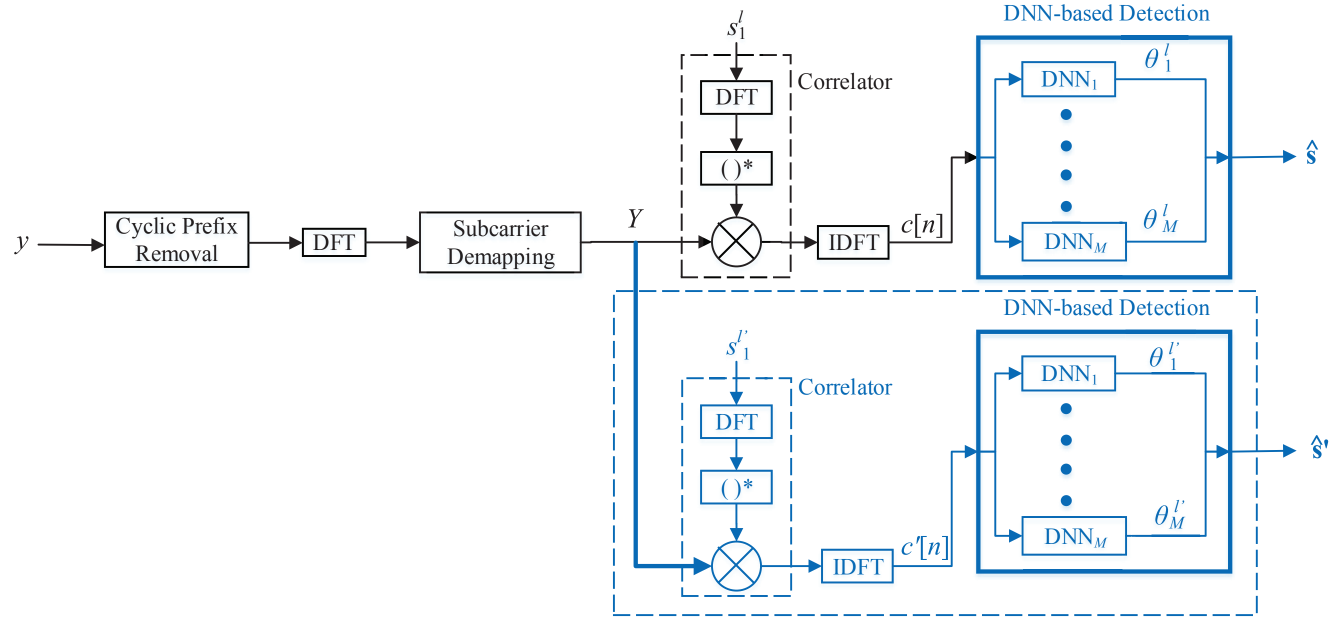



Fig. 5. The proposed PRACH receiver architecture with DNN-based preamble detection.

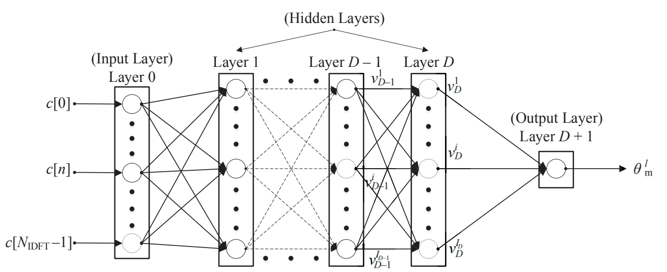



Fig. 6. The architecture of the DNN employed for the detection of preamble $s_m^l, m \in \mathcal{M}, l \in \mathcal{L}$.

the signature detection stage with the DNN-based detection stage (i.e., a group of DNN classifiers) to determine $\hat{\mathbf{s}}$ and $\hat{\mathbf{s}}'$ in Steps 4 and 7, respectively.

*2) DNN Architecture:* Each DNN classifier consists of an input layer, $D$ hidden layers, and an output layer as shown in Fig. 6. The input layer takes the input features, i.e., the correlation vector $c[n]$, $0 \leq n < N_{\text{IDFT}}$ (or $c'[n]$), and forwards them to the first hidden layer in the DNN. Each hidden layer $d$, $1 \leq d \leq D$, consists of $I_d$ neurons and takes the output of the preceding layer $d-1$ as input. We denote the input layer as layer 0 and the output layer as layer $D+1$. The output $v_d^i$ of neuron $i$, $1 \leq i \leq I_d$, in hidden layer $d$ is the value of the activation function $\phi_d^i(\cdot)$ obtained for the weighted sum of the outputs of the preceding layer $v_{d-1}^{i'}$, $1 \leq i' \leq I_{d-1}$,

$$v_d^i = \phi_d^i \left( \sum_{i'=1}^{I_{d-1}} w_d^{i,i'} v_{d-1}^{i'} + w_d^{i,0} \right), \quad 1 \leq i \leq I_d, \; 1 \leq d \leq D, \tag{9}$$

where $w_d^{i,i'}$ is the weight associated with $v_{d-1}^{i'}$ at neuron $i$ of hidden layer $d$ and $w_d^{i,0}$ is a bias variable. The output layer $D+1$ consists of a single neuron that estimates $\theta_m^l$ based on the output of the last hidden layer $D$ as follows:

$$\theta_m^l = \phi_{D+1}^1 \left( \sum_{i'=1}^{I_D} w_{D+1}^{1,i'} v_D^{i'} + w_{D+1}^{1,0} \right), \quad l \in \mathcal{L}, \; m \in \mathcal{M}, \tag{10}$$

where $\theta_m^l \in \{0, 1\}$. Note that the preamble and root indices are removed from the subscripts and superscripts of the DNN variables for ease of notation. In our DNN, we use the rectified linear unit (ReLU) function as activation function for all neurons in the hidden layers and a sigmoid function as activation function for the output layer neuron. For an arbitrary variable $\nu$, we have

$$\phi_d^i(\nu) = \begin{cases} \max(0, \nu), & 1 \leq d \leq D, \; 1 \leq i \leq I_d, \\ \frac{1}{1+\exp(-\nu)}, & d = D+1, i = 1. \end{cases} \tag{11}$$

*3) Loss Function:* To construct a DNN that predicts the output from the input with the minimum number of classification errors, the DNN has to be trained to determine the weights for the neurons in all layers. To train the DNNs, we use training sets where the output $\theta_m^l$ is known for a given sample of correlation vector input $c[n]$. In the beginning of the training phase, the weights are randomly initialized and then iteratively updated using the backpropagation algorithm [41]. In the iterations of the backpropagation algorithm, the RMSprop optimizer [42] is used for optimizing the weights in each iteration such that the loss defined by the binary cross-entropy is minimized:

$$\text{Loss}_{\text{Binary Cross-entropy}} = -\theta_m^l \log(\hat{\theta}_m^l) - (1 - \theta_m^l) \log(1 - \hat{\theta}_m^l), \quad l \in \mathcal{L}, \; m \in \mathcal{M}, \tag{12}$$

where $\hat{\theta}_m^l$ is the estimate of $\theta_m^l$ for given DNN weights.

*4) DNN Architecture Choice Rationale:* Determining the appropriate DNN hyperparameters (e.g., the number of hidden layers, the number of neurons per layer) requires evaluating the performance of the DNN after training with a validation set that is not used for training or testing, and the selection of the values of the hyperparameters that minimize the validation loss or error [32, Ch. 1]. However, in this preamble detection problem, we can give a higher priority to other metrics when selecting the DNN hyperparameters. In the DNN-based receiver, we need to achieve a balance between two different objectives, which are minimizing the probability of false alarm and minimizing the probability of misdetection. Hence, for example, we may choose a DNN architecture (or DNN weights) that may yield a higher loss or error but maintains the probability of false alarm below a certain threshold.

### *B. Training Procedure*

In this subsection, we describe in details the process of generating the training datasets that are used to train the modules of the DNN-based receiver. To generate the training samples, we first generate the PRACH signals for the IoT devices with SNR values chosen from a specific SNR range. In particular, the PRACH signals are generated according to (3) after randomly selecting a preamble for each IoT device. After propagating through independent channels, all the PRACH signals are added at the receiver along with AWGN. Finally, the received signal is passed through the first five stages of the DNN-based receiver (i.e., until the IDFT module in Fig. 5) to obtain the correlation input vector(s) $c[n]$. The output labels $\{\theta^l_m\}$ are stored according to the preambles selected by the IoT devices.

