Alma Mater Studiorum - University of Bologna

Department of Electrical, Electronic, and Information Engineering "Guglielmo Marconi" - DEI

Second Cycle Degree in
Telecommunications Engineering

Master Thesis in
Neural Network based Non Orthogonal Random Access for 6G NTN-IoT

Author:
Ali Georganaki

Supervisor:
Prof. Ing. **Alessandro Vanelli Coralli**
Co-supervisor:
Dr. **Carla Amatetti**
Dr. **Riccardo Campana**

Academic Year 2020 /2022
Graduation Session Dec 5th, 2022

May Science Be Our Prime Guide.

Contents

3GPP The 3^{rd} Generation Partnership Project

3G the third generation of wireless mobile telecommunications technology

MIMO Multiple Input Multiple Output

M2M Machine-to-Machine

H2H Human-to-Human

IoT Internet of Things

NB-IoT Narrowband Internet of Things

NR New Radio

eMBB enhanced Mobile BroadBand

URLLC Ultra Reliable Low Latency Communications

mMTC massive Machine Type Communications

LPWA Low Power Wide Area

AI artificial intelligence

ML Machine Learning

DL Deep Learning

FL Federated Learning

CNN Convolutional Neural Networks

NGSO Non-geostationary Earth Orbit

GSO GEO Synchronous Orbit

DL Downlink

UL Uplink

gNB gNodeB

BS Base Station

ToA Time of Arrival

SCS sub-carrier spacing

CP cyclic prefix

OFDM orthogonal frequency-division multiplexing

BW Bandwidth

TA Timing Advance

TAC Timing Advance Command

MAC Medium Access Control

BWP bandwidth part

RRC radio resource control

RF radio frequency

SI system information

NPRACH narrowband physical random access channel

NPDCCH narrowband physical downlink control channel

NPBCH narrowband physical broadcast channel

NPUSCH narrowband physical uplink shared channel

NPSS narrowband primary synchronization signal

NSSS narrowband secondary synchronization signal

RAR random access response

DFT Discrete Fourier Transform

MSE Mean Square Error

ADAM adaptive momentum

RMSE Root Mean Square Error

NRS narrowband reference signal

RAPID Random Access Preamble Identifier

SC-FDMA single-carrier frequency-division multiple access

FDD frequency division duplexing

TDD time division duplexing

SG symbol group

SG-S SG-sum

CFO carrier frequency offset

NOMA Non-Orthogonal Multiple Access

vLEO very Low Earth Orbit

HEO High Elliptical Orbit

SatCom satellite communication

RedCap Reduced Capability

LPWAN Low Power Wide Area Network

PAPR peak-to-average power ratio

RAO random access occasion

ISI inter-symbol interference

GNSS Global Navigation Satellite Systems

UE User Equipment

RA Random Access

RACH Random Access Channel

NPRACH Narrowband Physical Random Access Channel

GW Gateway

GEO Geostationary Earth Orbit

MEO Medium Earth Orbit

LEO Low Earth Orbit

PHY physical layer

NR New Radio

NTN Non-terrestrial Network

ML Machine Learning

UAS Unmanned Aircraft System

UAV Unmanned Aerial Vehicle

HAPS High-altitude Platform Systems

ISL inter-satellite links

NN Neural Networks

LTE Long-Term Evolution

NN Neural Networks

RTD Round-Trip Delay

PHY Physical Layer

RL Reinforcement Learning

KL Kullback-Leibler

ReLu rectified linear activation function

BP back propagation

MLP Multi-layer Perceptron

LoS Line of Sight

NLoS Non Line of Sight

CNR Carrier to Noise Ratio

CL clutter loss

SDT small data transmission

RTD round-trip delay

List of Figures

List of Tables

Introduction

The being-altering 21st century has evolved the human race towards sustainable living habits. It has schooled engineers whose design objectives are not just to meet the needs of the present but also to verify that it will not compromise the ability of future generations to meet their own needs. Henceforward, More powerful and faster technology is not the prime target of a modern design and Telecommunication Engineering is no exception amongst disciplines. Future generations of wireless communications will focus on adaptations of a more sociotechnical design into the system and its applications.

