Deep Learning for Replica Detection and Combining in Asynchronous Grant-Free mMTC

# Deep Learning for Replica Detection and Combining in Asynchronous Grant-Free mMTC

Dania De Crescenzo, Enrico Testi and Enrico Paolini

***Abstract*—The Internet of Things has fostered the rapid growth of new services such as massive machine-type communications (mMTC), featuring transmission of short packets from a massive number of low duty-cycle devices. grant-free (GF) access is gaining an increasing interest to support the mMTC traffic and tackle the related challenges. This letter proposes a deep learning-based solution for GF asynchronous mMTC systems where packet diversity is exploited to enhance the probability of successful uplink transmission. The proposed processing aims at detecting twin packets, i.e., replicas of the same packet, among the received signal samples, which allows combining them to achieve a higher successful decoding probability. Numerical results highlight remarkable gains over more conventional approaches.**

***Index Terms*—Asynchronous grant-free access, massive machine-type communication, deep learning, convolutional neural networks.**

## I. Introduction

grant-free (GF) random access represents a very attractive option in massive machine-type communications (mMTC), as it lets a large number of sporadically active devices contend to deliver short packets to a common base station (BS) without any resource allocation procedure [1], [2]. artificial intelligence (AI) holds an important role in enhancing the performance of these systems, including latency reduction, network throughput increase, and achievement of a higher reliability [3].

Over time, random access protocols have evolved from simple ALOHA to more advanced ones such as (synchronous or asynchronous) coded random access (CRA), featuring packet diversity and successive interference cancellation (SIC) [4]–[6]. In these schemes, devices send multiple copies of their packets (called replicas), each containing information about the positions of its twins. At the receiver, SIC can effectively remove interference due to decoded packets and their replicas, thereby enhancing the decoding probability of users that are experiencing high interference levels. asynchronous contention resolution diversity ALOHA (ACRDA) [7] is a prominent example of asynchronous CRA protocol: users are asynchronous with each other, but each of them is synchronized to its local virtual frame (VF) and sends its replicas in discrete positions within its VF. The VFs of different users are asynchronous. Another example of asynchronous CRA protocol is enhanced contention resolution ALOHA (ECRA) in [8]–[10]. In it, combining techniques are exploited to resolve collision patterns where SIC alone is unable to succeed: knowing the position of multiple replicas of the same packet allows exploitation of joint decoding. The main drawback is the need to know the replica positions prior to decoding.

Several solutions based on deep learning (DL) have been proposed in recent literature to enhance the performance of massive GF access (e.g., [11]–[13]). In [11] the packet detection capabilities of two machine learning algorithms (neural networks and random forests) for GF access are investigated. In [12], a DL-based solution is proposed to enhance packet detection in asynchronous GF uplink. In [13], the composite issues of activity detection, channel estimation, and data recovery in asynchronous massive access are tackled via DL. The works [14], [15] address packet replica identification; however, the solutions proposed therein rely on correlation and not on learning-based techniques.

In the framework of the growing interest in the exploitation of AI to enhance massive GF access, in this letter we propose a solution to the problem of localizing replicas in asynchronous CRA, leveraging DL techniques. Due to uncoordinated transmissions, the BS needs to initially perform preamble detection, followed by channel estimation, demapping, and decoding. The decoding operation, in particular, heavily affects the overall packet loss rate (PLR). Remarkably, the proposed approach enhances the decoding step by allowing combining of twins for joint decoding. The main contributions of this letter may be summarized as follows:

- We design a deep neural network (DNN) capable of classifying pairs of packets as twins (i.e., replicas form the same user) or non-twins.
- We propose a processing scheme at the receiver that exploits the designed DNN to classify twin packets among the detected but undecodable ones, so as to perform joint decoding by replica combining.
- We compare the performance of the proposed approach with that of conventional asynchronous CRA, a genie-aided processing, and an energy detector (ED)-based one.

Throughout the paper, matrices, vectors, and scalars are represented by boldface uppercase, boldface lowercase, and lowercase letters, respectively. $\Re(\cdot)$ and $\Im(\cdot)$ indicate real and imaginary parts of a complex number, respectively. The $\ell^2$ norm is indicated as $\|\cdot\|^2$. The operators $(\cdot)^T$ and $(\cdot)^H$ denote transpose and conjugate transpose, while we use $[\boldsymbol{A};\boldsymbol{B}]$ to denote row-wise concatenation of matrices $\boldsymbol{A}$ and $\boldsymbol{B}$. The uniform distribution between $a$ and $b$ is denoted as $\mathcal{U}(a,b)$. 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.