*1) Generating Training Samples (Transmitter Side):* We generate the training samples by mimicking realistic scenarios, where multiple IoT devices, with independent channel conditions, transmit random access preambles to the base station. In particular, the training samples are generated using simulations where the number of IoT devices is randomly chosen between 0 and $N^{\max}_{\text{Dev}}$. This enables training the DNN-based receiver to detect multiple preambles at the same time. Each device randomly selects an aggregate preamble for a given root index $l \in \mathcal{L}$. The transmitted signal of each device is generated according to (3) using the MATLAB function "ltePRACH" and propagates through an independent ETU fading channel with a Doppler frequency of 70 Hz as described in [39]. The received SNR of each signal is randomly determined according to a uniform distribution with maximum and minimum SNR values.

*2) Generating Training Samples (Receiver Side):* The signals received from all devices are added along with AWGN, where each received signal has its own SNR. We assume that the base station is equipped with four antennas so that the correlations resulting from the four versions of the received signal can be added to increase the detection capability. Then, the received signal is input to the first stages of the receiver in Fig. 5 (until the IDFT module) to obtain the correlation vector $c[n]$ using the MATLAB function "ltePRACHDetect". $c[n]$ represents the input features of the training samples. The output labels are obtained by setting $\theta^l_m$ to 1 if preamble $s^l_m$ has been selected by any of the devices for preamble transmission. Similarly, we obtain $c'[n]$ after correlation with the secondary root sequences and save the corresponding labels $\theta^{l'}_m$, $l' = g(m)$. For example, if two IoT devices transmit the aggregate preambles $q^{l,l'}_{a_1,b_1}$ and $q^{l,l''}_{a_2,b_2}$, we save three training entries in our training set. The first entry contains the correlation vector $c[n]$ obtained from correlating the received signal with $s^l_1$ as a vector of features and the labels $\theta^l_{a_1}$ and $\theta^l_{a_2}$ are set to 1. The second entry contains the correlation vector $c'[n]$ obtained from correlating the received signal with $s^{l'}_1$, $l' = g(a_1)$, as a vector of features and the label $\theta^{l'}_{b_1}$ is set to 1. The third entry contains the correlation vector $c''[n]$ obtained from correlating the received signal with $s^{l''}_1$, $l'' = g(a_2)$, as a vector of features and the label $\theta^{l''}_{b_2}$ is set to 1.

*3) Training Set per DNN Module:* In order to train a DNN that detects a specific preamble $s^l_a$, we need to generate a training dataset with a sufficient number of training samples that include this specific preamble. Hence, we generate a specific dataset where at least one device transmits this specific preamble (i.e., $\theta^l_a = 1$) with a probability of 0.5 in each training sample. This setting enables us to have a more balanced training set and the DNN receiver does not become biased towards not detecting preambles during training since they are only present in a few samples if a uniform probability distribution is considered.

*4) Training SNR Selection:* Training the DNN-based receiver using samples within a certain SNR range still allows it to operate over a wider SNR range. In particular, a DNN-based receiver can still operate at SNR values that are greater than the maximum training SNR as preamble detection becomes easier. Moreover, the receiver can also operate for SNR values that are lower than the minimum training SNR but with degraded performance. However, training the DNN-based receiver using very low SNR values is not advisable since it causes the receiver to lose its ability to generalize in the high SNR regime [37] as the receiver tends to detect preambles that are not present more often (i.e., higher numbers of false alarms) due to confusing low correlations resulting from low SNR preambles with low correlations resulting from noise. For training the DNN classifiers to detect the preambles that form the proposed aggregate preambles, we set the maximum SNR value to 0 dB and the minimum SNR value to 0, $-3$, or $-6$ dB.

### *C. Testing Procedure*

We consider different types of validation and testing sets that are described in the following:

*1) Validation Set:* This set is similar to the training set but it is not used to train the DNN parameters (i.e., weights). However, it is used to decide the DNN parameters with minimum classification errors (or minimum number of false alarms) for a given a given preamble to avoid overfitting the training data. We evaluate multiple DNN models for each preamble and choose the model that achieves the lowest number of false alarms, i.e., the model with the lowest number of detection errors caused by the detection of a preamble that has not been transmitted by any device. If multiple models achieve the same performance in terms of the false alarms, we select the model with the lowest number of misdetections, i.e., the model with the lowest number of detection errors caused by not detecting a preamble that has been transmitted by a device.

*2) Fixed SNR Test Sets:* Each set consists of test samples that are generated in a similar manner as the training samples. However, all preambles are selected by the devices with equal probability, and all devices have the same SNR value. For a given test set, there are multiple devices with the same fixed SNR. Multiple test sets are used to evaluate the performance of the DNN-based PRACH receiver at different SNR values. This allows us to plot the probability of misdetection or the probability of false alarm versus SNR.

## IV. Performance Evaluation

In this section, we first analyze the outage probability of a single device for the transmission of a single or an aggregate preamble, respectively. Then, based on simulations, we evaluate the PAPR, the probability of preamble collision, and the total service time for a group of backlogged IoT devices when using the proposed preamble sequence design. We also evaluate the performance of the proposed receiver architectures in terms of the probabilities of misdetection and false alarm. Throughout this section, equal power allocation is assumed, i.e., $\alpha_a = \alpha_b = \frac{1}{\sqrt{2}}$. First, equal power allocation ensures that each of the two preambles forming the aggregate preamble is detected with equal probability at the receiver without prior knowledge of the preamble selection at the transmitter. Second, the IoT devices cannot optimize the power allocation since they do not have CSI before performing the random access procedure.

### *A. Analysis of Outage Probability*

We perform an outage probability analysis in order to illustrate the impact of aggregating two preambles on the probability of preamble misdetection. Consider a typical IoT device that transmits a random access preamble to the base station with transmit power $P$ over a Rayleigh fading channel with gain $h$ and propagation loss $\Upsilon$. If a single preamble is transmitted, the entire transmit power is allocated to that preamble. The received SNR at the base station is given by $\gamma = \frac{P\Upsilon|h|^2}{\sigma^2} = \overline{\gamma}|h|^2$, where $\sigma^2$ is the noise power and $\overline{\gamma} = \frac{P\Upsilon}{\sigma^2}$ is the average SNR of the typical device. To detect a single preamble, the received SNR $\gamma$ should be greater than a certain threshold $\overline{\gamma}_{\mathrm{s}}$ [16]. Hence, the outage probability in case of transmitting a single preamble can be expressed as

$$P_{\mathrm{out}}^{\mathrm{s}} = \Pr[\overline{\gamma}|h|^2 \leq \overline{\gamma}_{\mathrm{s}}] = \Pr\left[|h|^2 \leq \frac{\overline{\gamma}_{\mathrm{s}}}{\overline{\gamma}}\right] = 1 - \exp\left(\frac{-\overline{\gamma}_{\mathrm{s}}}{\overline{\gamma}}\right). \tag{13}$$

On the other hand, if the aggregate preamble $q_{a,b}^{l,l'}$ is transmitted, the transmit power $P$ is divided between the two aggregated preambles, where transmit powers $\alpha_a^2 P$ and $\alpha_b^2 P$ are allocated to preambles $s_a^l$ and $s_b^{l'}$, respectively. Throughout this analysis, we assume equal power allocation, i.e., $\alpha_a^2 = \alpha_b^2 = \frac{1}{2}$. To detect an aggregate preamble, the received SNR of each preamble should be greater than a certain threshold $\overline{\gamma}_{\mathrm{agg}}$, i.e., $\alpha_a^2\overline{\gamma}|h|^2 \geq \overline{\gamma}_{\mathrm{agg}}$ and $\alpha_b^2\overline{\gamma}|h|^2 \geq \overline{\gamma}_{\mathrm{agg}}$. The correlation between $s_a^l$ and $s_b^{l'}$ is small ($1/\sqrt{N_{zc}}$) and neglected. Hence,

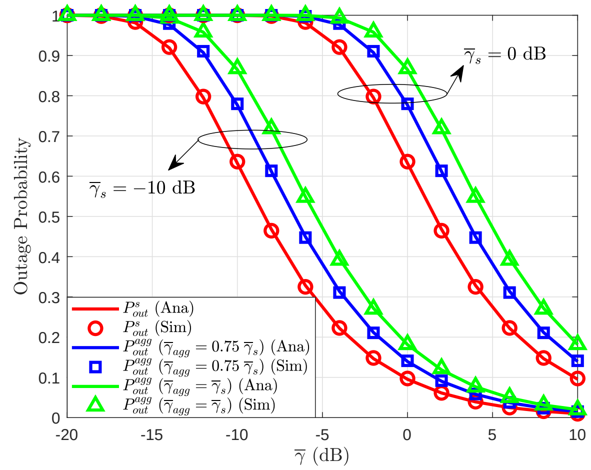



Fig. 7. Average outage probability for the single and aggregate preambles versus the average SNR of a typical device $\overline{\gamma}$ at different minimum SNR detection threshold $\overline{\gamma}_{\mathrm{s}}$.

