Enhancing IoT Connectivity in NTN: A Machine Learning Approach for Packet Detection in Grant-Free Access

Pol Simon*, Estefanía Recayte‡, Giuseppe Cocco*, Andrea Munari‡,
* Polytechnic University of Catalonia (UPC), Barcelona, Spain
‡ Institute of Communications and Navigation, German Aerospace Center (DLR), Weßling, Germany
Emails: pol.simon@upc.edu, estefania.recayte@dlr.de, giuseppe.cocco@upc.edu, andrea.munari@dlr.de

***Abstract*—Non-terrestrial networks (NTN) are key to the 6G-Internet of Things (IoT) ecosystem, providing broader coverage and reliable connectivity. As IoT adoption grows, the demand for scalable communication solutions increases. IoT communications, relying on grant-free random access protocols, face challenges in packet detection due to uncoordinated transmissions. Traditional detection methods struggle in NTN environments, especially due to interference and Doppler shifts. This paper proposes a convolutional neural network for efficient packet detection in IoT-NTN scenarios. The method outperforms a traditional correlator-based approach, showing superior detection performance under harsh conditions. Our results highlight the potential of machine learning for enhancing IoT connectivity over NTN.**

I. Introduction

Non-terrestrial networks (NTN) are a fundamental component of the emerging 6G-Internet of Things (IoT) ecosystem, enabling ubiquitous connectivity, broader coverage and improved service reliability. In view of their strategic significance, NTN have been incorporated into the third-generation partnership project (3GPP) standards starting from Release 17. In parallel to this, the widespread adoption of IoT connectivity across a broad range of industrial, commercial, and consumer applications calls for seamless, reliable, and scalable communication solutions.

IoT communications are primarily characterized by sporadic transmissions of short packets, typically occurring in a grant-free manner relying to random access protocols based on variations of the plain ALOHA policy [1] at the link layer. This implies that each device transmits independently and in an uncoordinated fashion, without relying on a centralized scheduling approach. While appealing to support potentially massive population of devices falling within the coverage area and that transmit in unpredictable fashion, random access poses intrinsic challenges. In this respect, for instance, the receiver is not aware of when incoming data will arrive, and packet detection algorithms become paramount to properly trigger the subsequent steps in the decoding chain [2].

Detection is typically performed by pre-pending a preamble to each transmitted packet. The sequence of symbols is known also at the receiver, and can be used to identify the start of an incoming message. This is commonly achieved by performing simple correlation algorithms, which offer generally reasonable performance. The task becomes hovever much more challenging when supporting IoT traffic via NTN low Earth orbit (LEO) satellites due to interference among users as well as to potentially large and non-precompensated Doppler shifts, e.g., in the presence of terminals that have no or limited GNSS capabilities or that are not provided with accurate satellite ephemerides. From this standpoint, the definition of efficient packet detection techniques becomes pivotal towards truly enabling IoT connectivity via NTN, and machine learning (ML) based schemes appear promising. Indeed, the problem at hand can be cast in terms of classification, aiming to determine whether the samples falling within the window of interest correspond to a preamble or not.

Taking the lead from these findings, this work investigates ML solution based on a convolutional neural network (CNN) tuned for the detection of short packets in NTN. The proposed detector is evaluated under LEO link conditions, including Doppler shifts and interference representative of narrowband IoT transmissions. A bank of correlators is used as a benchmark to assess performance in terms of detection and false-alarm probabilities.

Results show that the CNN consistently outperforms traditional correlation-based methods, maintaining reliable detection even under high interference and low SNR. Moreover, the CNN can distinguish packets affected by different interference levels, providing useful side information to optimize subsequent decoding stages. These results confirm the strong potential of ML-based detection for enhancing grant-free IoT access over NTN and motivate further research in this direction. The novelty of this study lies in demonstrating that a lightweight CNN, adapted to the unique Doppler and interference characteristics of NTN, can achieve robust packet detection where classical methods fail.

A. Related Works

Machine learning has been explored for packet detection in grant-free access scenarios as alternative to conventional correlators [3]. In [4], neural networks and random forests were used to detect short packets, showing the potential of ML in multi-user settings. However, the simplified model adopted does not consider NTN impairments, such as Doppler shifts and phase rotations.

