

# Preamble Detection in Asynchronous Random Access Using Deep Learning

Muhammad Usman Khan<sup>1</sup>, Graduate Student Member, IEEE, Enrico Testi<sup>2</sup>, Member, IEEE, Enrico Paolini<sup>3</sup>, Senior Member, IEEE, and Marco Chiani<sup>4</sup>, Fellow, IEEE

**Abstract**—Grant-free random access protocols are among the enabling techniques for massive machine-type communications, where a large number of devices activate sporadically and transmit short packets, typically containing a preamble (or a pilot sequence), without any resource allocation from the base station (BS). One of the critical tasks to be accomplished by the BS is thus the preamble-based detection of the transmitted packets. This letter proposes a deep learning (DL)-based solution for detecting preambles in an asynchronous grant-free random access uplink scenario, assuming multiple antennas at the BS. The DL-based approach outperforms the classical correlator-based approach.

**Index Terms**—Deep learning, grant-free random access, massive machine-type communication, preamble detection, 6G.

## I. INTRODUCTION

IN RECENT years, the demand for wireless data transmission has grown tremendously, leading to the rise of new applications involving communication among machines. In a conventional cellular communication system, resources are allocated to the users in a coordinated manner. However, resource allocation would be highly inefficient in massive machine-type communications (mMTC) scenario due to control signalling overhead. To address these challenges, grant-free random access-based approaches have been proposed. In such schemes, devices transmit packets without coordination with the base station (BS) over the shared time or frequency resources. Over the years, random access protocols have evolved from ALOHA to more sophisticated protocols involving repetitions of packets and successive interference cancellation (SIC), with the aim to reduce signalling related to grants management and packets retransmission [1], [2], [3].

In grant-free access, detecting transmissions by identification of packet preambles is one of the most critical tasks for the BS, as failures in this initial step preclude correct packet reception. This problem is particularly challenging in asynchronous access schemes, where the time is not organized into global slots and frames and packets may in principle arrive at the receiver at any time. Compared to synchronous access protocols, asynchronous grant-free ones are characterized by a minimum amount of control signalling, which reduces the

burden on the network control plain and improves the device battery life. In [4], a correlator-based approach is used to detect packets in a satellite-based scenario, where devices start private and asynchronous virtual frames (VFs) independently of each other and transmit multiple replicas of a packet within them. In [5], a deep learning (DL)-based solution is proposed for the detection of preambles in satellite communication. Both approaches assumed single antenna receiver and additive white Gaussian noise (AWGN) channel. A convolutional neural network (CNN) architecture is presented in [6] to identify the active user preambles in a slotted synchronous grant-free random access scenario with a single antenna at the BS. In [7] a neural network and logistic regression was developed to detect orthogonal preambles, and their multiplicity, for random access in Long Term Evolution systems. In [8], a closed-form expression for the probability of detection of tagged preamble sequences at Next Generation NodeB is proposed.

In a mMTC scenario, we have to take into account a different propagation model, characterized by fading, shadowing, and possibly multiple antennas at the receiver. The main contributions of this letter are summarised as follows.

- We perform preamble detection in an asynchronous grant-free random access uplink scenario exploiting multiple antennas at the BS.
- We take into account a channel model with fading, path-loss, and shadowing, assuming no power control. Due to uncoordinated transmissions, preamble detection is performed by the BS before channel estimation.
- We propose a DL-based preamble detection method consisting in a CNN that strikes a good trade-off between performance and complexity, compared to a classical correlator-based approach.

The rest of this letter is organized as follows. We present the system model in Section II. In Section III, we explain the CNN architecture and correlator-based approach. Section IV contains the computational complexity analysis. Numerical results along with simulation setup are given in Section V. Conclusions are drawn in Section VI.

Matrices, vectors, and scalars are represented by boldface uppercase, boldface lowercase, and lowercase letters, respectively. The real and imaginary parts of a complex number are indicated as  $\Re(\cdot)$  and  $\Im(\cdot)$ , respectively. The operations  $(\cdot)^T$  and  $(\cdot)^H$  denote the transpose and conjugate transpose, respectively. Notation  $\mathcal{U}(a, b)$  indicates a uniform distribution between  $a$  and  $b$ . The normal and circularly-symmetric complex normal distributions with mean 0 and variance  $\sigma^2$  are denoted by  $\mathcal{N}(0, \sigma^2)$  and  $\mathcal{CN}(0, \sigma^2)$ , respectively.