The authors are with CNIT/WiLab, DEI, University of Bologna, via dell'Università 50, 47522 Cesena (FC), Italy. Email: {dania.decrescenzo2, enrico.testi, e.paolini}@unibo.it. This work is supported by the European Union under the Italian National Recovery and Resilience Plan of NextGenerationEU, partnership on "Telecommunications of the Future" (PE00000001 - "RESTART").

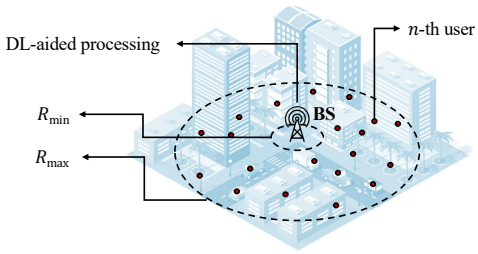


Fig. 1. An illustration of the GF mMTC scenario where a plethora of users aims to transmit data towards a single BS.

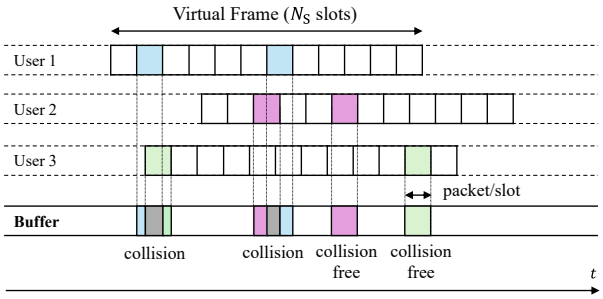


Fig. 2. Pictorial representation of the users initiating virtual frame and transmitting replicas in an asynchronous scenario.

## II. System Model

With reference to Fig. 1, we consider a system with a single BS placed at the center of an annulus with inner and outer radii $R_{\min}$ and $R_{\max}$, respectively, and devices (or "users") uniformly distributed within the annulus. The BS is equipped with $M$ antennas, while all devices are single antenna ones. The number of users becoming active in an uplink symbol time is Poisson distributed with mean $\lambda\,[\text{wake up / symbol time}]$.

At wake up, a user starts its own VF, of length $N_S$ slots; the slot time equals the packet time-on-air, as depicted in Fig. 2. Note that VFs are local to the users: although the BS knows the VF length $N_S$, it is unaware of the set of active users and, consequently, of the start time of their VFs. Each user transmits $N_{\text{rep}}$ replicas of its message in $N_{\text{rep}}$ slots of its private VF. These slots are randomly drawn by the user from the set $\{1,\dots,N_S\}$, with uniform probability and without replacement. Packet arrivals at the receiver are assumed symbol-wise synchronous, to enable a tractable and computationally efficient simulation setup [11]. A packet consists of a preamble of $N_P$ symbols, $\boldsymbol{s}=[s_1,\cdots,s_{N_P}]\in\mathbb{C}^{1\times N_P}$, which is a Zadoff-Chu (ZC) sequence common to users, and a user-specific data payload of length $N_D$.

We adopt a Rayleigh block fading channel model with coherence time equal to the packet time-on-air. Thus, the channel gain between a user and any BS antenna remains constant across all symbols of a packet but changes independently from replica to replica. We also assume independent channel gains for a single device across different BS antennas and no power control. The vector of received signal samples at the $M$ BS antennas at symbol time $i$, $\boldsymbol{y}(i)\in\mathbb{C}^{M\times 1}$, can be expressed as

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

where:

- $\mathcal{A}_P$ and $\mathcal{A}_D$ are the subsets of users that transmit a preamble and data symbol at symbol time $i$, respectively;
- $p_j(i)$ is the symbol of preamble $\boldsymbol{s}$ transmitted by user $j\in\mathcal{A}_P$ at symbol time $i$; $q_l(i)$ is the data symbol transmitted by user $l\in\mathcal{A}_D$ at symbol time $i$;
- $\boldsymbol{h}_k=[h_{k,1},\cdots,h_{k,M}]^T\in\mathbb{C}^{M\times 1}$, where $h_{k,m}\sim\mathcal{CN}(0,\sigma^2_{h_k})$ for $m=1,...,M$, is the vector of channel gains between user $k$ and the BS;
- $\boldsymbol{n}(i)\in\mathbb{C}^{M\times 1}$ is the vector of independent and identically distributed noise samples, each distributed as $\mathcal{CN}(0,\sigma^2_{\text{n}})$.