The work of G. Cocco is supported by the Ramon y Cajal fellowship program (grant RYC2021-033908-I) funded by the Spanish Ministry for Science and Innovation/State Research Agency MCIN/AEI/10.13039/501100011033 and by the European Union «NextGenerationEU» Recovery Plan for Europe.

E. Recayte and A. Munari would like to thank the Federal Ministry of Research, Technology, and Space (BMFTR) for supporting the xG RIC project as part of the research program Communication Systems “Souverän. Digital. Vernetzt.” (grant number 16KIS2429K) and the Bavarian Ministry of Economic Affairs, Regional Development and Energy.

Deep learning has also been applied to preamble detection in asynchronous random access scenarios, particularly in terrestrial networks. In [5], a CNN-based detector achieved superior accuracy compared to traditional correlator-based methods. The approach remains tailored to terrestrial environments, where Doppler shifts and severe path loss are less critical. Similarly, [6] proposed deep learning-based algorithms for device activity detection in grant-free random access, showing that Zadoff–Chu sequences outperform random preambles and that ML can achieve competitive performance with low complexity.

Such promising results lead to consider machine learning techniques in the context of existing cellular network standards, including 4G LTE, 5G NR, and Narrowband IoT (NB-IoT) [7]–[10] . These studies primarily focus on the PRACH, a crucial mechanism for enabling devices to initiate communication with the network. In [8] a neural network combined with logistic regression was developed to detect orthogonal preambles and their multiplicity in LTE. The use of preamble multiplicity for classifying the number of colliding users and estimating their time of arrival in the initial step of PRACH for NB-IoT using neural networks is also addressed in [10]. Unlike prior works restricted to terrestrial IoT channels, this study is the first to evaluate ML-based packet detection in a NTN environment characterized by LEO-induced Doppler up to ±48 kHz, grant-free access, and large SNR variability. These impairments fundamentally alter the signal and cannot be addressed by existing detectors trained on terrestrial data.

II. System Model

We consider a LEO satellite receiving messages sent by fixed ground terminals spread over its beam coverage. Modeling IoT applications, the potentially large terminal population access the shared uplink channel of bandwidth B to transmit short packets in a sporadic and uncoordinated fashion, following a plain ALOHA policy [1]. Binary phase shift keying (BPSK) modulation is used, and each sent packet is composed of a preamble of $\mathsf{n_P}$ symbols, followed by $\mathsf{n_D}$ symbols of data payload. We denote the whole packet length as $\mathsf{N} = \mathsf{n_P} + \mathsf{n_D}$.

Assuming symbol-level synchronization,[1] the baseband, discrete-time signal model at the satellite receiver for a packet transmitted by a node with slant range d is given by

$$y_n = \sqrt{g(d)}\, x_n e^{j\phi_n} + w_n. \tag{1}$$

for $n \in \{1, \dots, \mathsf{N}\}$. In (1), $w_n \sim \mathcal{CN}(0, 2\sigma^2)$ is additive white Gaussian noise with power σ^2 per complex component, whereas $g(d)$ captures the incoming signal power, considering attenuation due to path loss and other factors. Specifically,

$$g(d) = \frac{P_T G_T G_R}{\ell} \left(\frac{c}{4\pi d\, f_0} \right)^2 \tag{2}$$

where c is the speed of light, f_0 is the carrier frequency, P_T is the transmit power, G_T and G_R are the transmit and receive antenna gains, respectively, and ℓ accounts for other losses. Accordingly, the signal to noise ratio (SNR) for a transmitter at distance d from the satellite is $\Gamma(d) = g(d)/(2\sigma^2)$.

TABLE I
Reference Parameters

Parameter	Value	
satellite altitude, h	600	[km]
satellite speed, $\|\boldsymbol{v}\|$	7.5	[km/s]
carrier frequency, f_0	2.4	[GHz]
bandwidth, B	100	[kHz]
transmit power, P_T	0	[dBW]
transmit antenna gain, G_T	$[-0.6, 5.2]$	[dB]
receive antenna gain, G_R	$[9, 12]$	[dB]
additional losses, ℓ	2.55	[dB]
receiver noise temperature, T	438.93	[K]