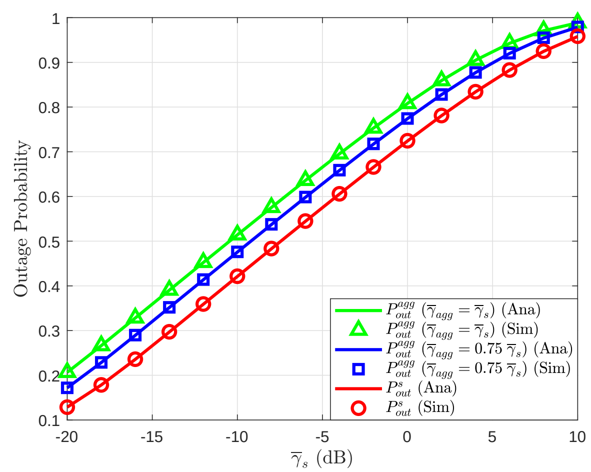



Fig. 8. Average outage probability for the single and aggregate preambles versus different values of the minimum SNR detection threshold $\overline{\gamma}_{\mathrm{s}}$.

if an aggregate preamble is transmitted, the outage probability is given by

$$\begin{aligned} P_{\mathrm{out}}^{\mathrm{agg}} &= \Pr[\min\{\alpha_a^2\overline{\gamma}|h|^2, \alpha_b^2\overline{\gamma}|h|^2\} \leq \overline{\gamma}_{\mathrm{agg}}] \\ &= \Pr[\min\{0.5\overline{\gamma}|h|^2, 0.5\overline{\gamma}|h|^2\} \leq \overline{\gamma}_{\mathrm{agg}}] \\ &= \Pr\left[|h|^2 \leq \frac{2\overline{\gamma}_{\mathrm{agg}}}{\overline{\gamma}}\right] = 1 - \exp\left(\frac{-2\overline{\gamma}_{\mathrm{agg}}}{\overline{\gamma}}\right). \end{aligned} \tag{14}$$

Note that choosing $\overline{\gamma}_{\mathrm{agg}}$ to be equal to $\overline{\gamma}_{\mathrm{s}}$ makes the base station less capable of detecting the aggregate preambles since the received power from each component of the preamble pair is less than that of the single preamble. To make the outage probability for the aggregate preambles equal to that for the conventional single preambles, we have to set the SNR threshold as $\overline{\gamma}_{\mathrm{agg}} = \frac{1}{2}\overline{\gamma}_{\mathrm{s}}$. However, this would increase the probability of false alarm, i.e., the detection of a preamble that is not actually transmitted by any device.

We consider an average SNR $\overline{\gamma}$ with a range from $-20$ dB to 10 dB for the IoT device. We evaluate the outage probabilities for single and aggregate preamble detection for different values of the detection thresholds $\overline{\gamma}_{\mathrm{s}}$ and $\overline{\gamma}_{\mathrm{agg}}$, respectively. As shown in Fig. 7, the outage probability in case of aggregate preamble transmission becomes closer to that of the single preamble transmission for smaller $\overline{\gamma}_{\mathrm{agg}}$. Moreover, Fig. 8

shows that the outage probability for aggregate preamble transmission increases as the detection threshold value increases, which makes detecting aggregate preambles more difficult at the base station. In both figures, the simulation (Sim) results match well with the analytical (Ana) results.

### *B. PAPR Performance*

We use the PAPR of the transmitted PRACH signals as a metric to evaluate the energy consumption of the IoT devices. Since most IoT devices are battery-powered, transmitting a signal with a high PAPR implies more energy consumption [43]. The PAPR of the transmitted signal $x[t]$, denoted as $\tau_{a,b}^{\alpha_a,\alpha_b}$, is defined as the ratio between its maximum power and its average power, given that the preamble pair $q_{a,b}^{l,l'}$ is selected and power scaling coefficients $\alpha_a$ and $\alpha_b$ are used. Hence, we have

$$\tau_{a,b}^{\alpha_a,\alpha_b} = (T_{\text{CP}}+T_{\text{SEQ}}) \frac{\max\{|x[0]|^2, \ldots, |x[T_{\text{CP}}+T_{\text{SEQ}}-1]|^2\}}{\sum_{t=0}^{T_{\text{CP}}+T_{\text{SEQ}}-1} |x[t]|^2}, \tag{15}$$

where $T_{\text{CP}} + T_{\text{SEQ}}$ is the length of the transmitted signal $x[t]$.

Our goal is to find a set of aggregate preambles for each primary root index $l \in \mathcal{L}$, denoted as $\Lambda^l$, that includes only the aggregate preambles whose PAPR is less than a threshold value $\tau_{\max} + \beta$. We define $\tau_{\max}$ as the maximum PAPR of all preambles $s_m^l$ for all $m \in \mathcal{M}$ and $l \in \mathcal{L}$, and $\beta$ is the tolerance threshold. Then, $\Lambda^l$ can be defined as

$$\Lambda^l = \{q_{a,b}^{l,l'} \mid l' = g(a), 10\log_{10}(\tau_{a,b}^{\alpha_a,\alpha_b}) \leq \tau_{\max}+\beta\}, \ \forall\, l \in \mathcal{L}. \tag{16}$$

When $\beta$ is equal to 0 dB, we select only the aggregate preambles whose PAPR does not exceed the maximum PAPR of conventional LTE preambles. Hence, the number of random access preambles available for the proposed aggregate preamble sequence design is reduced, which results in a higher probability of preamble collisions. On the other hand, setting $\beta > 0$ dB enables $\Lambda^l$ to include the aggregate preambles whose PAPR is less than $\tau_{\max}+\beta$, which leads to the inclusion of more aggregate preambles in set $\Lambda^l$ compared to the case of $\beta = 0$ dB. If $\beta$ is set to $\infty$, then all 4,096 aggregate preambles are included without any PAPR constraint. Hence, $\beta$ can be set to balance the PAPR and the probability of preamble collisions.

In Fig. 9, we show the cumulative distribution function (CDF) of the PAPR of the proposed aggregate preamble sequence design for preamble format 0 [9]. Here, $\tau_{\max}$ is equal to 7.5 dB, and we vary the PAPR tolerance threshold $\beta$ which affects the number of aggregate preambles that satisfy the PAPR threshold criterion. As $\beta$ decreases, $\Lambda^l$ contains fewer aggregate preambles that can be used but the maximum and median PAPR decrease which reduces the energy consumption. The average number of available aggregate preambles at primary root $l$ is equal to $\textbf{card}(\Lambda^l)$, where $\textbf{card}(\cdot)$ denotes the cardinality of a set. For example, if $\beta$ is equal to 0 dB and 0.2 dB, the average number of available aggregate preambles is 1,992 and 3,600, respectively, rather than 4,096 when $\beta \geq 0.6$ dB. In Fig. 10, the ratio between the average number of available preambles per root and the number of single preambles $M$ (where $M = 64$) is shown for different values

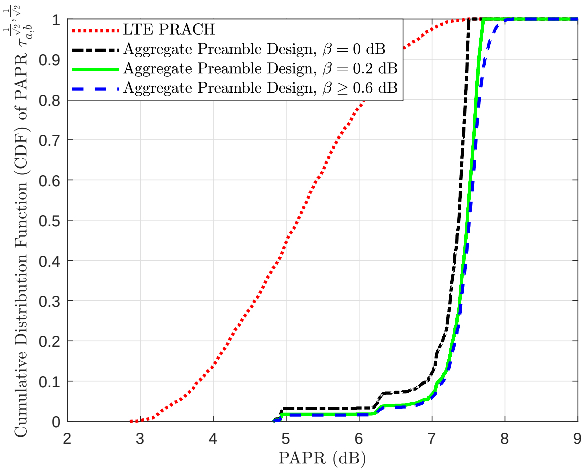



Fig. 9. CDF of the PAPR for different values of tolerance threshold $\beta$. As $\beta$ decreases, we select aggregate preambles with lower PAPRs which enhances the PAPR performance but reduces the number of available aggregate preambles.

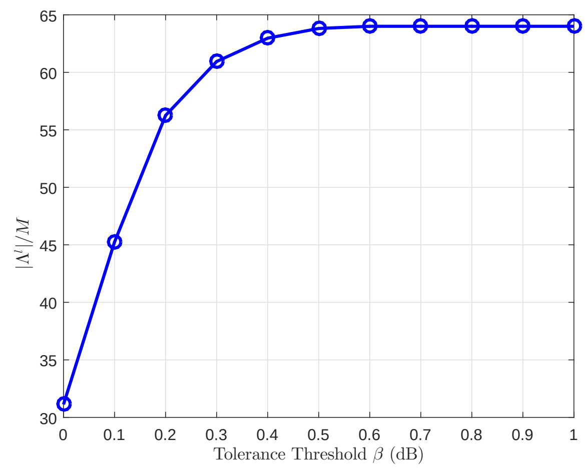



Fig. 10. Ratio between the average number of available preambles per root $\textbf{card}(\Lambda^l)$ and the number of single preambles $M$ versus different values of tolerance threshold $\beta$.

of tolerance threshold $\beta$. This ratio shown on the vertical axis represents the number of times by which the number of available preambles increases when using the aggregate preambles. This result shows that the number of available preambles increases by up to a factor of $31.3 - 64$ depending on the affordable PAPR, when the proposed preamble design is used.