## II. SYSTEM MODEL

We consider an asynchronous grant-free random access uplink scenario, where users are uniformly distributed within

Manuscript received 23 August 2023; accepted 15 October 2023. Date of publication 19 October 2023; date of current version 9 February 2024. This work was supported in part by the CNIT National Laboratory WiLab and the WiLab-Huawei Joint Innovation Center and in part by the European Union through the Italian National Recovery and Resilience Plan of NextGenerationEU, partnership on Telecommunications of the Future under Grant PE00000001 (RESTART). The associate editor coordinating the review of this article and approving it for publication was H. J. Yang. (Corresponding author: Muhammad Usman Khan.)

The authors are with CNIT/WiLab, DEI, University of Bologna, 47522 Cesena, Italy (e-mail: muhammadusman.khan8@unibo.it; enrico.testi@unibo.it; e.paolini@unibo.it; marco.chiani@unibo.it).

Digital Object Identifier 10.1109/LWC.2023.3325918

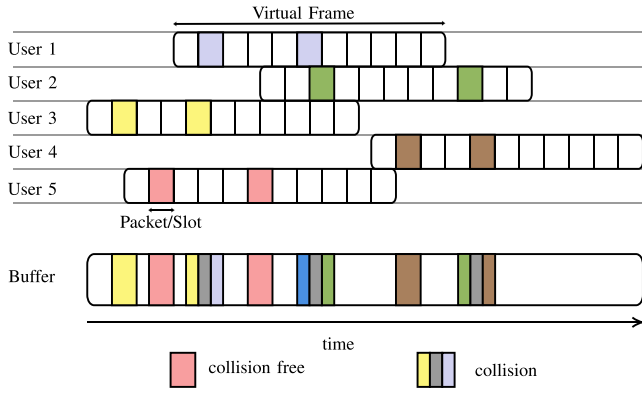


Fig. 1. Pictorial representation of the users initiating virtual frame and transmitting replicas in an asynchronous scenario.

an annulus with inner and outer circles of radius  $D_{\min}$  and  $D_{\max}$ , respectively. The BS is positioned in the center of the annulus. Each device has a single antenna whereas the BS is equipped with  $M$  antennas. The number of users becoming active in an uplink symbol time follows a Poisson distribution with mean  $\lambda$ . When a user becomes active, it initiates a VF comprising  $N_S$  slots, with each slot duration equal to the packet length as shown in Fig. 1. The VF is local to the device: the BS is unaware of the starting time of VFs but it is aware of the number of slots in a VF. Each user transmits multiple packet replicas to boost performance as in [1], [3]. To transmit  $N_{\text{rep}}$  replicas of the packet, the user selects  $N_{\text{rep}}$  slots from the set  $\{1, \dots, N_S\}$  without replacement and with uniform probability. The packet transmission is considered symbol-wise synchronous. A packet consists of a preamble of  $N_P$  symbols,  $\mathbf{s} = [s_1, \dots, s_{N_P}]^T \in \mathbb{C}^{N_P \times 1}$ , which is the same for all users, and a user-specific data payload of length  $N_D$ .

We assume a Rayleigh block fading channel model with no power control and with a coherence time equal to the packet (and virtual slot) time. Accordingly, the channel gain between a device and one BS antenna is constant during transmission of a packet, but independent from replica to replica from the same user. We also assume the independence of the channel gains between a single device and different BS antennas. The  $N_{\text{rep}}$  replicas from the same user experience the same path-loss and large-scale fading, but independent Rayleigh-distributed small-scale fading. The vector of received samples at the  $M$  BS antennas at symbol time  $i$ ,  $\mathbf{y}(i) \in \mathbb{C}^{M \times 1}$ , may be expressed as