Besides attenuation, the signal model in (1) encompasses a phase factor ϕ_n, defined as

$$\phi_n = \theta + \frac{2\pi n f_{\mathsf{D}}}{B}. \tag{3}$$

The term θ in (3) captures a uniformly distributed offset, i.e., $\theta \sim \mathcal{U}[-\pi, \pi)$, which is constant throughout the duration of a packet. In turn, the second term introduces a phase rotation that changes across subsequent symbols, due to the Doppler shift f_{D}. The shift depends on the radial component of the relative velocity of the satellite with respect to the transmitter, and is computed neglecting for simplicity the Earth curvature. To this aim, let $\boldsymbol{r} = (\mathsf{x}, \mathsf{y})$ be the planar coordinates of a terminal within the satellite beam, such that the nadir corresponds with the center of the coordinate system $(0, 0)$, and let $\|\boldsymbol{r}\| = r = \sqrt{\mathsf{x}^2 + \mathsf{y}^2}$. Denote further by h the satellite altitude, and by $\boldsymbol{v} = (v_{\mathsf{x}}, v_{\mathsf{y}})$ its velocity components along the x and y axes. We then have

$$f_{\mathsf{D}} = \frac{v_{\mathsf{eff}}\, f_0}{c}, \quad \text{with } v_{\mathsf{eff}} = (v_{\mathsf{x}} \cos\zeta + v_{\mathsf{y}} \sin\zeta) \cos\varepsilon$$

where v_{eff} is the effective satellite velocity along the line of sight direction, $\varepsilon = \arctan(h/r)$ is the elevation angle and $\zeta = \arctan(\mathsf{y}/\mathsf{x})$ is the azimuth angle.

The reference parameters considered in our study are summarized in Tab. I, and are representative of a LEO satellite providing service to IoT terminals, with minimum elevation angle of $\sim 36^\circ$ and a beam radius of 800 km. In this setting, the SNR ranges between -3 and 10 dB, whereas a Doppler shift up to ± 48 kHz can be experienced. We remark that the latter value is particularly challenging, as the receiver has to cope with frequency offsets of up to half the signal bandwidth.

As discussed, terminal follows a random-access ALOHA policy. This results in packets potentially overlapping at the receiver, generating uncoordinated interference. The overall signal at any time takes thus the form of a sum of a random number of signals, each affected by independent path loss, phase rotation and Doppler plus noise. We consider symbol-level alignment, yet messages of different users can start at different times and thus overlap only partially (if at all).

[1] This simplifying assumption is used to isolate the impact of other impairments such as Doppler shift. The non-timing-synchronous model is currently under study. This approach is adopted to decouple the timing synchronization problem from the Doppler and interference mitigation challenges, which are the primary focus in this NTN study. The results thus provide a fundamental benchmark of the CNN's robustness to uncompensated frequency offsets.

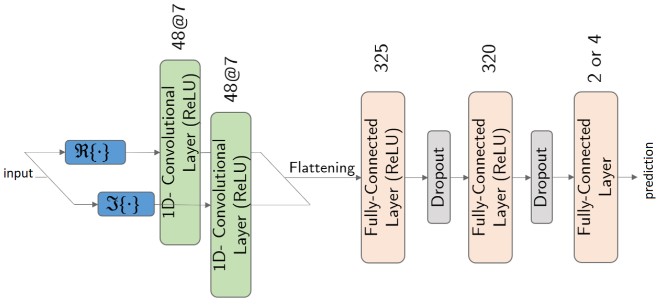

input
$\Re\{\cdot\}$
$\Im\{\cdot\}$
48@7
1D- Convolutional Layer (ReLU)
48@7
1D- Convolutional Layer (ReLU)
Flattening
325
Fully-Connected Layer (ReLU)
Dropout
320
Fully-Connected Layer (ReLU)
Dropout
2 or 4
Fully-Connected Layer
prediction

Fig. 1. Architecture of CNN proposed for packet detection.