Human-to-Human (H2H) communication has been the main driving force for the development of communications technologies and was addressed by the third generation of wireless mobile telecommunications technology (3G) and below. Ever since millions of distant people are end-to-end real-time connected with voice calling capabilities of 2G, along with data transfer, video calling, and mobile internet capabilities of 3G. The objective of H2H communications has been met with excellency by LTE, and LTE-Advanced embodying 4G, and their vital technologies, *e.g.*, Multiple Input Multiple Output (MIMO) as well as orthogonal frequency-division multiplexing (OFDM), delivering higher speed, higher quality, and higher capacity, with lower cost per bit to users.

Machine-to-Machine (M2M) communications was addressed in 4G, along with mobile and internet, to support the need for a shift from human-centric to machine-centric communications with strongly evolved requirements of connectivity everywhere and every time. The Internet of Things (IoT) technology has come forth with the basic premise to have smart devices collaborate directly without human involvement and deliver a new class of applications by connecting physical objects together in support of intelligent and coordinated decision making. The IoT market is projected to grow from USD 478.36 billion in 2022 to USD 2,465.26 billion by 2029; based on end-use industry, the market is shared in 2022 with healthcare, manufac-

turing, agriculture, BFSI, transportation, sustainable energy, IT & Telecom, retail, government, and others [1].

The 3^{rd} Generation Partnership Project (3GPP) standards body has defined a set of "flexible" (supremely-doped with complexity!) specifications as 5G New Radio (NR); addressing different use cases and their own particular requirements. 5G Core Network includes: enhanced Mobile BroadBand (eMBB), massive Machine Type Communications (mMTC), and Ultra Reliable Low Latency Communications (URLLC). eMBB provides greater data-bandwidth complemented by moderate latency improvements; this will help to develop today's mobile broadband use cases (*e.g.*, Augmented and Virtual Realities, 360-degree video streaming). mMTC has been already developed as part of 3GPP Release 13/14 Low Power Wide Area (LPWA) technologies, which includes Narrowband Internet of Things (NB-IoT). Thus this technology has inherent advantages such as global coverage, an integrated application platform, interference mitigation, massive connectivity, and security. URLLC requiring more bandwidth will need the 5G Core deployment for full end-to-end latency reduction for mission critical applications that are especially latency-sensitive *e.g.*, remotely-controlled automated surgery procedures, or a subtle Quartet in the Victoria Hall with the piano, the saxophone, the cello, and the drums each in a separate continent!

6G, a self-contained ecosystem with flexible management, control, and automated human-like decision-making processes, is expected to deliver efficiency clearly superior to 5G and satisfy evolving services and applications. 6G will link everything to enable full-vertical applications; provide full-dimensional wireless coverage, and combine all features, such as sensing, connectivity, computing, caching, monitoring, positioning, radar, navigation, and imaging. It will build on top of the current human-centric network architecture engaging any type of connectivity and device. With substantial growth in satellite communication (SatCom), and Non-terrestrial Network (NTN) in general, a great deal of interest and research went towards a unified architecture: a unified 6G, where Terrestrial and NTN communication infrastructure are to co-exist *i.e.*, utilizing dynamic spectrum management and sharing between the two the segments as well as among different layers of the architecture, *e.g.*, Geostationary Earth Orbit (GEO) and Non-geostationary Earth Orbit (NGSO) nodes, space- and air-borne nodes; that can be therefore orchestrated to provide cost-efficient network configuration by dynamically moving functionality creating a flexible network topology.

Pervasive and distributed Internet of Things (IoT) devices demand ubiquitous coverage beyond No-man's land. To satisfy plethora of IoT devices with resilient connectivity, Non-Terrestrial Networks (NTN) will be pivotal to assist and complement terrestrial systems. In a massiveMTC scenario over NTN, characterized by sporadic uplink data reports, all the terminals within a satellite beam shall be served during the short visibility window of the flying platform, thus generating congestion due to simultaneous access attempts of IoT devices on the same radio resource. The more terminals collide, the more average-time it takes to complete an access which is due to the decreased number of successful attempts caused by Back-off commands of legacy methods. A possible countermeasure is represented by Non-Orthogonal Multiple Access scheme, which requires the knowledge of the number of superimposed NPRACH preambles. This work addresses this problem by proposing a Neural Network (NN) algorithm to cope with the uncoordinated random access performed by a prodigious number of Narrowband-IoT devices. Our proposed method classifies the number of colliding users, and for each estimates the Time of Arrival (ToA). The performance assessment, under Line of Sight (LoS) and Non-LoS conditions in sub-urban environments with two different satellite configurations, shows significant benefits of the proposed NN algorithm with respect to traditional methods for the ToA estimation.