### *C. Random Access Performance*

We evaluate the random access performance based on three metrics, namely the probability of preamble collision, the average total service time, and the probability of random access success. The probability of preamble collision is the ratio between the number of preamble collision occurrences and the total number of preamble transmissions. This indicates the capability of the proposed aggregate preamble sequence design to support a larger number of devices by increasing the number of preambles from 64 to 4,096. We evaluate the probability of preamble collision in a simulation setup where the number of devices $N_{\text{Dev}}$ varies from 1,000 to 30,000 devices. These devices follow a uniform arrival process over a period of 10 sec. Preamble retransmissions are allowed for

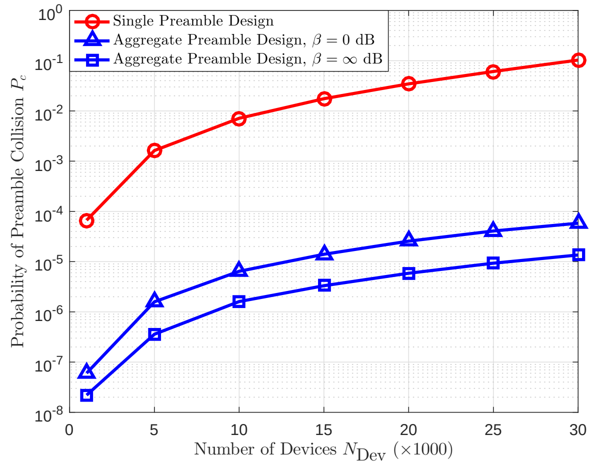



Fig. 11. Probability of preamble collision versus the number of devices $N_{\text{Dev}}$ for different values of PAPR tolerance threshold $\beta$ with uniform arrival process and retransmissions.

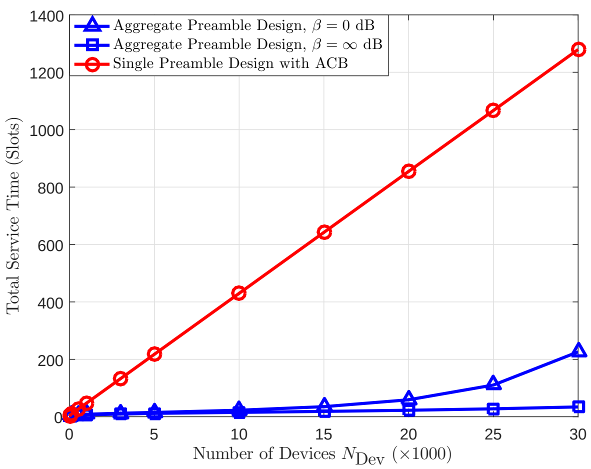



Fig. 12. Average total service time versus the number of devices $N_{\text{Dev}}$ for different values of PAPR tolerance threshold $\beta$.

up to 10 times. The backoff, RAR message, and contention window delays are taken into account and set to be 20 ms, 5 ms, and 48 ms, respectively [44]. Each device has a random access opportunity to start its random access procedure every 5 ms. In this simulation, the probability of preamble detection in case of no collision is modelled according to [44] and is equal to $(1 - 1/\exp(\text{transmission attempt index}))$. For the proposed aggregate preamble sequence design, we consider $\beta = 0$ dB (i.e., a subset of only 1,992 preambles is considered to maintain a PAPR less that $\tau_{\max} = 7.5$ dB) and $\beta = \infty$ dB (i.e., all 4,096 preambles are considered). We compare the proposed preamble sequence design with conventional LTE PRACH (i.e., 64 preambles). Fig. 11 shows the impact of the number of IoT devices on the probability of preamble collision. It can be observed that the proposed preamble sequence design can be used to support mIoT with a probability of preamble collision that is less than $10^{-4}$. Furthermore, setting a PAPR threshold (i.e., using fewer preambles per root) reduces the PAPR without incurring a significant increase in the collision probability.

The second considered performance metric is the average total service time, which is defined as the time needed until all the backlogged IoT devices successfully transmit

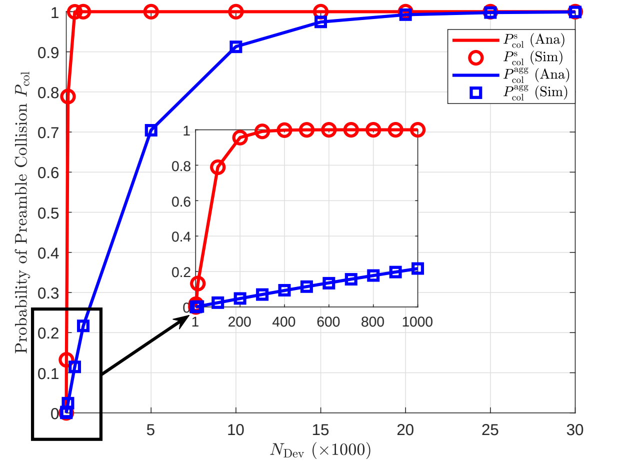



Fig. 13. Probability of preamble collision in a single random access opportunity versus the number of devices $N_{\text{Dev}}$ for the single and aggregate preamble cases.

a preamble without encountering a collision. In the first random access opportunity slot, there are 1,000 to 30,000 devices that transmit randomly selected preambles. If a device encounters a preamble collision, it will backoff randomly for 1 to 4 random access opportunity slots (each random access opportunity slot is 5 ms long). We compare the performance of the proposed aggregate preamble sequence design with $\beta = 0$ dB and $\beta = \infty$ dB and the ACB scheme [11] in terms of the average total service time. In ACB, there are 64 preambles and the ACB probability of transmission is optimized, i.e., $p_{\text{ACB}} = \min\{1, \text{Number of preambles}/\text{Number of backlogged devices}\}$ [11]. It is assumed that the base station knows the total number of backlogged devices in the system. Fig. 12 shows the average total service time as a function of the number of IoT devices. The results show that the proposed aggregate preamble sequence design outperforms the ACB scheme and reduces the total service time by up to 97%. Note that, without ACB, having only 64 preambles may not be sufficient to serve devices with a finite delay constraint due to the very large number of preamble collisions.

To evaluate the probability of random access success, we consider a network of $N_{\text{Dev}}$ IoT devices that transmit preambles during the same random access opportunity. The probability of random access success for a given IoT device is defined as the probability that this device can successfully transmit a preamble without collision, i.e., the preamble is successfully decoded by the base station and the same preamble is not selected by any other device. We assume that the RAR message in the second step of the random access procedure is always received successfully. The probability of preamble collision for a typical device in the single and aggregate preamble cases, denoted as $P_{\text{col}}^{\text{s}}$ and $P_{\text{col}}^{\text{agg}}$, respectively, can be expressed as follows:

$$P_{\text{col}}^{\text{s}} = 1 - \left(\frac{M-1}{M}\right)^{N_{\text{Dev}}-1}, \tag{17}$$

$$P_{\text{col}}^{\text{agg}} = 1 - \left(\frac{M^2-1}{M^2}\right)^{N_{\text{Dev}}-1}. \tag{18}$$

Fig. 13 shows the probability of preamble collision for the

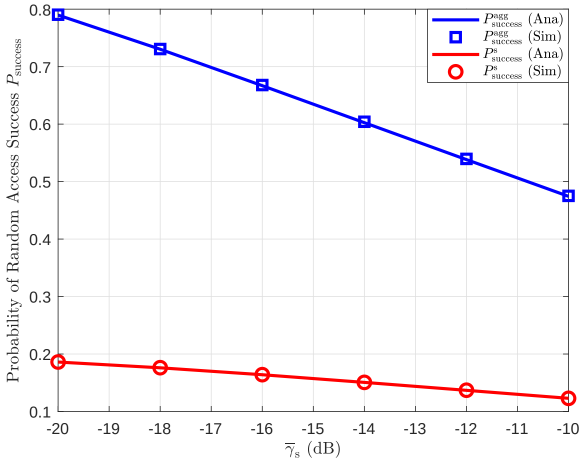



Fig. 14. Probability of random access success versus different values of the minimum SNR detection threshold $\overline{\gamma}_{\text{s}}$ when $N_{\text{Dev}} = 100$.