Performance Metrics: For the system at study, we are interested in evaluating the ability of the receiver to detect the start of an incoming packet, considering either a correlator based approach or a machine-learning solution. In both cases, the detection algorithm is fed with a sequence of n_p complex symbols, and outputs a prediction on whether the sequence corresponds to a preamble or not. Let us denote as s_i the event of the i-th input sequence indeed being a preamble, and by $\bar{\mathsf{s}}_i$ its complement. Similarly, indicate by o_i the event of the algorithm evaluating the i-th input as a preamble, and by $\bar{\mathsf{o}}_i$ the complementary event. Leaning on this notation, we define the *detection probability* as

$$\mathsf{P}_\mathsf{d} = \frac{\sum_i \mathbb{1}\{\mathsf{s}_i, \mathsf{o}_i\}}{\sum_i \mathbb{1}\{\mathsf{s}_i\}} \tag{4}$$

where $\mathbb{1}\{\cdot\}$ is the indicator function. The quantity captures the fraction of correctly detected preambles (true positives) over the whole set of inputs which contained a preamble (true positives and false negatives). On the other hand, we also gauge the *false alarm probability*, computed as

$$\mathsf{P}_\mathsf{fa} = \frac{\sum_i \mathbb{1}\{\bar{\mathsf{s}}_i, \mathsf{o}_i\}}{\sum_i \mathbb{1}\{\bar{\mathsf{s}}_i\}}$$

and capturing the fraction of times in which the algorithm erroneously notifies of a preamble (false positives) over all the cases in which the input does not contain a preamble (false positives and true negatives) [5].

To visualize and analyze the trade-offs between these metrics, receiver operating characteristic (ROC) curves will be used. In this respect, a compact way to quantitatively compare different algorithms is through the *area under the ROC curve*, denoted as AUC metric. Through a single scalar value, between 0.5 and 1, the AUC captures the ability to balance detection and false alarm probabilities, with higher values indicating a better-performing algorithm, i.e., closer to the ideal ROC curve (AUC equal to 1) [11].

III. Preamble Detection Procedures

A. Bank of Correlators

As reference benchmark for the performance of detection schemes, we consider a correlation-based approach. The purpose of the correlator is to compare the incoming signal with the known preamble to identify a match. This is achieved by computing the inner product between the two sequences of n_P symbols, which outputs a peak indicating the start of a packet. In other words, the output of the correlator is compared with a predefined threshold γ, and the start of a preamble is declared each time that the correlation exceeds this threshold [2].

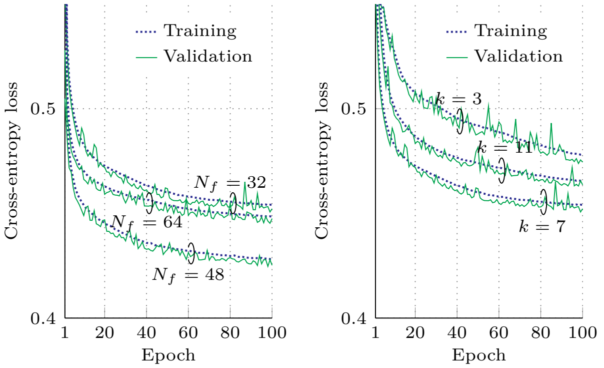


Fig. 2. Learning curves with the training and validation losses as function of epochs for different being the number of features N_f and kernel size k.

To improve performance in the presence of uncompensated Doppler shift, we consider a bank of M parallel correlators, each operating at a different center frequency. Specifically, denoting by $f_\mathsf{D}^{\max}$ the maximum Doppler shift experienced by users in the beam, the center frequency f_i of each correlator is given by

$$f_i = -f_\mathsf{D}^{\max} + i\,\frac{2\,f_\mathsf{D}^{\max}}{M-1} \qquad \text{with } i \in \{0, ..., M-1\}.$$

The start of a packet is declared when one of them outputs a correlation value above γ.

B. Convolutional Neural Network

Preamble detection can be formulated as an ML classification problem [4] where the goal is to identify whether a set of nP incoming symbols represents the start of a packet. In this work, a supervised approach is adopted, which involves training a neural network using a labeled dataset. Each training sample consists of a fragment of an input signal paired with a label indicating its corresponding class. During the training, the network learns to extract features that distinguish among different signal classes. Once trained, it can perform online inference to determine whether an incoming stream corresponds to the start of a packet.