$$\mathbf{y}(i) = \sum_{j \in \mathcal{A}_P} \mathbf{h}_j p_j(i) + \sum_{l \in \mathcal{A}_D} \mathbf{h}_l q_l(i) + \mathbf{n}(i) \quad (1)$$

where

- $\mathcal{A}_P$  and  $\mathcal{A}_D$  are the set of users transmitting a preamble and data symbol at  $i^{\text{th}}$  sample time, respectively;
- $p_j(i)$  is the symbol of preamble  $\mathbf{s}$  transmitted by user  $j \in \mathcal{A}_P$  and  $q_l(i)$  represents the data symbol transmitted by user  $l \in \mathcal{A}_D$  at the  $i^{\text{th}}$  sample time;
- $\mathbf{h}_k = [h_{k,1}, \dots, h_{k,M}]^T \in \mathbb{C}^{M \times 1}$  is the vector of channel gains between the  $k^{\text{th}}$  user and the BS, where  $h_{k,m} \sim \mathcal{CN}(0, \sigma_{h_k}^2)$  for  $m = 1, \dots, M$ . The variance  $\sigma_{h_k}^2$  is given by  $\gamma (D_{\max}/d_k)^\beta$ , where  $\gamma$  is the log-normal shadowing coefficient in linear scale, i.e.,  $\gamma_{\text{dB}} \sim \mathcal{N}(0, \sigma_{\text{dB}}^2)$ ,  $\beta$  is the path-loss exponent, and  $d_k$  is the distance between

the  $k^{\text{th}}$  device and the BS. The distance  $d_k$  is randomly distributed as  $\sqrt{D_{\min}^2 + (D_{\max}^2 - D_{\min}^2) \cdot \mathcal{U}(0, 1)}$ ;

- $\mathbf{n}(i) \in \mathbb{C}^{M \times 1}$  is the vector of independent and identically distributed noise samples, each distributed as  $\mathcal{CN}(0, \sigma_n^2)$ .

### III. PREAMBLE DETECTION

This section presents the proposed DL-based approach, which consists of a CNN that performs preamble detection starting from raw received samples at the BS. We also introduce a correlator-based methodology as a benchmark, showing how it can be derived from the generalized likelihood ratio test (GLRT) design method.

#### A. CNN Architecture

Assume we want to check if  $N_P$  consecutive samples at an initial offset  $i_0$  correspond to a preamble or not. We then consider the observation matrix  $\mathbf{R} = \{r_{i,j}\} = [\mathbf{y}(i_0), \mathbf{y}(i_0 + 1), \dots, \mathbf{y}(i_0 + N_P - 1)]$ . As the samples are complex, we split the received samples into real and imaginary parts and then add the reference preamble sequence, obtaining the matrix

$$\mathbf{Y} = \begin{bmatrix} \Re(\mathbf{R}) & \Im(\mathbf{R}) \\ \Re(\mathbf{s}^T) & \Im(\mathbf{s}^T) \end{bmatrix}. \quad (2)$$

Matrix  $\mathbf{Y} \in \mathbb{R}^{(M+1) \times 2N_P}$  is a feature map obtained from the raw received samples at the BS and is the input to the DL model. Extensive investigation revealed that concatenating the reference preamble with the received symbols and feeding the resulting matrix into the DL model yields better performance. We explored various architectures with different numbers, types, and sizes of layers, to find a good balance between performance and complexity. Finally, considering the 2-dimensional nature of the input feature map, we selected the CNN architecture depicted in Fig. 2. In fact, besides reducing complexity, convolutional layers allow the exploitation of features shared among  $M$  antennas. In particular, we used two convolutional layers with 8 and 4 filters of the same size, respectively, without any padding. The filter tries to learn the mapping between the received symbols and the reference preamble, increasing the classification performance. The convolutional layers are followed by fully-connected and dropout layers. The fully-connected layer contains multiple neurons, each receiving multiple inputs, producing [9]

$$\mathbf{z} = f(\mathbf{W}\mathbf{a} + \mathbf{b}) \quad (3)$$

where  $\mathbf{W} \in \mathbb{R}^{\text{out} \times \text{in}}$  represents the weight matrix. The input to the layer is denoted by  $\mathbf{a} \in \mathbb{R}^{\text{in} \times 1}$ , while the bias is represented by  $\mathbf{b} \in \mathbb{R}^{\text{out} \times 1}$  [9]. The non-linear operation  $f(\cdot)$ , known as the activation function, is a crucial component of the DL model. It enables the network to learn non-linear relationships between input and output. One of the most widely used activation functions is rectified linear unit (ReLU), defined as  $\mathbf{a} = \max(0, \mathbf{z})$ , where  $\max(\cdot)$  operation is performed element-wise [9].