We let $\sigma^2_{h_k}=\gamma\,(R_{\max}/d_k)^\beta$, where $\gamma$ is the log-normal shadowing coefficient in linear scale, i.e., $\gamma_{\text{dB}}=10\log_{10}(\gamma)\sim\mathcal{N}(0,\sigma^2_{\text{dB}})$, $\beta$ is the path-loss exponent, and $d_k$ is the distance between user $k$ and the BS. The distance $d_k$ is randomly distributed as $\sqrt{R^2_{\min}+(R^2_{\max}-R^2_{\min})\cdot\mathcal{U}(0,1)}$.

## III. DL-Aided Twin Packets Identification

This section presents the proposed DL model and a more conventional ED-based methodology used as a benchmark. Both techniques aim at detecting twin packets (i.e., replicas from the same user) among non-decoded packets, to enable joint decoding by replica combining, thus improving payload decoding success rate. Then, we introduce the proposed processing scheme at the receiver, that exploits the designed DL-model. We adopt $N_{\text{rep}}=2$, as recommended in previous studies in the high traffic regime (e.g., [7]).

### A. CNN Architecture

We want to check if a pair of sequences of $N_D$ consecutive received samples are replicas from the same user. Firstly, the two channel gains $\boldsymbol{h}^{(1)}\in\mathbb{C}^{M\times 1}$ and $\boldsymbol{h}^{(2)}\in\mathbb{C}^{M\times 1}$ and the two payloads $\boldsymbol{x}^{(1)}\in\mathbb{C}^{1\times N_D}$ and $\boldsymbol{x}^{(2)}\in\mathbb{C}^{1\times N_D}$ are estimated by maximum likelihood (ML) estimation, as

$$\hat{\boldsymbol{h}}^{(i)}=\frac{1}{\|\boldsymbol{s}\|^2}\boldsymbol{Y}_{\text{p}}^{(i)}\boldsymbol{s}^H \;\text{ and }\; \hat{\boldsymbol{x}}^{(i)}=\frac{1}{\|\hat{\boldsymbol{h}}^{(i)}\|^2}(\hat{\boldsymbol{h}}^{(i)})^H\boldsymbol{Y}_{\text{d}}^{(i)} \qquad (2)$$

where $i\in\{1,2\}$ and $\boldsymbol{Y}_{\text{p}}^{(i)}\in\mathbb{C}^{M\times N_P}$ and $\boldsymbol{Y}_{\text{d}}^{(i)}\in\mathbb{C}^{M\times N_D}$ are the corresponding received pilot and payload symbols. The input to the DL model is represented by the observation matrix $\boldsymbol{Y}\in\mathbb{R}^{2\times(2N_D)}$, a feature map obtained from real and imaginary parts of the two estimated payloads, as

$$\boldsymbol{Y}=\begin{bmatrix}\Re(\hat{\boldsymbol{x}}^{(1)}) & \Im(\hat{\boldsymbol{x}}^{(1)})\\ \Re(\hat{\boldsymbol{x}}^{(2)}) & \Im(\hat{\boldsymbol{x}}^{(2)})\end{bmatrix}. \qquad (3)$$

The twin packets identification task is cast as binary classification problem with two classes, namely, class 1 if the received signal samples correspond to replicas transmitted by the same user, and class 0 otherwise.[1]

Considering the 2D nature of the input feature map, we designed the convolutional neural network (CNN) architecture shown in Table I, inspired by the one proposed in [12]. Simpler networks lack sufficient feature extraction capabilities, while deeper ones add complexity with limited gains. The proposed

[1]Class 0 includes also the case in which the received sequences do not correspond exactly to payload symbols.