To implement this, we propose a CNN architecture inspired by the promising results reported in [5], [6] for terrestrial scenarios. The structure of the CNN is shown in Fig. 1 and consists of two 1D convolutional layers, each with 48 filters of dimension 7, designed to extract features separately for the real and imaginary components of the signal. These parameters were determined after analyzing various combinations to refine the configuration and obtain the best results. This architecture is intentionally compact to adhere to LEO resource constraints. The convolutional layers are followed by two fully connected layers with 325 and 320 neurons, respectively, and include dropout regularization to promote robust learning and reduce overfitting. All layers utilize a ReLu activation function, which introduces non-linearity and accelerates convergence during training. The final layer consists of four neurons, each corresponding to one of the four classification classes,i.e. (i) no preamble start (np), (ii) preamble start with no interference (p), (iii) preamble start with exactly one interferer (p+1), and (iv) preamble start with multiple interferers (p+m). In turn, the probability of detection in (4) results from summing, in the numerator, all the events with (p), (p+1) and (p+m). The process used to generate the training dataset will be detailed in the next section.

The effectiveness of the CNN is primarily influenced by two key parameters: the kernel size k, and the number of feature maps N_f. The kernel size defines the portion of the input the layer processes at a time, determining how much local information is captured. In contrast, the number of features controls the range of patterns that the network can learn. Taking this into account, the network was trained using the cross-entropy loss function [12] and optimized with the Adam optimizer [13], testing different values of the kernel size and number of feature maps. The impact of these parameters on the learning process is illustrated in Fig. 2, which shows the learning curves for different architectural configurations. The most favorable performance was achieved with $N_f = 48$ and kernel size $k = 7$, which will be used in the rest of our study.

In the next section, we will report results in terms of ROC curves. For the CNN, these are obtained by comparing the sum of the probabilities given by neurons in the output layer which denote the presence of a preamble against a threshold. A detection is thus declared when the threshold is exceeded. By varying the threshold, the trade-off between false alarm and correct detection can be explored.

IV. Numerical Results and Discussion

A. Dataset Generation

The datasets used for training and testing were generated by simulating packet transmissions across a certain time window (also referred to as frame). Each dataset is populated with multiple frames, generated via simulation, based on the specific scenario under consideration. In each simulation run, packets are placed uniformly at random within the time window, and the noise is added accordingly. For each packet inserted, the two-dimensional sequence of $\mathsf{n_p}$ complex symbols containing the entire preamble is stored as a sample, along with a label indicating that it is the start of the packet. Additionally, each sample includes a feature representing the per-dimension power of the symbol sequence. This feature provides the model with a simple yet effective metric to differentiate signals from noise, thus enhancing its ability to recognize relevant patterns. The dataset is further populated with random sequences extracted from the transmission frame that do not include the start of a packet, labeled as non-preamble (np) starts. Frames are simulated until the desired number of samples is collected, maintaining a balanced distribution of 50% of preamble samples and 50% of non preamble samples to support effective model training and evaluation. In our results, we consider datasets of $2 \cdot 10^5$ samples.

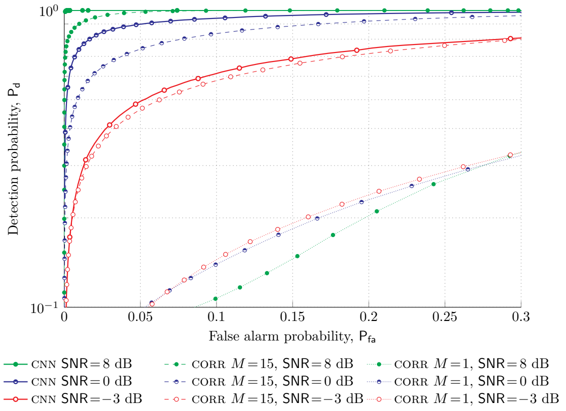
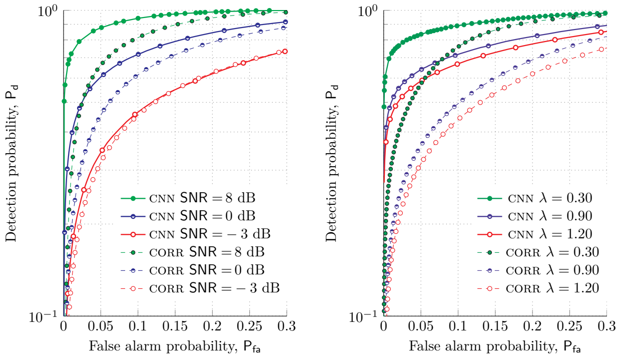


Fig. 3. ROC for single user scenario and channel occupancy 0.3.