Chapter I

NB-IoT Standard

In an IoT perspective, every domain specific application is interacting with domain independent services, whereas in each domain sensors and actuators communicate directly with each other. The IoT elements consist of: *i)* Identification (to name and match services with their demand), *ii)* Sensing (gathering data from related objects within the network and sending it back to a data warehouse, database, or cloud), *iii)* Communication (connecting heterogeneous objects together), *iv)* Computation (hardware, software, and cloud platforms), *v)* Services (Identity-related Services, Information Aggregation Services, Collaborative-aware Services and Ubiquitous Services), and *vi)* Semantics (the ability to extract knowledge smartly)[2].

NB-IoT is one of Low Power Wide Area Network (LPWAN) technology developed by the 3GPP to enable a wide range of cellular devices and services. IoT Technology is well known and has become diversified. Now several types of internet of small things appeared following IoT. Typical small IoT technologies are LoRa, Sigfox, LTE-M and NB-IoT. Currently LoRa and Sigfox have made their own business models and compared to those, NB-IoT is staggered in business expansion. Moveover, NR Reduced Capability (RedCap) device has appeared in 5G and 5GB networks (3GPP Rel. 16 and above), and will address the wide range of applications termed Mid-Speed Smart IoT with connectivity demands and services requirements higher than LPWAN (LTE-M/NB-IoT) but do not require URLLC and eMBB.

NB-IoT User Equipment (UE) operates in the Downlink (DL) using 12 subcarriers with a subcarrier Bandwidth (BW) of 15kHz, and up to the maximum BW of 180 kHz with all permissible smaller resource allocations. In the Uplink (UL) using a single subcarrier with a subcarrier BW of either 3.75kHz or 15kHz or alternatively 3, 6 or 12 subcarriers with a subcarrier BW of 15kHz (maximum UL BW of also 180

kHz corresponding to 12×15kHz) [3]. The radio frame structure type 1 is only applicable to frequency division duplexing (FDD) (for both full duplex and half duplex operation) and has a duration of 10ms and consists of 20 slots with a slot duration of 0.5ms. Two adjacent slots form one subframe of length 1ms. When the subcarrier bandwidth is 15kHz, a slot can be further subdivided into three subslots of length 2 or 3 OFDM or single-carrier frequency-division multiple access (SC-FDMA) symbols for reduced latency operation. The radio frame structure type 2 is only applicable to time division duplexing (TDD) and consists of two half-frames with a duration of 5ms each [4]. The scope of this work only considers FDD mode which is preferable for NTN, as described in [5]. This is because TDD mode requires a guard time that directly depends on the propagation delay between UE and satellite to prevent UE from transmitting and receiving simultaneously. However, such a guard time might be excessive in NTN and would lead to a very inefficient radio interface [6].

I.1 Random Access Procedure

Digital communications, whether wired or wireless, require precise synchronization in time and frequency between the transmitter and the receiver. This is the most important precondition that needs to be satisfied upon connection request in both DL and UL. The synchronization process in the DL is pretty straight forward; when a UE wakes up, it begins listening to specific signals broadcast with a known frequency and periodicity in time by the covering Base Station (BS)s of its area, named gNodeB (gNB) in NR context. UE chooses the strongest reference signal, which belongs to the electromagnetically-closest gNB to itself (or the beam with the highest directivity), and synchronizes with the DL channel of the corresponding beam/cell. Where in frequency, when in time, and what gNB is conveying is understood by the UE – we are half way through this relationship. From a complexity point of view, DL synchronization is easy since everything about the gNB as a fixed BS (or the ephemeris information of the satellite in NTN case) is known a priori or is included in the received reference signals. However, UL synchronization is explicitly complex and essentially needs a number of handshakes *i.e.*, two-way communications, which is known as Random Access (RA) Procedure; Nonetheless, failure in synchronizations phase would fundamentally deprive users to access the network.