To mitigate overfitting and enhance the model's generalization capacity, the CNN incorporates a dropout layer, which randomly drops connections between the fully-connected

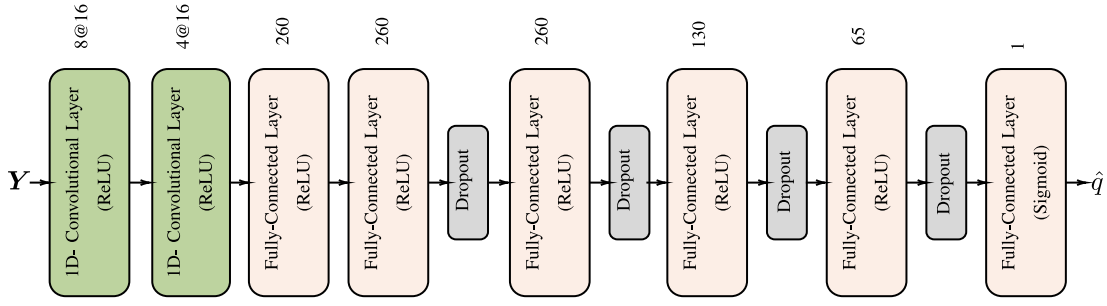


Fig. 2. A schematic representation of the architecture of the proposed CNN for preamble detection, where the size of each layer is specified. For instance, the first convolutional layer has 8 filters of dimensions 16, and the first fully-connected layer contains 260 neurons.

layers during the training phase. This process helps to minimize the dependencies between neurons.

The preamble detection is essentially a binary classification problem, i.e., classifying the received symbols as preamble or non-preamble. Consequently, the last fully-connected layer consists of only one neuron employing sigmoid as an activation function. It is defined as  $\hat{q} = 1/(1 + e^{-z})$ , where  $\hat{q}$  estimates the likelihood of the input being a preamble. A threshold of 0.5 is applied to this value to perform classification.

The neural network architecture is linked to a cost function, which equals zero for ideal classification and increases when the inputs are misclassified. From this standpoint, we utilize a binary cross-entropy loss, which is formulated as

$$\mathcal{J}(q, \hat{q}) = -q \log \hat{q} - (1 - q) \log(1 - \hat{q}) \quad (4)$$

where  $q$  is the true label, equal to 1 if the input samples correspond to a (possibly interfered) preamble and to 0 otherwise. The objective of the training is to determine the suitable weights of the DL model that minimizes the cost function. In our approach, we utilize the Adam optimizer, which is an extended version of the gradient descent algorithm [9], adopting mini-batches to enhance training efficiency.

### B. Correlator-Based Approach

We compare our DL-based solution with a classical approach based on hypothesis testing. Let the two hypotheses be  $\mathcal{H}_0$ ,  $\mathcal{H}_1$  corresponding to noise-only samples, and (non-interfered) preamble plus noise, respectively.<sup>1</sup> Conditional on the channel gains, the logarithmic likelihoods of the two hypotheses are

$$\ln f_{\mathbf{R}|\mathcal{H}_0}(\mathbf{R}|\mathcal{H}_0) = K - \sum_{m=1}^M \sum_{i=1}^{N_P} \frac{|r_{m,i}|^2}{2\sigma_n^2} \quad (5)$$

and

$$\ln f_{\mathbf{R}|\mathcal{H}_1, \mathbf{h}}(\mathbf{R}|\mathcal{H}_1, \mathbf{h}) = K - \sum_{m=1}^M \sum_{i=1}^{N_P} \frac{|r_{m,i} - h_{k,m} s_i|^2}{2\sigma_n^2} \quad (6)$$

where  $K$  is a common constant, and  $k$  is the user transmitting the preamble. Similar to [10], since channel gains shall be estimated from the preamble and, therefore, are unknown during preamble detection, we can resort to the GLRT, which