B. Numerical Results

The superior detection performance and significant P_{fa} reduction justify the CNN's complexity, making its feasibility the primary demonstration. Computational optimization is relegated to future study. To gain preliminary insights on the performance of the different detection methods, we focus on a simple single-user setting. Specifically, each frame in the dataset generation contains packets from a given transmitter, randomly placed within the satellite coverage beam and affected by the corresponding uncompensated Doppler shift. To isolate the effect of the impairment, the SNR of the user was kept constant irrespective of its position. The ROC curves for this study are reported in Fig. 3, considering an overall channel occupancy of 0.3 [pkt/pkt duration], i.e., packets of the user are placed uniformly at random so that 30% of the frame duration is covered by its transmissions. In the plot, solid curves denote the performance of the CNN, whereas dashed lines refer to a single correlator centered at the carrier frequency, and to a bank of $M = 15$ correlators uniformly spaced to cover the whole available bandwidth, respectively. We show trends considering three different SNR values, representative of the conditions experienced at beam edge (-3 dB), close to beam center (8 dB), as well as of an intermediate situation (0 dB). In all cases, the CNN was specifically trained with samples generated with the same SNR used for testing.

The plot highlights two key take-aways. First, a single correlator cannot provide reasonable detection, with ROC curves that lie on the bisector of the $\mathsf{P_{fa}}$-$\mathsf{P_d}$ plane. The outcome stems from the very high Doppler shift that can be experienced by users, reaching up to half the bandwidth, and effectively pinpoints how challenging detection can become at LEO satellites in the presence of uncompensated frequency offsets. On the other hand, as expected, performance can be improved by resorting to a bank of correlators, at the expense of an increased complexity. Notably, the CNN consistently outperforms the correlation-based approach even for $M = 15$ in all considered conditions. Under the most challenging SNR of -3dB, both solutions struggle, with a detection probability



(a) traffic load $\lambda = 0.90$ (b) Distributed SNR

Fig. 4. ROC for multiuser scenario

of 80% attained only for a false alarm rate of roughly 25%. In this case, the improvement offered by the CNN solution is marginal. On the other hand, the gap increases significantly under more practical SNR: for 0 dB, a reduction in terms of false alarm rate of more than 25% if provided with respect to the correlators bank for a target $\mathsf{P_d}$ of 90%. Finally, at 8 dB both solutions offer a nearly ideal detection, with an AUC of 0.99965 for the CNN and of 0.99634 for the correlators.

The potential of ML-aided packet detection is also confirmed in a multi-user setting, as reported in Fig. 4a. In this case, we only report the performance of the CNN solution and of a bank of $M{=}15$ correlators for clarity. Let us define the load as the average number of aggregate transmitted packets in the system per packet duration. The results were obtained for a channel load $\lambda{=}0.9$ [pkt/pkt duration], corresponding to a random access channel operating in relatively congested conditions. As such, the study represents IoT systems where the satellite is covering an area populated by a large number of active devices. In such case packets can be randomly overlapping with each other (ALOHA-like access), and detection is further hindered by uncoordinated interference.

Once again, both solutions are challenged at SNR$=-3$ dB, facing a fundamentally noise-limited, rather than interference-limited setup. CNN is still outperforming the correlation-based solution, yet false alarm rates of around 25% have to be tolerated to reach a detection probability of 0.7. It must be noted however that such conditions are especially challenging, with preamble symbols affected by Doppler, interference and a received useful-signal power well below the noise level. More robust (e.g., longer) detection sequences might thus be targeted to improve performance at these SNR. More interestingly, a significant change can be observed with respect to the single user case when considering the 0 dB case, with the bank of correlators suffering a significant degradation. For instance, targeting a $\mathsf{P_{fa}} = 0.05$, a detection probability of 0.8 was attained in the absence of interference, whereas the value plummets in the multi-user setting to 0.45. The benefits offered by the CNN become in this case evident, with the ROC curve improving especially in the more appealing low $\mathsf{P_{fa}}$ region, e.g., with a 15% higher detection rate for $\mathsf{P_{fa}} = 0.05$. The trend becomes even more remarkable at high SNR (8 dB). In the single user case, the bank of correlators offers nearly ideal detection with relatively low false alarm. This feature is disrupted by interference, with $\mathsf{P_d} = 0.7$ for $\mathsf{P_{fa}} = 0.05$, and a reliability of at least 90% achieved only for $\mathsf{P_{fa}} > 0.15$ (area under the curve of ~ 0.94). The CNN approach reaches the same detection performance for a false alarm rate as low as 5%, significantly reducing the number of times further decoding blocks in the receiver are unnecessarily triggered, enabling substantial computational and energy savings.