TABLE I
CNN ARCHITECTURE AND COMPLEXITY.

| **Layer type** | **Layer size** | **Activation** | **FLOPs** ($N_{\mathrm{D}} = 256$) |
|---|---|---|---|
| Input | $2 \times 2N_{\mathrm{D}}$ | | |
| 2D convolutional | $8@2 \times 32$ | ReLu | 496,392 |
| 2D convolutional* | $4@1 \times 32$ | ReLu | 664,200 |
| Fully connected* | 260 | ReLu | 936,260 |
| Fully connected* | 130 | ReLu | 67,730 |
| Fully connected* | 65 | ReLu | 16,965 |
| Output | 1 | Sigmoid | 4 |

* Followed by a dropout layer with threshold 0.2.

architecture offers a good trade-off between accuracy and efficiency. In particular, the first convolutional layer contains $8$ filters of size $2 \times 32$, while the second contains $4$ filters of size $1 \times 32$, both with no padding. Then, we alternate fully-connected and dropout layers. The fully-connected layers contain multiple neurons $(260, 130, 65$, and $1$, respectively), with rectified linear unit (ReLU) activation function. The dropout layers help minimizing inter-dependencies between neurons, enhancing the model's generalization capacity. The network ends with a classification layer using sigmoid activation function. We utilize a binary cross-entropy loss, which is formulated as $\mathcal{J}(q, \hat{q}) = -q \log \hat{q} - (1 - q) \log(1 - \hat{q})$, where $q$ is the true label, equal to $1$ if the input samples correspond to a pair of twin packets and to $0$ otherwise, while $\hat{q}$ is the network output label. The training aims to determine the proper weights of the DL model that minimize the cost function. In this framework, we adopt the Adam optimizer, employing mini-batches to improve training efficiency [16]. The computational complexity of the CNN is calculated in terms of floating point operations (FLOPs), as shown in Table I. The overall complexity of the CNN is approximately $2.182 \cdot 10^6$ FLOPs.

### B. ED-Based Approach

We compare our DL-based classification solution with a more conventional detector that uses an ED-based test derived analytically. The two hypotheses of non-twin packets and twin packets can be formulated as

$$\mathcal{H}_0 : \boldsymbol{y}_k^{(1)} = \boldsymbol{h}^{(1)} a_k + \boldsymbol{n}_k^{(1)}, \quad \boldsymbol{y}_k^{(2)} = \boldsymbol{h}^{(2)} b_k + \boldsymbol{n}_k^{(2)} \tag{4}$$

$$\mathcal{H}_1 : \boldsymbol{y}_k^{(1)} = \boldsymbol{h}^{(1)} a_k + \boldsymbol{n}_k^{(1)}, \quad \boldsymbol{y}_k^{(2)} = \boldsymbol{h}^{(2)} a_k + \boldsymbol{n}_k^{(2)} \tag{5}$$

respectively, with $k = 1, ..., N_{\mathrm{D}}$. In (4) and (5), $\boldsymbol{y}_k^{(i)} \in \mathbb{C}^{M \times 1}$ is the $k$-th column of $\boldsymbol{Y}_{\mathrm{d}}^{(i)}$, $i \in \{1, 2\}$, $a_k$ and $b_k$ are the $k$-th payload symbols from two different users under $\mathcal{H}_0$, $\boldsymbol{h}^{(i)}$, $i \in \{1, 2\}$, is the channel gain, while $\boldsymbol{n}_k^{(i)}$, $i \in \{1, 2\}$, is a vector of noise samples. Assuming for the moment $\boldsymbol{h}^{(i)}$ known at the receiver, we let $\eta_k = \hat{x}_k^{(1)} + \hat{x}_k^{(2)}$ where $\hat{x}_k^{(i)} = (\boldsymbol{h}^{(i)})^H \boldsymbol{y}_k^{(i)} / \|\boldsymbol{h}^{(i)}\|^2$, $i \in \{1, 2\}$, which yields

$$\eta_k | \mathcal{H}_0 = a_k + b_k + \tilde{n}_k^{(1)} + \tilde{n}_k^{(2)}, \ \eta_k | \mathcal{H}_1 = 2a_k + \tilde{n}_k^{(1)} + \tilde{n}_k^{(2)} \tag{6}$$

where $\tilde{n}_k^{(i)} = (\boldsymbol{h}^{(i)})^H \boldsymbol{n}_k^{(i)} / \|\boldsymbol{h}^{(i)}\|^2$, $i \in \{1, 2\}$. As $\mathbb{E}[\eta_k | \mathcal{H}_0] = \mathbb{E}[\eta_k | \mathcal{H}_1] = 0$, the variances of $\eta_k$ under the two hypotheses are given by

$$\mathbb{V}[\eta_k | \mathcal{H}_0] = 2 + \sigma_n^2 \left( \frac{1}{\|\boldsymbol{h}^{(1)}\|^2} + \frac{1}{\|\boldsymbol{h}^{(2)}\|^2} \right) \tag{7}$$

$$\mathbb{V}[\eta_k | \mathcal{H}_1] = 4 + \sigma_n^2 \left( \frac{1}{\|\boldsymbol{h}^{(1)}\|^2} + \frac{1}{\|\boldsymbol{h}^{(2)}\|^2} \right). \tag{8}$$