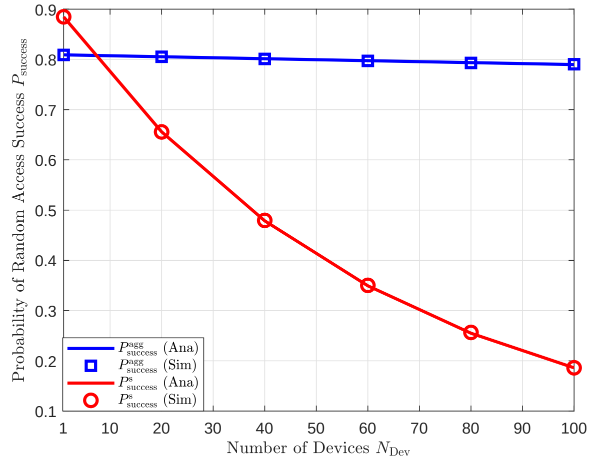



Fig. 15. Probability of random access success versus the number of devices $N_{\text{Dev}}$ when $\overline{\gamma}_{\text{s}} = -20$ dB.

single and aggregate preamble cases for different values of $N_{\text{Dev}}$.

Then, we can obtain the probability of random access success for the typical device using the outage probability expression obtained in Section IV.A as follows:

$$P^{\text{s}}_{\text{success}} = (1 - P^{\text{s}}_{\text{col}})(1 - P^{\text{s}}_{\text{out}}), \tag{19}$$

$$P^{\text{agg}}_{\text{success}} = (1 - P^{\text{agg}}_{\text{col}})(1 - P^{\text{agg}}_{\text{out}}). \tag{20}$$

Equations (19) and (20) show the tradeoff between outages and preamble collisions. The outage probability for the single preamble case is lower than that for the aggregate preamble case. However, the proposed aggregate preamble design helps reduce the collision probability due to the increase in the number of available preambles. To obtain numerical and simulation results, $\overline{\gamma}_{\text{agg}}$ is set to be equal to $\overline{\gamma}_{\text{s}}$. As shown in Fig. 14, when the number of devices $N_{\text{Dev}}$ is equal to 100, aggregate preambles become more essential for the success of the random access process despite their higher outage probability. This is due to the significant increase in the collision probability for single preambles. Furthermore, Fig. 15 reveals that, when the number of devices increases, the probability of random access success in the single preamble case decreases at a faster rate compared to the aggregate preamble case for a given SNR detection threshold $\overline{\gamma}_{\text{s}}$.

### *D. Preamble Detection Performance*

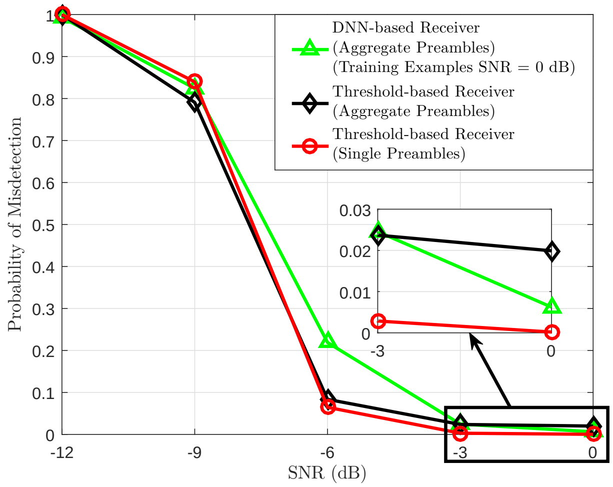
In this subsection, we evaluate the preamble detection performance of both the threshold-based receiver (shown in Fig. 3) and the DNN-based receiver (shown in Fig. 5). We consider the case where each IoT device randomly selects a preamble pair to transmit. We consider up to $N^{\max}_{\text{Dev}} = 10$ devices per sample. All aggregate preambles are chosen with equal probability. The preambles are transmitted over an ETU fading channel with a Doppler frequency of 70 Hz [39]. A uniform random delay of up to ten samples is applied to the transmitted preamble signal. We consider a sampling rate of 1.92 MHz and a frequency offset of 270 Hz when generating the training and testing samples [39]. Preamble sequences are generated based on preamble format 0 [9]. Hence, $N_{zc}$ is set to 839 and $N_{cs}$ is set to 13 generating $M = 64$ Zadoff-Chu single preamble sequences, which are aggregated to generate the $4,096$ aggregate preambles. We vary the received SNR from $-12$ dB to $0$ dB. The proposed receiver architecture is evaluated in terms of the probabilities of misdetection and false alarm. The probability of misdetection is the ratio between the number of undetected transmitted preambles and the total number of transmitted preambles, i.e., $\frac{\text{Number of False Negatives}}{\text{Number of True Positives}+\text{Number of False Negatives}}$. The probability of false alarm is the ratio between the number of detected preambles that were not actually transmitted (e.g., incorrect detection decisions) and the total number of transmitted preambles, which is given by $\frac{\text{Number of False Positives}}{\text{Number of Testing Samples}}$.

TABLE I
LAYOUT OF THE DNN CLASSIFIERS OF THE AGGREGATE PREAMBLE PRACH RECEIVER.

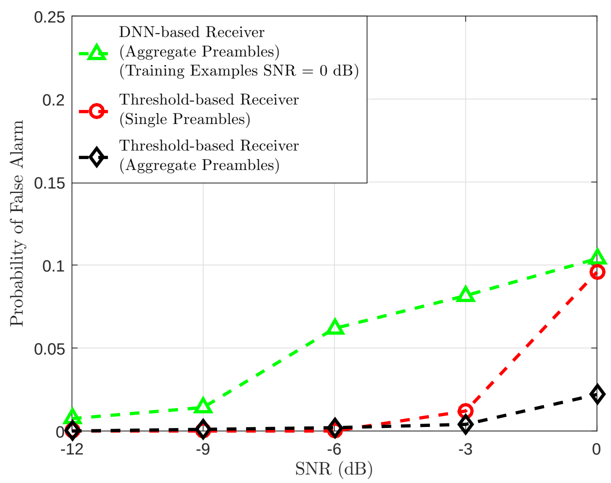
| Layer | Activation Function | Output Dimensions |
|---|---|---|
| Input Layer (Layer 0) | $--$ | $N_{\text{IDFT}}$ |
| Hidden Layers (Layers $1-6$) | ReLU | $512, 256, 128, 64, 32, 8$ |
| Output Layer (Layer 7) | sigmoid | 1 |

The layout of the DNN classifiers for detecting the aggregate preambles is presented in Table I. We consider a layout with six hidden layers (i.e., $D = 6$), which have $512, 256, 128, 64, 32, 8$ neurons, respectively. ReLU is used as an activation function in the neurons of all the hidden layers. The output layer uses a sigmoid activation function to determine $\theta^l_a$.

To obtain the DNN weights $w^{i,i'}_d$, the backpropagation algorithm is invoked multiple times since the final values of these weights are sensitive to their random initialization. We choose the weights $w^{i,i'}_d$ that achieve the lowest probability of false alarm for the validation set as long as the probability of misdetection is less than $10\%$ when SNR $= 0$ dB. For each run of the backpropagation algorithm, a batch size of 64 samples is used and 20 epochs (i.e., iterations) are applied. The detection of the aggregate preambles requires that both preambles are successfully detected while allocating half of the




(a)




(b)

Fig. 16. (a) The probability of misdetection for the DNN-based and the threshold-based receiver structures versus SNR in the aggregate preambles case. (b) The probability of false alarm of the DNN-based and the threshold-based receiver structures versus SNR in the aggregate preambles case.

transmit power to each one of them. As a result, the proposed receiver architecture is expected to have a higher probability of misdetection than the conventional LTE PRACH. However, the aggregate preamble sequence design provides a larger set of preambles to support the random access procedure in mIoT networks.

Fig. 16 (a) shows the probability of misdetection versus the SNR. These results are obtained by testing the DNN-based PRACH receiver using a test set with multiple devices and a fixed SNR. As the SNR increases, all the preamble detection receivers achieve a lower probability of misdetection. The threshold-based receiver has a lower probability of misdetection than the DNN-based receiver when the SNR is less than $-6$ dB. However, the DNN-based receiver outperforms the threshold-based receiver when the SNR is greater than $-3$ dB. Note that, in small cell deployments, the SNR is expected to be high due to the low propagation loss. Hence, the proposed aggregate preamble sequence design can be used for random access in small cell deployments supporting a massive number of IoT devices to make use of the reduced collision probability and the improved average total service time.

Fig. 16 (b) shows that the DNN-based PRACH receiver achieves a probability of false alarm that is less than 0.11 over the considered SNR range from $-12$ dB to 0 dB when detecting the aggregate preambles. An advantage of the proposed preamble sequence design compared to that in [1] is that it maintains a lower false alarm probability as the number of the simultaneous IoT devices increases. For the DNN-based PRACH receiver in the aggregate preambles case, a high probability of false alarm occurs at 0 dB, and this probability decreases as the SNR becomes lower.