<sup>1</sup>In the derivation, we do not take into account interference.

after some simple derivations, gives the test based on a non-coherent correlation

$$\Lambda(\mathbf{R}) = \sum_{m=1}^M \left| \sum_{i=1}^{N_P} r_{m,i} s_i^H \right|^2 \underset{\mathcal{H}_1}{\overset{\mathcal{H}_0}{\gtrless}} \eta \quad (7)$$

where  $\eta$  is the test threshold.

## IV. COMPUTATIONAL COMPLEXITY

We evaluate the computational complexity in terms of floating point operations (FLOPs). We assume the real addition, subtraction, and multiplication, as a single FLOP while division and exponential operations as 4 and 8 FLOPs, respectively. For the complex addition and subtraction operations, we consider two FLOPs and complex multiplication as six FLOPs [11], [12], [13].

### A. CNN Complexity

The number of FLOPs of a convolutional layer is given by

$$C_{cv} = 2N_{cv} F_{cv} G_{cv} D_{cv} \quad (8)$$

where  $N_{cv}$ ,  $F_{cv}$ ,  $G_{cv}$ , and  $D_{cv}$  represent the number of convolution filters, size of the filter, number of channels, and output shape, respectively. The output shape  $D_{cv}$  is expressed as  $(I - F + 2 \cdot P)/S + 1$ , where  $I$ ,  $F$ ,  $P$ , and  $S$  specify the input size, filter size, padding, and stride. The ReLU is applied to the output of the convolutional layers, resulting in

$$C_{ReLU} = N_{cv} D_{cv}. \quad (9)$$

The number of FLOPs in a fully-connected layer (3) can be expressed as

$$C_{FC} = 2 \cdot \text{in} \cdot \text{out} + \text{out}. \quad (10)$$

The dropout layer involves elementwise multiplication operations; for a single operation, the complexity is 1. The sigmoid function and thresholding in the last layer yield 14 FLOPs. The total complexity of the CNN, with  $\alpha$  representing the number of neurons in the first fully-connected layer, is given by

$$\begin{aligned} C_{CNN} = & 2N_{cv1} F_{cv1} M D_{cv1} + 2N_{cv2} F_{cv2} N_{cv1} D_{cv2} \\ & + N_{cv1} D_{cv1} + N_{cv2} D_{cv2} + 2N_{cv2} D_{cv2} \alpha \\ & + \frac{13\alpha^2}{4} + 5\alpha + 14. \end{aligned} \quad (11)$$

### B. Correlator Complexity

The inner sum  $\sum_{i=1}^{N_P} r_{m,i} s_i^H$  for the  $m^{\text{th}}$  antenna requires  $N_P$  and  $N_P - 1$  complex multiplication and addition operations, respectively. The  $|\cdot|^2$  operation results in 2 real multiplication operations and 1 real addition operation for each antenna. The total computational cost is then

$$C_{\text{corr}} = 8M N_P + 2M - 1. \quad (12)$$

## V. IMPLEMENTATION AND RESULTS

### A. Simulation Setup

The performance analysis, for both the correlator-based approach (7) and the CNN, is conducted assuming  $M = 32$  and  $M = 64$  antennas at the BS, with signal-to-noise ratio (SNR) per antenna ranging from  $-10$  dB to  $20$  dB. The SNR is defined as  $\text{SNR} = 1/\sigma_n^2$ , and represents the median SNR per antenna element for a user on the edge of the cell. Clearly, the average SNR inside the cell is higher than that on the boundary. The minimum and maximum distances of a user from the BS are  $D_{\text{min}} = 5$  m and  $D_{\text{max}} = 100$  m, respectively. The path-loss exponent is set to  $\beta = 2$  and the standard deviation of the log-normal shadowing is taken as  $\sigma_{\text{dB}} = 3$ .