The detector makes a decision about $\mathcal{H}_0$ versus $\mathcal{H}_1$ based on the variance of $\eta_k$. To implement the test in practice, once the two preambles have been detected, (2) is applied to obtain the estimates $\hat{\boldsymbol{h}}^{(i)}$ and $\hat{\boldsymbol{x}}^{(i)}$, $i \in \{1, 2\}$, and the sample variance of $\eta_k$ is used. After simple manipulation, the test takes the form

$$\frac{1}{N_{\mathrm{D}} - 1} \sum_{k=1}^{N_{\mathrm{D}}} |\eta_k|^2 - \sigma_n^2 \left( \frac{1}{\|\hat{\boldsymbol{h}}^{(1)}\|^2} + \frac{1}{\|\hat{\boldsymbol{h}}^{(2)}\|^2} \right) \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \varepsilon \tag{9}$$

where $\varepsilon$ is a detection threshold to be set to ensure a target false alarm probability. Theoretically, the test assumes values between $2$ and $4$. Actually, this range extends to account for practical impairments such as channel estimation errors.

### C. DL-Aided Processing Scheme

We now describe the DL-aided processing scheme at the receiver. The traditional processing scheme operates as follows: (i) preamble detection is performed using a correlator with a very low threshold to minimize missed detections; (ii) channel and payload estimation are performed as per (2); (iii) upon successful channel decoding and cyclic redundancy check (CRC) test, we subtract the interference of the decoded packet and of its twin (in this latter case, by estimating the channel using preamble plus data), and repeat the processing from (i) until no more messages can be decoded. In the DL-aided processing scheme, an additional step is introduced when no more messages can be decoded: (iv) pairs of replicas among the detected but undecoded packets are identified using the proposed CNN; if two packets are classified as twins and jointly decoded successfully, SIC is performed as in step (iii). Joint decoding of the replica pairs is performed as per (2) using the aggregated pilot and payload received symbols, $[\boldsymbol{Y}_{\mathrm{p}}^{(1)}; \boldsymbol{Y}_{\mathrm{p}}^{(2)}]$ and $[\boldsymbol{Y}_{\mathrm{d}}^{(1)}; \boldsymbol{Y}_{\mathrm{d}}^{(2)}]$, for channel estimation and payload detection, respectively. Step (iv) is repeated until no more messages can be decoded.

## IV. IMPLEMENTATION AND RESULTS

This section describes the setup implemented both for the dataset creation used for the CNN training, and for the simulation of the overall system. Subsequently, we present numerical results for twins classification using the CNN, the ED-based method and a correlation-based approach based on [14], [15]. Finally, we report the performance of the receiver processing system obtained integrating the designed CNN, in comparison to the classical approach that does not exploit twin packets combining and joint decoding.

### A. Simulation Setup

The number of active users that wake up in a symbol time is randomly generated by sampling a Poisson distribution with mean $\lambda$. When a device becomes active, it initiates a VF consisting of $N_{\mathrm{S}} = 100$ slots. Each user transmits $N_{\mathrm{rep}} = 2$ replicas in slots selected randomly without replacement. We consider a preamble and a payload of length $N_{\mathrm{P}} = 63$ and $N_{\mathrm{D}} = 256$, respectively. The preamble sequence, possessing

good auto-correlation properties, is the ZC$(63, 1)$, while the payload consists of equiprobable quadrature phase-shift keying (QPSK) symbols. Bose–Chaudhuri–Hocquenghem (BCH) coding of length 511 (plus an extra null padding bit) is employed. We also assume that messages contain a CRC field that allows message integrity validation after decoding. The minimum and maximum distances between a user and the BS are $R_{\text{min}} = 5$ m and $R_{\text{max}} = 100$ m, respectively. The path-loss exponent and the log-normal shadowing are taken as $\beta = 2$ and $\sigma_{\text{dB}} = 3$, respectively. The signal-to-noise ratio (SNR) is formulated as $\text{SNR} = 1/\sigma_n^2$, and represents the median SNR per antenna for a device at the edge of the cell. During the training phase of the CNNs we consider SNR $= 10$ dB and $M = 32$ antennas at the BS.