TABLE II
Confusion matrices for multiuser scenarios with SNR = 8 dB for $\lambda = 0.30$ and $\lambda = 0.90$.

Convolutional Neural Network				
$\lambda = 0.30$	np	p	p+1	p+m
np	**0.9770**	0.0010	0.0172	0.0048
p	0	**0.9994**	0.0006	0
p+1	0.0519	0.0316	**0.9048**	0.0117
p+m	0.1232	0.0006	0.2114	**0.6648**
$\lambda = 0.90$	np	p	p+1	p+m
np	**0.9300**	0.0008	0.0271	0.0421
p	0	**0.9995**	0.0005	0
p+1	0.1067	0.0307	**0.8368**	0.0258
p+m	0.2209	0.0004	0.1070	**0.6717**

To complement these results, we report two confusion matrices obtained for the CNN solution, in Table II for $\lambda = 0.30$, and $\lambda = 0.90$, considering an SNR of 8 dB. In each case, the columns represent the predicted labels, while the rows correspond to the true labels. Recalling the structure of the output layer of the CNN described in Sec. III-B, the ML-based solution estimates whether the provided sample of $\mathsf{n_P}$ symbols does not contain a preamble (np), contains a preamble without interference (p), a preamble superposed to a single interferer (p+1), or a preamble experiencing more than one interferer (p+m). Accordingly, the confusion matrix allows to draw two further insights on the capabilities of the ML-based approach. On the one hand, it pinpoints how detection works based on the interference level. Note indeed that the value of $\mathsf{P_d}$ given the number of interferers (0, 1 or more than 1) is obtained by taking, for the corresponding row, the sum of the values in the three rightmost columns, as a correct detection flag is raised whenever any of these three decisions is made by the CNN. For both channel loads, the detection of an incoming packet not affected by interference is almost perfect, with $\mathsf{P_d}$ values above 0.999. Notably, excellent performance is also attained in the presence of a single interferer, with detection rates as high as 95% for $\lambda = 0.3$ and 90% for $\lambda = 0.9$. The slight degradation experienced in the latter case can be intuitively explained by the higher probability of having a larger portion of the preamble overlapped by the interfering signal as channel load increases. Detection becomes naturally more difficult when multiple interferers are superimposed to the incoming preamble. Nonetheless, detection of 88% and 78% can still be granted. In this respect, it is relevant to remark that such packets are also the ones which are less likely to be decoded even if detected, so that the degradation is somewhat less impactful on the overall system behavior. Along this line, the confusion matrices also

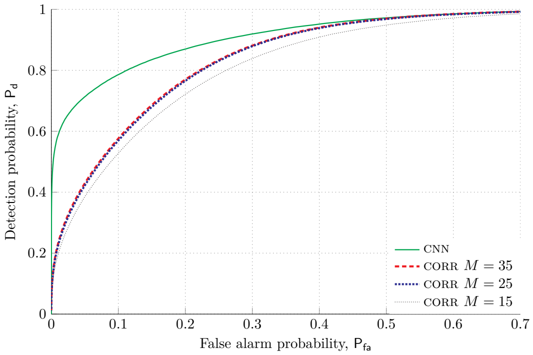


Fig. 5. ROC for general IoT-NTN scenario.

provide a second interesting take-away, as they reveal the ability of the ML-based solution to classify the interference conditions that characterize a detected packet. This feature can be particularly relevant, as it enables, for instance, the possibility for the receiver not to trigger the decoder in case of excessive distortions, or to prioritize decoding of packets with lower intereference level if successive interference cancellation is later to be performed. The high values found on the diagonal of the confusion matrices confirm the potential of the CNN in allowing such potential advantages.