We consider a preamble and payload of length  $N_P = 63$  and  $N_D = 150$ , respectively. The preamble sequence is generated by a linear feedback shift register of length 6 with primitive polynomial  $p(x) = x^6 + x + 1$  over the Galois field GF(2). The sequence is designed to have good (aperiodic) auto- and cross-correlation properties and allow accurate channel estimation. The pilot sequence bits are then converted to  $N_P = 63$  binary phase shift keying symbols with unitary energy using  $x_i = e^{j(\pi/4 + \phi_i \pi)}$ , where  $\phi_i \in \{0, 1\}$  is the  $i^{\text{th}}$  bit of the pilot sequence and  $x_i$  is the corresponding complex symbol. The payload of each user is populated randomly with quadrature phase-shift keying symbols having an equal probability of occurrence.

For generating a dataset, we consider a buffer of  $M \times 213,000$  complex symbols, i.e., one sub-buffer for each antenna. The number of active users in a symbol time in the buffer is randomly generated by Poisson distribution with  $\lambda$  equal to  $[0.05, 0.25, 0.5, 0.75, 1, 1.2, 1.45] \times 10^{-2}$ , such that the average number of packet collisions per slot ranges from 1 to 7. When a user becomes active, it initiates a virtual frame consisting of  $N_S = 100$  slots, where each slot equals the packet size. The user sends  $N_{\text{rep}} = 2$  replicas in slots chosen randomly without replacement. As we consider an asynchronous scheme, the user packet may get partially or fully interfered by packets from other users; at time  $i$  the received sample is mathematically expressed by (1).

We extract the samples for training and test sets from the buffer after the placement of packets, as described above. For the preamble case, we obtain the  $N_P$  consecutive samples from the buffer that contains the entire preamble sequence. For the non-preamble case, we randomly select  $N_P$  consecutive samples from the buffer that do not satisfy the preamble case condition. To have a well-balanced dataset, we obtain an equal number of examples for both preamble and non-preamble cases and consider an equal number of examples for each  $\lambda$  value. For instance, we generate  $8 \cdot 10^3$  examples per  $\lambda$

per class (preamble or non-preamble). For each SNR value, we train a separate CNN but with the same architecture as depicted in Fig. 2. Each dataset comprises  $1.12 \cdot 10^5$  samples, which are split into 70% training set and 30% test set.

For each hyperparameter (learning rate, epochs, mini-batch size, dropout rate, number of neurons, etc.) of the model, we evaluate the performance on a range of values by fixing other hyperparameters and selecting the one which results in the best performance [9]. Due to space constraints, details on the hyperparameter investigation are omitted. The final result of this search yielded learning rate, epochs, and mini-batch size 0.001, 20, and 512, respectively, with the architecture depicted in Fig. 2. We employ a drop-out rate of 0.2 and 0.3 for  $M = 32$  and  $M = 64$ , respectively.

### B. Numerical Results

As for performance metrics, we use the detection rate (or recall) which is defined as  $R = \text{TP}/(\text{TP} + \text{FN})$ , and the false alarm rate  $F = \text{FP}/(\text{FP} + \text{TN})$ , where the true positives, true negatives, false positives, and false negatives are denoted by TP, TN, FP, and FN, respectively. TP and TN correspond to the instances when the preamble and non-preamble cases are correctly identified, respectively. Likewise, FP and FN indicate the number of instances when the non-preamble/preamble is misclassified as preamble/non-preamble, respectively.

The receiver operating characteristics (ROC) curves are reported in Fig. 3(a) and Fig. 3(b) for  $M = 32$  and  $M = 64$ , respectively. The curves are obtained for the correlator-based approach by varying the threshold  $\eta$ , from 0 to  $\eta_{\text{max}}$  with a step size of 10000, where  $\eta_{\text{max}}$ , depending on the number of antennas at the BS, is the maximum correlation value over all the examples. In the same figures, the CNN ROC curves are obtained by varying the threshold  $\eta$ , from 0 to 1, with a step size of 0.01. The points represent the performance of the CNN classifier at  $\eta = 0.5$ .