We train separate CNNs, each with the architecture described in Table I for different traffic scenarios, i.e., for $\lambda \in \{0.005, 0.01, 0.017, 0.03, 0.05\}$. The networks remain effective under traffic load variations, but perform best when training and inference conditions align. To have well-balanced datasets we collect $10 \cdot 10^3$ examples for each class; 70% are used for training and 30% for validation and testing. To create the datasets, we assume use of a perfect detector, meaning all packets start indices are known. We extract the samples from a buffer of $M \times 319{,}000$ complex symbols, i.e., one sub-buffer for each antenna. A training point consists of a pair of sequences of length $N_{\text{D}}$, twins for class 1 and non-twins for class 0. These samples are then manipulated to create the observation matrix $\boldsymbol{Y}$ defined in (3), given in input to the CNN.[2] For each hyperparameter (learning rate, epochs, mini-batch size, dropout rate, number of neurons, etc.), we test performance by varying each value while keeping the others fixed. This search yielded learning rate, dropout rate and mini-batch size of 0.001, 0.2, and 20, respectively, as the values that lead to the best performance [16].

To implement the receiver processing we utilize a sliding window of length $W = 3N_{\text{S}}(N_{\text{P}}+N_{\text{D}})$ symbols, with a shifting step of $\text{d}W\, N_{\text{S}}(N_{\text{P}} + N_{\text{D}})$ ($\text{d}W = 0.3$, so that two subsequent windows are partially overlapped). The length $W$ was chosen based on prior work, which identified it as a good trade-off between complexity and performance. The BS processes samples in the current window position until no more messages can be decoded; then the window is slid forward. For each simulation, a buffer with length $10N_{\text{S}}(N_{\text{P}} + N_{\text{D}}) = 319{,}000$ symbols per antenna is generated. Simulations are carried out under four different settings: traditional processing [7], DL-aided processing described in Section III-C using the designed CNN, genie-aided processing employing an ideal classifier that always detects all twin pairs, and ED-based processing using (9) for twins classification. The genie-aided and ED-based processing schemes follow the same steps as the DL-aided one, the only difference being that twin detection is performed by an ideal classifier (the genie) or an ED-based test, respectively.

To consider a steady-state traffic situation, both for collection of datasets points and evaluation of system performance, we exclude from our analysis packets transmitted by nodes with wake-up instant within the first and last simulated VFs.

[2]Two packets can be twins only if their start indices are spaced by a multiple of the packet length and their distance is less than one VF. Thus, we immediately label these pairs as non-twins and exclude them from the training set. Moreover, the case of twins, one of which is completely interfered by another packet, is always associated with class 1.

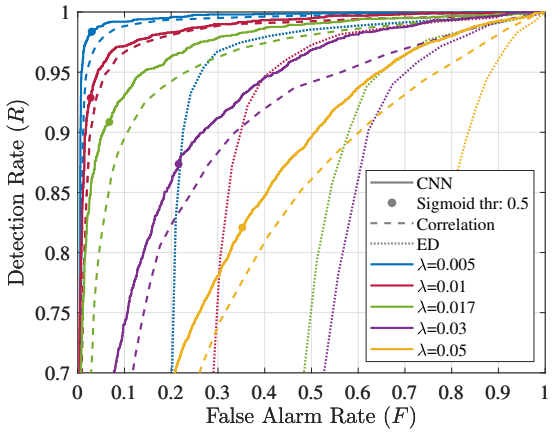


Fig. 3. Performance comparison between the proposed CNN, the ED-based and the correlation-based classifiers. $M = 32$ BS antennas; SNR $= 10$ dB. Markers highlight the CNN's working points obtained with a sigmoid threshold of 0.5.