### *E. Impact of Training SNR on Preamble Detection Performance*

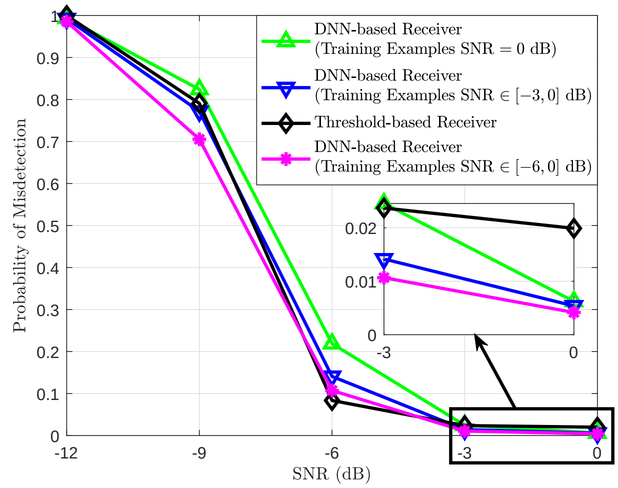
In this subsection, we investigate the impact of the value of SNR that is used for generating the training data sets on the preamble detection performance. We train the DNN-based receiver for detecting the proposed aggregate preambles using: (a) training samples generated assuming that all the IoT devices have a received SNR of 0 dB (similar to Section IV.D); (b) training samples generated assuming that the IoT devices have a received SNR that ranges from $-3$ dB to 0 dB; (c) training samples generated assuming that the IoT devices have a received SNR that ranges from $-6$ dB to 0 dB. Figs. 17 (a) and 17 (b) show the probability of misdetection and the probability of false alarm for the aforementioned training strategies, respectively.

We observe from Fig. 17 (a) that the training strategies that use an SNR range for training the DNN-based PRACH receiver (e.g., strategies (b) and (c)) can achieve a lower probability of misdetection than the strategy that uses samples from a single SNR value (e.g., strategy (a)). This can be explained by the fact that if the DNN classifiers are trained with samples from different SNR values, they achieve better performance in detecting testing samples with lower SNR. However, strategy (a) still achieves the lowest probability of misdetection at SNR $=$ 0 dB. In addition, we note that the DNN-based receiver achieves a lower probability of misdetection than the threshold-based receiver at SNR values greater than $-3$ dB.

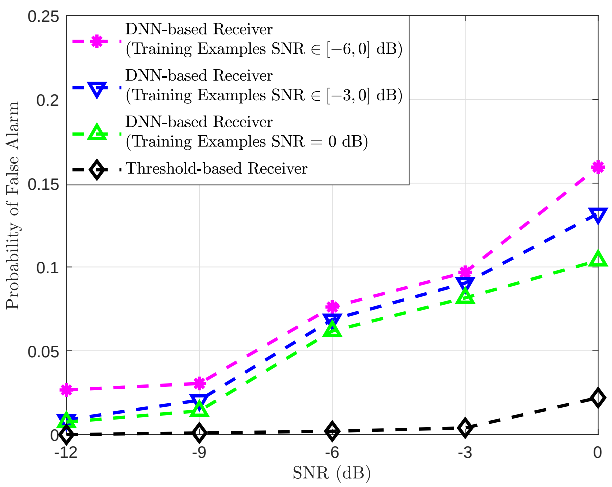
On the other hand, training with higher SNR samples (e.g., strategy (a)) causes the DNN classifiers to not learn to predict preambles at low SNR (i.e., low values of correlation) which enables the DNN classifiers to avoid false alarms as shown in Fig. 17 (b). In contrast, training with low SNR samples (e.g., strategies (b) and (c)) teaches the DNN classifiers to predict more preambles at lower SNR values causing more false alarms compared to training with higher SNR (e.g., strategy (a)).

### *F. Computational Complexity*

We evaluate the computational complexity of the detection stages of both the threshold-based receiver and the DNN-based receiver after the correlation vectors $c[n]$ and $c'[n]$ have been obtained. For the threshold-based receiver, we need to compare $N_{\text{IDFT}}$ inputs with a certain threshold $C$ or $C'$, which has a complexity of $O(N_{\text{IDFT}})$. These comparisons are carried out at most $M+1$ times to detect all the primary and




(a)




(b)

Fig. 17. (a) The probability of misdetection versus SNR for the DNN-based PRACH receiver in the aggregate preambles case with different training SNR ranges. (b) The probability of false alarm versus SNR for the DNN-based PRACH receiver in the aggregate preambles case with different training SNR ranges.

secondary preambles. This results in an overall complexity of $O(MN_{\text{IDFT}})$.

For the DNN-based receiver, although the training process may incur a higher computational complexity due to the application of backpropagation algorithm, the training process can be performed offline and it does not impact the computational complexity of the online preamble detection. Hence, the complexity of the DNN-based detection is due to the processing of the input correlation vectors in the trained DNN modules, i.e., the complexity of the forward propagation. For a given DNN module, propagation of the inputs from layer $d$ with $I_d$ neurons to layer $d+1$ with $I_{d+1}$ neurons requires multiplying a $1 \times I_d$ vector by an $I_d \times I_{d+1}$ matrix, which has a computational complexity of $O(I_d I_{d+1})$. Also, the computational complexity of applying the activation function at any layer $d \in [1, D+1]$ is $O(I_d)$. Hence, forward propagation from layer $d$ to layer $d+1$ has a computational complexity of $O(I_d I_{d+1} + I_{d+1}) = O(I_d I_{d+1})$. For worst-case complexity evaluation, we assume $I_d = \max\{N_{\text{IDFT}}, I_1, \ldots, I_D, 1\} = N_{\text{IDFT}}$ (where $I_0 = N_{\text{IDFT}}$). Since the DNN has $D$ hidden layers, the computational complexity of a single DNN module is $O(DN_{\text{IDFT}}^2)$. Given $M$ DNN modules that can be used at most $M+1$ times to detect all primary and secondary preambles, the overall complexity of the DNN-based detection stage is $O(M^2 DN_{\text{IDFT}}^2)$.

## G. Discussion

The proposed preamble sequence design can increase the number of available preambles (i.e., reduce the probability of preamble collision), which is essential for supporting mIoT applications. Moreover, the preambles can easily be detected using the threshold-based receiver. However, there are potential tradeoffs which require further study and are highlighted in this subsection.

*1) Preamble Sequence Planning:* The proposed preamble sequence design consumes more roots per cell as we need one primary root and $M$ secondary roots compared to $1-5$ roots per cell for most LTE PRACH planning schemes. However, the secondary roots are much less utilized than the primary root. Avoiding the allocation of the secondary roots in the neighbouring cells can mitigate the impact of PRACH interference between cells.

*2) DNN-based Receiver Performance:* Overall, the threshold-based receiver achieves a balance between the probability of false alarm and the probability of misdetection. However, the DNN-based receiver has the potential to achieve good performance in multi-label classification problems with multiple classes. The DNN-based receiver with training strategy (c) (where the training samples are generated with received SNR values ranging from $-6$ dB to 0 dB) has a lower probability of misdetection than that of the threshold-based receiver in the SNR regions from $-12$ dB to $-9$ dB and from $-3$ dB to 0 dB. In addition, the DNN-based receiver with training strategies (a) and (b) (where the training samples are generated with received SNR values of 0 dB and received SNR values ranging from $-3$ dB to 0 dB, respectively) has a lower probability of misdetection than the threshold-based receiver in the high SNR region from $-3$ dB to 0 dB.

*3) Training SNR Range:* If the training set includes samples with a wide range of SNR values (including low SNR values), this will falsely bias the DNN-based receiver towards assuming that random access preambles are received (i.e., more false alarms) at high SNR values. Training with high SNR samples would overcome the aforementioned problem, but it would also result in a performance degradation at low SNR values.

*4) Practical Considerations:* The proposed aggregate preamble sequence design is a good candidate for adoption in future standards as it only requires minor changes to the conventional PRACH transmitter and receiver design. For the deployment of the proposed DNN-based receiver in practical systems, sufficient training data should be collected from the real environment before online operation. The real datasets can be obtained by transmitting multiple aggregate preamble sequences from different locations in the cell and storing the resulting correlation vector information after the first stages of the PRACH receiver. If a limited range of SNR values is required for training the DNN-based receiver, then the devices in the cellular network can transmit aggregate preambles after establishing connections with the base station. In this case, the received SNR range is known for each training sample. The training data obtained from the real environment can be used to either train the DNN modules or tune DNN modules that have been pre-trained with simulation-based data using the concept of transfer learning [45]. We note that the DNN weights of the DNN-based receiver can differ from cell to cell due to differences in the real environment, i.e., the DNN-based receiver can be customized for each cell and environment type (e.g., urban, suburban, or rural).