The CNN-based classifier shows a significant improvement over the correlation-based detector. Indeed, it can be observed that the same detection rate provided by the CNN can be achieved with the correlator-based approach but at a higher false alarm rate, for all SNRs. For example, with  $\text{SNR} = 20$  dB,  $M = 32$ , and assuming a target detection rate  $R = 0.998$ , the correlator gives a false alarm rate  $F = 0.023$ , while the CNN achieves  $F = 0.001$ . As the number of antennas increases from  $M = 32$  to  $M = 64$ , the improvement given by the CNN is even more pronounced. In Fig. 3(a), the correlation-based approach for SNR 10 dB outperforms the 20 dB one because the hypothesis testing-based method does not consider interference. To assess the robustness of our proposed CNN architecture in scenarios where the BS is also unaware of the median SNR at the edge of the cell, we train a single CNN model, referred to as CNNx, on examples obtained with all the considered SNR values. We compare the performance of the CNNx, with CNN models trained specifically for each SNR value, simply regarded as CNN. The results of this comparison are presented in Table I for  $M = 32$  and  $M = 64$ . In CNNx, we utilize a dropout rate of 0.1 and 0.2 for  $M = 32$  and  $M = 64$ , respectively. As expected, the numerical results show that training a single model on multiple SNRs leads to performance degradation. However,



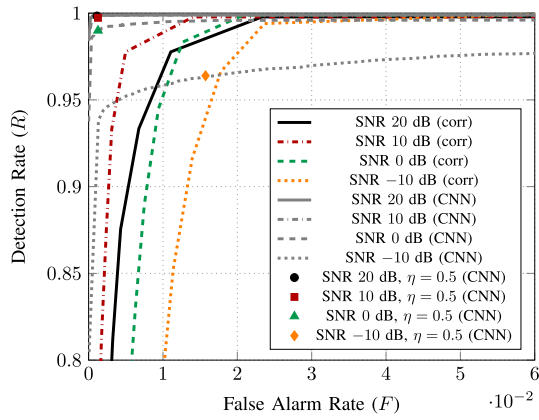
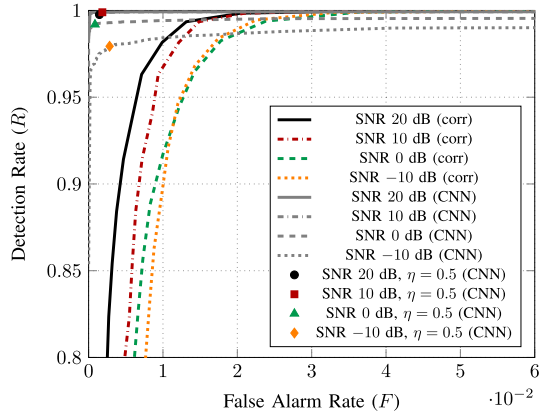
(a) Base station with  $M = 32$  antennas.(b) Base station with  $M = 64$  antennas.

Fig. 3. Comparison between the CNN and the correlator.

TABLE I  
COMPARISON BETWEEN CNN AND CNNx,  $\eta = 0.5$ 

SNR (dB)	CNN		CNNx		
	$R$	$F$	$R$	$F$	
20	0.9984	0.0011	0.9553	0.0192	$M = 32$
10	0.9973	0.0012	0.9578	0.0126	
0	0.9899	0.0012	0.9529	0.0175	
-10	0.9639	0.0157	0.9292	0.1185	
20	0.9976	0.0014	0.9546	0.0078	$M = 64$
10	0.9989	0.0018	0.9563	0.0081	
0	0.9919	0.0008	0.9442	0.0145	
-10	0.9794	0.0028	0.9298	0.0472	