Further insights on the behavior under interference are provided in Fig. 4b. Here, we remove the simplification of having all users received with the same power, and consider instead that the position of the user determines not only its Doppler shift, but also its SNR. Specifically, whenever a user sends a packet, its location is chosen uniformly at random within the coverage beam, and its path loss is computed via (2).[2] This step leads to a more practical setting, but also poses an additional challenge for the ML-based approach, as training and testing are performed with packets that have an additional degree of freedom on top of what seen so far, being received with different and unknown power levels. In Fig. 4b we show the ROC curves for a bank of $M = 15$ correlators and for the CNN considering three channel loads, ranging from a lightly loaded to a highly congested setting. The plot highlights that, at high loads, the traditional correlation solution may no longer be practical, failing to provide a detection of at least 80% even for false alarm rate of 30%. Conversely, the CNN proves to be much more robust, and also in this case is more capable of improving performance especially in the leftmost region of the ROC, starting to provide decent detection already for low false alarm even in challenging conditions. For example, more than 70% detection rate is achieved for P_{fa} as opposed to the 45% of the correlator case. A similar trend can also be observed in lightly loaded channel conditions ($\lambda = 0.3$), with a remarkable gap between the two solutions (AOC of 0.967 for the CNN and 0.928 with correlators).

To conclude our discussion, we tackle a final setting in which the network is trained using both multi-user and single-user data across different traffic load values to achieve generalized performance. This setup is of particular practical relevance, as it is representative of how a system may be trained and operated in a real scenario, with the LEO satellite passing over different areas, characterized by distinct (and not known a priori) levels of traffic. The results of the study are shown in Fig. 5, where the CNN is compared with banks of correlators of different sizes, specifically $M = 15$, 25, and 35. Also in this case, the generalized CNN model demonstrates a clear advantage, particularly at operational points with low false alarm probabilities, achieving an improvement in detection probability of more than 20%. In turn, the benefits of adding more correlators becomes progressively marginal.

V. Conclusions

In this work, we proposed a CNN to address the challenge of short packet detection in IoT networks. The model was evaluated in both single-user and multi-user scenarios, taking into account practical channel impairments such as phase noise and Doppler shifts. Results demonstrate the CNN's robustness and superior detection capabilities compared to traditional methods. By reliably identifying packet presence even under interference and harsh channel conditions, the proposed approach contributes to enhancing uplink performance and scalability in future IoT-NTN environments. The results also validate the hypothesis that learning-based detectors can adapt to distortions and Doppler effects that are difficult to analytically model or compensate.

References

[1] N. Abramson, "The throughput of packet broadcasting channels," *IEEE Trans. Commun.*, vol. 25, no. 1, pp. 117–128, 1977.

[2] P. Robertson, "Optimal frame synchronization for continuous and packet data transmission," 1995.

[3] E. Recayte and A. Munari, "On the performance of correlation-based packet detection techniques," *Physical Communication*, vol. 59, p. 102098, 2023.

[4] E. Recayte, A. Munari, and F. Clazzer, "Grant-free access: Machine learning for detection of short packets," in *IEEE 10th ASMS/SPSC*, 2020.

[5] 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, 2024.

[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, 2023.

[7] D. H. Yang and S. W. Choi, "On the structured design for efficient machine learning based PRACH preamble detection," in *2023 14th Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, 2023.

[8] D. Magrin, C. Pielli, C. Stefanović, and M. Zorzi, "Enabling LTE RACH collision multiplicity detection via machine learning," in *2019 Int. Symp. Model. Optim. Mob., Ad Hoc, Wireless Netw. (WiOPT)*, 2019, pp. 1–8.

[9] N. Modina, R. Ferrari, and M. Magarini, "A machine learning-based design of PRACH receiver in 5G," *Procedia Comput. Sci.*, vol. 151, pp. 1100–1107, 2019.

[10] C. Amatetti, R. Campana, A. Georganaki, and A. Vanelli-Coralli, "Neural network based non orthogonal random access for 6G NTN-IoT," in *2022 IEEE Globecom Wkshps*, 2022, pp. 1389–1394.

[11] K. E. Atkinson, *An Introduction to Numerical Analysis*, 2nd ed. New York: John Wiley & Sons, Inc., 1989.

[12] O. Simeone, "A brief introduction to machine learning for engineers," 2018. [Online]. Available: https://arxiv.org/abs/1709.02840

[13] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017. [Online]. Available: https://arxiv.org/abs/1412.6980

[2] In the calculation, we account for the impact of user location on transmission and receive antenna gains. For a given position, the angle of view between satellite and user is determined, and a linear interpolation of the gains between the maximum and minimum values reported in Tab. I is performed.