*5) Aggregate Preambles in New Radio (NR):* LTE PRACH [9] and New Radio (NR) PRACH [46] have many similarities. First, a similar contention-based random access procedure is used as shown in Fig. 1. The preambles in both LTE PRACH and NR PRACH are generated by applying cyclic shifts to a Zadoff-Chu sequence. In addition, the supported preamble formats in LTE PRACH (i.e., formats 0, 1, 2, and 3) are still supported in NR PRACH. NR PRACH supports many short preamble formats with preamble repetition (e.g., formats A1–A3, B1–B4, C0, and C2), mainly for indoor environments and small cells. The new short preamble formats are generated using Zadoff-Chu sequences of length $N_{zc} = 139$ instead of 839 for the longer formats. The NR PRACH signals are generated in a way that is similar to LTE PRACH [46]. Hence, the aggregate preambles and the proposed receiver design (i.e., threshold-based receiver and DNN-based receiver) can be used for decoding the aggregate preambles resulting from aggregating two short format preambles. For the DNN-based receiver, the DNN modules may require a different architecture with different number of neurons and hidden layers according to the dimension of the correlation vector $c[n]$.

## V. Conclusion

In this paper, we proposed a preamble sequence design to support the mIoT use case by aggregating two Zadoff-Chu preamble sequences to enlarge the set of available random access preambles. We proposed corresponding transmitter and receiver architectures that require only minor changes compared to the conventional LTE PRACH architecture. The proposed preamble sequence design reduced the probability of preamble collision to less than $10^{-4}$. We found that selecting a subset of preambles that meet a certain PAPR threshold criterion can reduce the energy consumption of battery-powered IoT devices without considerably increasing the preamble collision probability.

In addition, we proposed a new DNN-based PRACH receiver architecture, which can detect if a specific preamble sequence is received given the correlation vector between the received signal and possible preamble sequences. Furthermore, the threshold-based and the DNN-based PRACH receiver architectures successfully decoded the new aggregate preamble sequences with a low probability of misdetection at high SNR (e.g., SNR $\geq -6$ dB) and a low probability of false alarm of less than $0.11$. We also investigated the impact of the SNR range used for training the DNN-based receiver on the detection performance.

For future work, it is an interesting direction to study the preamble sequence planning (i.e., assigning primary roots to cells) and its impact on aggregate preamble detection in multi-cell scenarios. In addition, the transition from training with simulation-based data to real environment-based data is an interesting topic for future research.

## References


[1] A. E. Mostafa, V. W. S. Wong, S. Liao, R. Schober, M. Ding, and F. Wang, "Aggregate preamble sequence design for massive machine-type communications in 5G networks," in *Proc. of IEEE Global Commun. Conf. (Globecom)*, Abu Dhabi, UAE, Dec. 2018.

[2] Cisco, "Cisco annual Internet report (2018–2023)," Mar. 2020.

[3] 3GPP TS 38.300 V16.0.0, "NR and NG-RAN - Overall description - Stage 2 (Release 16)," Dec. 2019.

[4] ITU-R M.2410-0, "Minimum requirements related to technical performance for IMT-2020 radio interface(s)," Nov. 2017.

[5] F. Ghavimi and H. H. Chen, "M2M communications in 3GPP LTE/LTE-A networks: Architectures, service requirements, challenges, and applications," *IEEE Commun. Surveys & Tuts.*, vol. 17, pp. 525–549, Oct. 2014.

[6] Next Generation Mobile Networks (NGMN) Alliance, "NGMN 5G white paper," Feb. 2015.

[7] A. Laya, L. Alonso, and J. Alonso-Zarate, "Is the random access channel of LTE and LTE-A suitable for M2M communications? A survey of alternatives," *IEEE Commun. Surveys & Tuts.*, vol. 16, no. 1, pp. 4–16, Dec. 2013.

[8] 3GPP TS 36.300 V16.4.0, "Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN) - Overall description - Stage 2 (Release 16)," Dec. 2020.

[9] 3GPP TS 36.211 V16.4.0, "Evolved Universal Terrestrial Radio Access (E-UTRA) - Physical channels and modulation (Release 16)," Dec. 2020.

[10] S.-Y. Lien, T.-H. Liau, C.-Y. Kao, and K.-C. Chen, "Cooperative access class barring for machine-to-machine communications," *IEEE Trans. Wireless Commun.*, vol. 11, no. 1, pp. 27–32, Jan. 2012.

[11] S. Duan, V. Shah-Mansouri, Z. Wang, and V. W. S. Wong, "D-ACB: Adaptive congestion control algorithm for bursty M2M traffic in LTE networks," *IEEE Trans. Veh. Technol.*, vol. 65, pp. 9847–9861, Dec. 2016.

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

[13] M. Lee, Y. Kim, Y. Piao, and T. Lee, "Recycling random access opportunities with secondary access class barring," *IEEE Trans. Mobile Comput.*, vol. 19, no. 9, pp. 2189–2201, Sep. 2020.

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

[15] K. Lee, J. Shin, Y. Cho, K. S. Ko, D. K. Sung, and H. Shin, "A group-based communication scheme based on the location information of MTC devices in cellular networks," in *Proc. of IEEE Int. Conf. on Commun. (ICC)*, Ottawa, Canada, Jun. 2012.

[16] N. Jiang, Y. Deng, A. Nallanathan, X. Kang, and T. Q. S. Quek, "Analyzing random access collisions in massive IoT networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 10, pp. 6853–6870, Oct. 2018.

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

[18] S. Moon, H. Lee, and J. Lee, "SARA: Sparse code multiple access-applied random access for IoT devices," *IEEE Internet of Things J.*, vol. 5, no. 4, pp. 3160–3174, Aug. 2018.

[19] N. K. Pratas, H. Thomsen, C. Stefanovic, and P. Popovski, "Code-expanded random access for machine-type communications," in *Proc. of IEEE Globecom Workshops*, Anaheim, CA, Dec. 2012.

[20] E. Paolini, C. Stefanovic, G. Liva, and P. Popovski, "Coded random access: Applying codes on graphs to design random access protocols," *IEEE Commun. Mag.*, vol. 53, no. 6, pp. 144–150, Jun. 2015.

[21] Z. Chen, Q. Yao, H. H. Yang, and T. Q. S. Quek, "Massive wireless random access with successive decoding: Delay analysis and optimization," *IEEE Trans. Commun.*, vol. 67, no. 1, pp. 457–471, Jan. 2019.

[22] R. Abbas, M. Shirvanimoghaddam, Y. Li, and B. Vucetic, "Random access for M2M communications with QoS guarantees," *IEEE Trans. Commun.*, vol. 65, no. 7, pp. 2889–2903, Jul. 2017.

[23] D. Malak, H. Huang, and J. G. Andrews, “Throughput maximization for delay-sensitive random access communication,” *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 709–723, Dec. 2018.

[24] U. Raza, P. Kulkarni, and M. Sooriyabandara, “Low power wide area networks: An overview,” *IEEE Commun. Surveys & Tuts.*, vol. 19, no. 2, pp. 855–873, Jan. 2017.

[25] 3GPP:R1-1712357, “NR RACH preamble format for capacity enhancement,” *3GPP TSG RAN WG1, CATT*, Aug. 2017.

[26] 3GPP:R1-1711600, “PRACH preamble sequences and formats for capacity enhancement and beam management,” *3GPP TSG RAN WG1, Huawei*, Jun. 2017.

[27] J. S. Kim, S. Lee, and M. Y. Chung, “Efficient random-access scheme for massive connectivity in 3GPP low-cost machine-type communications,” *IEEE Trans. Veh. Technol.*, vol. 66, no. 7, pp. 6280–6290, Jul. 2017.

[28] R. A. Pitaval, B. M. Popovic, P. Wang, and F. Berggren, “Overcoming 5G PRACH capacity shortfall: Supersets of Zadoff-Chu sequences with low-correlation zone,” *IEEE Trans. Commun.*, vol. 68, no. 9, pp. 5673–5688, Sep. 2020.

[29] T. A. Pham and B. T. Le, “A proposed preamble detection algorithm for 5G-PRACH,” in *Proc. of Int. Conf. Advanced Technol. for Commun. (ATC)*, Hanoi, Vietnam, Oct. 2019.

[30] T. Kim, I. Bang, and D. K. Sung, “An enhanced PRACH preamble detector for cellular IoT communications,” *IEEE Commun. Lett.*, vol. 21, no. 12, pp. 2678–2681, Dec. 2017.

[31] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 5438–5453, May 2015.