TABLE II  
COMPUTATIONAL COST

	$M = 32$	$M = 64$
Correlator	$1.62 \times 10^4$	$3.24 \times 10^4$
CNN	$1.23 \times 10^6$	$2.14 \times 10^6$

the performance degradation is only about 3.5 – 4.3% and 4.3 – 5% for the detection rate, in the case of  $M = 32$  and  $M = 64$ , respectively. The computational cost of the algorithms is reported in Table II. It can be observed that the correlator is computationally less expensive than the CNN. However, the latter outperforms the former as discussed earlier. Furthermore,

as the number of antennas at the BS increases from  $M = 32$  to  $M = 64$ , the computational complexity of the correlator doubles, while for the CNN it increases by a factor 1.74. This is due to the fact that we employed convolutional layers which reduce the computational complexity, as only the first layer has a linear relationship with the number of antennas  $M$ , while the rest of the architecture is independent of  $M$ . To ensure a comprehensive complexity analysis, we measure the execution time in seconds of both algorithms on a Nvidia Quadro RTX 5000 GPU using Keras. For  $M = 64$ , the execution time per sample for CNN and correlator-based approach is  $3.08 \times 10^{-4}$  s and  $2.69 \times 10^{-6}$  s, respectively.

## VI. CONCLUSION

In this letter, we have proposed a CNN architecture to detect the preamble in an asynchronous grant-free random access uplink scenario with no power control. The proposed deep learning model employs convolutional layers, which not only reduce the computational complexity but also extract the features shared between the antennas. The results, obtained for several values of the SNR and number of antennas, show that the CNN achieves better performance when compared to a classical solution based on the correlation, at the price of an increase in complexity.

## REFERENCES

- [1] E. Casini, R. De Gaudenzi, and O. Del Rio Herrero, "Contention resolution diversity slotted ALOHA (CRDSA): An enhanced random access scheme for satellite access packet networks," *IEEE Trans. Wireless Commun.*, vol. 6, no. 4, pp. 1408–1419, Apr. 2007.
- [2] G. Liva, "Graph-based analysis and optimization of contention resolution diversity slotted ALOHA," *IEEE Trans. Commun.*, vol. 59, no. 2, pp. 477–487, Feb. 2011.
- [3] E. Paolini, G. Liva, and M. Chiani, "Coded slotted ALOHA: A graph-based method for uncoordinated multiple access," *IEEE Trans. Inf. Theory*, vol. 61, no. 12, pp. 6815–6832, Dec. 2015.
- [4] F. Clazzer, F. Lazaro, G. Liva, and M. Marchese, "Detection and combining techniques for asynchronous random access with time diversity," in *Proc. SCC*, Hamburg, Germany, Feb. 2017, pp. 1–6.
- [5] E. Recayte, A. Munari, and F. Clazzer, "Grant-free access: Machine learning for detection of short packets," in *Proc. 10th Adv. Satell. Multimedia Syst. Conf.*, Graz, Austria, Oct. 2020, pp. 1–7.
- [6] J. H. I. de Souza and T. Abrão, "Deep learning-based activity detection for grant-free random access," *IEEE Syst. J.*, vol. 17, no. 1, pp. 940–951, Mar. 2023.
- [7] D. Magrin, C. Pielli, Č. Stefanović, and M. Zorzi, "Enabling LTE RACH collision multiplicity detection via machine learning," in *Proc. WiOPT*, Avignon, France, Jun. 2019, pp. 1–8.
- [8] S. Riolo, D. Panno, and L. Miuccio, "Modeling and analysis of tagged preamble transmissions in random access procedure for mMTC scenarios," *IEEE Trans. Wireless Commun.*, vol. 20, no. 7, pp. 4296–4312, Jul. 2021.
- [9] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: <http://www.deeplearningbook.org>
- [10] M. Chiani, "Noncoherent frame synchronization," *IEEE Trans. Commun.*, vol. 58, no. 5, pp. 1536–1545, May 2010.
- [11] X. Wu, *Performance Evaluation, Prediction and Visualization of Parallel Systems*. Boston, MA, USA: Kluwer Acad. Publ., 1999.
- [12] W. Kim, Y. Ahn, and B. Shim, "Deep neural network-based active user detection for grant-free NOMA systems," *IEEE Trans. Commun.*, vol. 68, no. 4, pp. 2143–2155, Apr. 2020.
- [13] M. U. Khan, E. Paolini, and M. Chiani, "Enumeration and identification of active users for grant-free NOMA using deep neural networks," *IEEE Access*, vol. 10, pp. 125616–125625, 2022.