### B. Numerical Results

*1) CNN performance:* As performance metrics we use the detection rate (or recall) $R = \text{TP}/(\text{TP} + \text{FN})$, and the false alarm rate $F = \text{FP}/(\text{FP} + \text{TN})$, where true positives, true negatives, false positives, and false negatives are denoted by TP, TN, FP and FN, respectively. The cases when twins and non-twins are correctly identified, coincide with TP and TN, respectively. Likewise, FP and FN correspond to the number of instances when twins and non-twins are misclassified.

receiver operating characteristics (ROC) curves in Fig. 3 are obtained with the ED-based approach varying the threshold $\varepsilon$ from 1.8 to 4.2. This range extends beyond the nominal interval of 2 to 4 to account for practical factors such as channel estimation errors and interference. Fig. 3 also shows the ROC curves for the proposed CNN and the correlation-based approach [14], [15]. The CNN outperforms all alternatives, combining high detection accuracy with low false alarm rates. Notably, its advantage over correlation-based detection grows with traffic load, highlighting superior feature extraction and generalization in dense scenarios.

*2) DL-aided receiver performance:* The main performance metric is the packet loss rate (PLR), representing the fraction of user messages lost at the receiver over the total messages sent by active users.

In Fig. 4 we compare the performance of the four different processing schemes: traditional [7], CNN-aided, genie-aided, and ED-aided with threshold $\varepsilon = 3.5$. A fixed $\varepsilon$ yields varying detection and false alarm rates across $\lambda$, but ensures consistently high detection rates, allowing meaningful comparison with the CNNs. The proposed CNN consistently improves PLR across all traffic regimes, outperforming both traditional processing and the ED test. It supports an increase of $\lambda$ by 21.1% and 24.8% at PLR $10^{-2}$ and $10^{-3}$, respectively, confirming its effectiveness across various reliability targets. It is worth noting

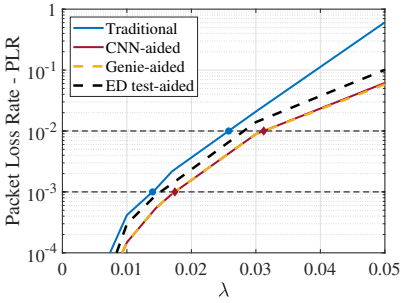


Fig. 4. PLR with $M = 32$ BS antennas, $t = 23$ and SNR $= 0$ dB. Markers highlight the performance gain for PLR $10^{-2}$ and $10^{-3}$.

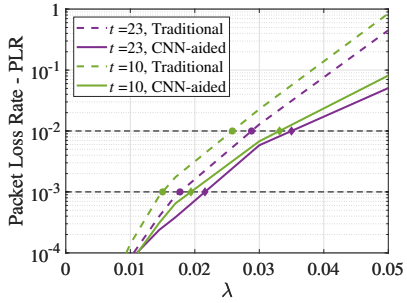


Fig. 5. PLR varying $t$, with $M = 32$ BS antennas, SNR $= 10$ dB. Markers highlight the performance gain for PLR $10^{-2}$ and $10^{-3}$.

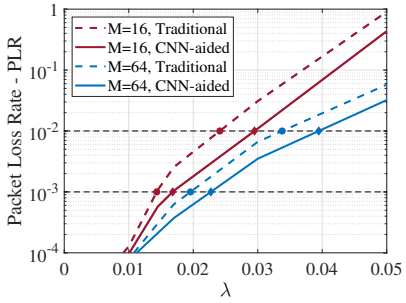


Fig. 6. PLR varying $M$ BS antennas, with $t = 23$, SNR $= 10$ dB. Markers highlight the performance gain for PLR $10^{-2}$ and $10^{-3}$.

that the proposed CNN-aided scheme performs remarkably close to the genie-aided one. This occurs because the twin pairs not detected by the designed CNN are those most affected by interference: decoding fails even if the genie detects them, thus they do not impact the overall PLR.

Fig. 5 shows the PLR obtained using $t = 23$ and $t = 10$ as the number of correctable errors of the BCH code. Comparing traditional processing with the CNN-aided one, with $t = 23$ we obtain an increase of the supported $\lambda$ value of about $21.5\%$ and $21.9\%$ at PLR $10^{-2}$ and $10^{-3}$, respectively, while with $t = 10$ we achieve an increase of about $28.3\%$ and $29.4\%$ at PLR $10^{-2}$ and $10^{-3}$, respectively. We can therefore conclude that the support given by the CNN is more pronounced with a poorer error correction capability.