[32] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[33] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, “Learning to optimize: Training deep neural networks for interference management,” *IEEE Trans. Signal Process.*, vol. 66, no. 20, pp. 5438–5453, Oct. 2018.

[34] W. Cui, K. Shen, and W. Yu, “Spatial deep learning for wireless scheduling,” *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1248–1261, Jun. 2019.

[35] J. Cui, Z. Ding, P. Fan, and N. Al-Dhahir, “Unsupervised machine learning based user clustering in mmWave-NOMA systems,” *IEEE Trans. Wireless Commun.*, vol. 17, no. 11, pp. 7425–7440, Nov. 2018.

[36] T. O’Shea and J. Hoydis, “An introduction to deep learning for the physical layer,” *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, Dec. 2017.

[37] S. Dörner, S. Cammerer, J. Hoydis, and S. t. Brink, “Deep learning based communication over the air,” *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 132–143, Feb. 2018.

[38] N. Farsad and A. Goldsmith, “Neural network detection of data sequences in communication systems,” *IEEE Trans. Signal Process.*, vol. 66, no. 21, pp. 5663–5678, Nov. 2018.

[39] 3GPP TS 36.104 V17.0.0, “Base station (BS) radio transmission and reception (Release 17),” Dec. 2020.

[40] M. Zhang and Z. Zhou, “A review on multi-label learning algorithms,” *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 8, pp. 1819–1837, Aug. 2014.

[41] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1*. Cambridge, MA, USA: MIT Press, 1986, ch. Learning Internal Representations by Error Propagation, pp. 318–362.

[42] G. Hinton, N. Srivastava, and K. Swersky, “Neural networks for machine learning-Lecture 6a: Overview of mini-batch gradient descent,” [online] Available: https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf, Toronto, Canada, Jul. 2012.

[43] S. H. Han and J. H. Lee, “An overview of peak-to-average power ratio reduction techniques for multicarrier transmission,” *IEEE Wireless Commun.*, vol. 12, no. 2, pp. 56–65, Apr. 2005.

[44] 3GPP TR 37.868 V11.0.0, “Study on RAN improvements for machine-type communications,” Sep. 2011.

[45] S. J. Pan and Q. Yang, “A survey on transfer learning,” *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[46] 3GPP TS 38.211 V16.4.0, “NR: Physical channels and modulation (Release 16),” Dec. 2020.

**Ahmed Elhamy Mostafa** (S’15) received the B.Sc. degree (with highest honors) in electronics engineering and the M.Sc. degree (with highest honors) in electronics and communications engineering from the American University in Cairo (AUC), Cairo, Egypt, in 2012 and 2015, respectively. He was a wireless application engineer with the Wireless Communications Lab, Intel Corporation, from 2012 to 2014. He is currently pursuing the Ph.D. degree with the University of British Columbia (UBC), Vancouver, BC, Canada. He has been a recipient of the Four Year Doctoral Fellowship (4YF) at UBC since 2015. He was also awarded the Graduate Support Initiative (GSI) Award at UBC in 2016 and 2017. His research interests include the Internet of Things (IoT), machine learning for wireless communications, 5G wireless networks, and radio resource allocation. Mr. Mostafa served as a TPC Member of the *IEEE VTC-Spring*’17. He also served as a Reviewer of *IEEE Transactions on Wireless Communications*, *IEEE Transactions on Vehicular Technology*, *IEEE Access*, *IEEE Communications Letters*, *IEEE Wireless Communications Letters*, and several conferences.

**Vincent W.S. Wong** (S’94, M’00, SM’07, F’16) received the B.Sc. degree from the University of Manitoba, Winnipeg, MB, Canada, in 1994, the M.A.Sc. degree from the University of Waterloo, Waterloo, ON, Canada, in 1996, and the Ph.D. degree from the University of British Columbia (UBC), Vancouver, BC, Canada, in 2000. From 2000 to 2001, he worked as a systems engineer at PMC-Sierra Inc. (now Microchip Technology Inc.). He joined the Department of Electrical and Computer Engineering at UBC in 2002 and is currently a Professor. His research areas include protocol design, optimization, and resource management of communication networks, with applications to wireless networks, smart grid, mobile edge computing, and Internet of Things. Currently, Dr. Wong is an Executive Editorial Committee Member of *IEEE Transactions on Wireless Communications*, an Area Editor of *IEEE Transactions on Communications* and *IEEE Open Journal of the Communications Society*, and an Associate Editor of *IEEE Transactions on Mobile Computing*. He has served as a Guest Editor of *IEEE Journal on Selected Areas in Communications*, *IEEE Internet of Things Journal*, and *IEEE Wireless Communications*. He has also served on the editorial boards of *IEEE Transactions on Vehicular Technology* and *Journal of Communications and Networks*. He was a Tutorial Co-Chair of *IEEE Globecom*’18, a Technical Program Co-chair of *IEEE VTC*2020-*Fall* and *IEEE SmartGridComm*’14, as well as a Symposium Co-chair of *IEEE ICC*’18, *IEEE SmartGridComm* (’13, ’17) and *IEEE Globecom*’13. He is the Chair of the IEEE Vancouver Joint Communications Chapter and has served as the Chair of the IEEE Communications Society Emerging Technical Sub-Committee on Smart Grid Communications. He was an IEEE Communications Society Distinguished Lecturer (2019 - 2020).

**Yong Zhou** (S’13, M’16) received the B.Sc. and M.Eng. degrees from Shandong University, Jinan, China, in 2008 and 2011, respectively, and the Ph.D. degree from the University of Waterloo, Waterloo, ON, Canada, in 2015. From Nov. 2015 to Jan. 2018, he worked as a postdoctoral research fellow in the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, Canada. He is currently an Assistant Professor in the School of Information Science and Technology, ShanghaiTech University, Shanghai, China. His research interests include Internet of Things, edge machine learning, and reconfigurable intelligent surface.

**Robert Schober** (S'98, M'01, SM'08, F'10) received the Diplom (Univ.) and the Ph.D. degrees in electrical engineering from Friedrich-Alexander University of Erlangen-Nuremberg (FAU), Germany, in 1997 and 2000, respectively. From 2002 to 2011, he was a Professor and Canada Research Chair at the University of British Columbia (UBC), Vancouver, Canada. Since January 2012 he is an Alexander von Humboldt Professor and the Chair for Digital Communication at FAU. His research interests fall into the broad areas of Communication Theory, Wireless Communications, and Statistical Signal Processing.

Robert received several awards for his work including the 2002 Heinz Maier Leibnitz Award of the German Science Foundation (DFG), the 2004 Innovations Award of the Vodafone Foundation for Research in Mobile Communications, a 2006 UBC Killam Research Prize, a 2007 Wilhelm Friedrich Bessel Research Award of the Alexander von Humboldt Foundation, the 2008 Charles McDowell Award for Excellence in Research from UBC, a 2011 Alexander von Humboldt Professorship, a 2012 NSERC E.W.R. Stacie Fellowship, and a 2017 Wireless Communications Recognition Award by the IEEE Wireless Communications Technical Committee. Since 2017, he has been listed as a Highly Cited Researcher by the Web of Science. Robert is a Fellow of the Canadian Academy of Engineering, a Fellow of the Engineering Institute of Canada, and a Member of the German National Academy of Science and Engineering. From 2012 to 2015, he served as Editor-in-Chief of the *IEEE Transactions on Communications*. Currently, he serves as Member of the Editorial Board of the *Proceedings of the IEEE* and as VP Publications for the IEEE Communication Society (ComSoc).

**Zhe Luo** received the B.Sc. degree in electronic and information engineering from Shanghai Jiao Tong University, Shanghai, China, in 2010, and the Ph.D. degree in information and communication engineering from Shanghai Jiao Tong University, Shanghai, China, in 2015. From 2015 to 2018, he worked as a research scientist at Shanghai Bell (now Shanghai Nokia Bell). He joined Huawei in 2018 and is currently a senior engineer. His research areas include 5G evolution, 6G, IoT, and wireless communication theory and techniques.

**Shuri Liao** received the B.Sc. degree in communication engineering from Nanjing University of Science and Technology, Nanjing, China, in 2014, and the M.Sc. degree in information and communication engineering from Southeast University, Nanjing, China, in 2017. He is currently a research engineer with Huawei Technologies Co., Ltd., Shanghai, China. His current research interests include massive connectivity technology of Machine-Type Communication and key technologies for wireless communications.

**Mengying Ding** received the B.Sc. degree in electronics and information engineering from Huazhong University of Science and Technology, Wuhan, China, in 2012, and the M.Sc. degree in electrical engineering from the University of Minnesota, Twin Cities, USA, in 2014. She is currently a senior engineer with Huawei Technologies Co., Ltd., Shanghai. Her research interests include 5G/6G wireless communication, spectrum management, and artificial intelligence.