Remarkably, the CNN input size does not depend on the number of BS antennas, making the proposed solution straightforwardly scalable across different scenarios. In Fig. 6, we show that CNN-aided processing reduces PLR even varying the number of antennas at the BS. With $M = 16$ the CNN permits an increase of supported $\lambda$ of about $22.3\%$ and $17.3\%$ at PLR $10^{-2}$ and $10^{-3}$, respectively, while for $M = 64$ we obtain an increase of about $16.8\%$ and $16.1\%$ at PLR $10^{-2}$ and $10^{-3}$, respectively.

## V. Conclusion

In this letter, we have proposed a DL-aided processing for asynchronous massive access that leverages the feature extraction and classification capabilities of CNNs to improve the system scalability. The designed CNN identifies replicas of the same packet, enabling joint decoding at the receiver. Extensive simulations demonstrate that the CNN achieves strong classification accuracy across all traffic scenarios, consistently reducing the system's PLR. As a result, CNN-aided processing enables the system to handle higher traffic levels compared to conventional methods. Future work could explore extending the approach to higher numbers of replicas ($N_{\text{rep}} > 2$) to better exploit temporal diversity. Another direction is to evaluate the proposed CNN-aided decoder under traffic conditions differing from those used during training.

## References


[1] A. C. Cirik, N. M. Balasubramanya, L. Lampe, G. Vos, and S. Bennett, "Toward the standardization of grant-free operation and the associated NOMA strategies in 3GPP," *IEEE Commun. Standards Mag.*, vol. 3, no. 4, pp. 60–66, Dec. 2019.

[2] 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.

[3] N. Ye, J. An, and J. Yu, "Deep-learning-enhanced NOMA transceiver design for massive MTC: Challenges, state of the art, and future directions," *IEEE Wireless Commun.*, vol. 28, no. 4, pp. 66–73, Aug. 2021.

[4] 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.

[5] 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.

[6] 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.

[7] R. De Gaudenzi, O. del Río Herrero, G. Acar, and E. Garrido Barrabés, "Asynchronous contention resolution diversity ALOHA: Making CRDSA truly asynchronous," *IEEE Trans. Wireless Commun.*, vol. 13, no. 11, pp. 6193–6206, Nov. 2014.

[8] F. Clazzer and C. Kissling, "Enhanced contention resolution Aloha – ECRA," in *Proc. 9th Int. ITG Conf. Syst., Commun. Coding*, Munich, Germany, Jan. 2013.

[9] F. Clazzer, F. Lazaro, G. Liva, and M. Marchese, "Detection and combining techniques for asynchronous random access with time diversity," in *Proc. 11th Int. ITG Conf. Syst., Commun. Coding*, Hamburg, Germany, Feb. 2017.

[10] F. Clazzer, C. Kissling, and M. Marchese, "Enhancing contention resolution ALOHA using combining techniques," *IEEE Trans. Commun.*, vol. 66, no. 6, pp. 2576–2587, Jun. 2018.

[11] E. Recayte, A. Munari, and F. Clazzer, "Grant-free access: Machine learning for detection of short packets," in *Proc. 10th Advanced Satellite Multimedia Syst. Conf. and 16th Signal Process. Space Commun. Workshop*, Graz, Austria, Oct. 2020.

[12] M. U. Khan, E. Testi, E. Paolini, and M. Chiani, "Preamble detection in asynchronous random access using deep learning," *IEEE Wireless Commun. Lett.*, vol. 13, no. 2, pp. 279–283, Feb. 2024.

[13] Y. Bai, W. Chen, B. Ai, and P. Popovski, "Deep learning for asynchronous massive access with data frame length diversity," *IEEE Trans. Wireless Commun.*, vol. 23, no. 6, pp. 5529–5540, Jun. 2024.

[14] H.-C. Bui, K. Zidane, J. Lacan, and M.-L. Boucheret, "A multi-replica decoding technique for contention resolution diversity slotted Aloha," in *Proc. 2015 IEEE Veh. Tech. Conf. Fall*, Boston, MA, USA, Sep. 2015.

[15] S. Zamoum, J. Lacan, M.-L. Boucheret, J.-B. Dupé, and M. Gineste, "Asynchronous packet localization with random SPOTiT in satellite communications," in *Proc. 2019 Int. Symp. Wireless Personal Multimedia Commun.*, Lisbon, Portugal, Nov. 2019.

[16] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.