With the RA procedure, the UEs can start UL data transmission, achieve UL synchronization, and obtain a permanent ID to allow the rest of the communications

between the terminal and the network. Prior to sending a connection request to the network *i.e.*, the preamble, a UE has to learn where, when, and how to perform the initial access procedure *i.e.*, synchronize with the UL. These information are signalled as narrowband reference signal (NRS) with a fixed periodicity, and the UE needs to listen before talking. To be able to listen properly, a UE acquires time and frequency synchronization with a cell and detects the narrowband physical layer Cell ID by a procedure known as cell search.

To facilitate cell search for NB-IoT devices the narrowband primary synchronization signal (NPSS) and the narrowband secondary synchronization signal (NSSS) are transmitted in the DL which are generated from a frequency-domain Zadoff-Chu sequence defined in [TS 36.211][4]. These signal blocks are packed with narrowband physical broadcast channel (NPBCH) carrying the system information (SI) and together are periodically broadcast by the cell/beam in fixed subframes depending on the frame structure type used. There are 504 unique narrowband physical layer cell identity N_{ID}^{Ncell} indicated by the NSSS. The UE can synchronizes in the UL after finding the strongest SS/PBCH (the packed NPSS, NSSS, and NPBCH) block and decoding the data carried in its NPBCH which is scrambled by an initialization value $c_{init} = N_{ID}^{Ncell}$. Narrowband Physical Random Access Channel (NPRACH) parameters (configuration) are obtained by the decoded NPBCH block and are informed to the Physical Layer (PHY) layer. From the PHY layer perspective, the L1 RA procedure encompasses the transmission of RA preamble and random access response (RAR). The remaining messages are scheduled for transmission by the higher layer on the shared data channel and are not considered part of the L1 random access procedure [7].

I.1.1 4-step RACH

The contention based RA procedure is made up by a handshaking procedure of four messages, in which the first two represent the actual RA phase while the latest two constitute the contention resolution phase. The legacy RA procedure between the UE and the BS is illustrated in fig. I.1.

The UE, after synchronizing in DL, begins its synchronization procedure in UL by transmitting the RA preamble, as **Msg1** through a dedicated bandwidth part (BWP) *i.e.*, NPRACH, based on the configuration broadcast by the network. The preamble is the first UL signal sent by a UE to establish connection with the gNB. Hence, it is designed to efficiently and reliably support attachment of several UEs

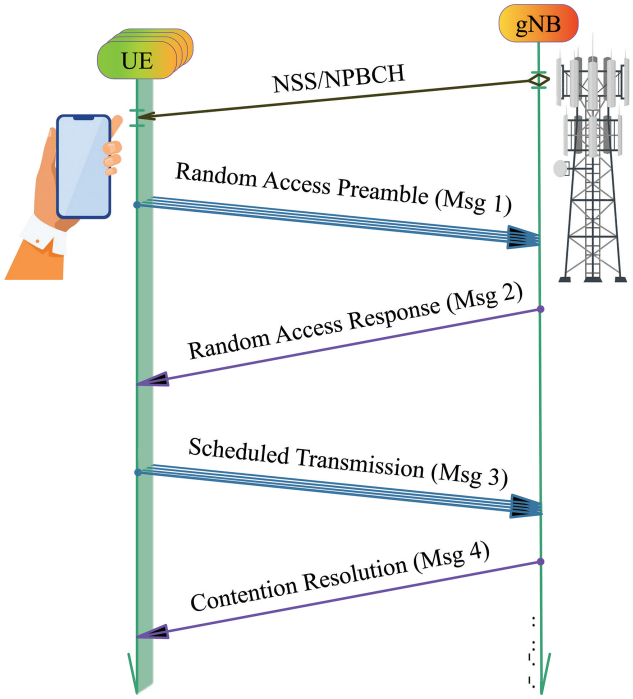


Figure I.1: General overview of 4-step RACH.

in a wide area but for a limited number of devices, *i.e.*, comparable to the number of resources defined for NB-IoT.

The BS, upon reception of the preamble, estimates the round-trip delay (RTD) of each user based on Time of Arrival (ToA) of the received preamble. In this way, the received waveforms from multiple terminals are aligned, both in time and frequency. After the preamble detection and the estimation of the synchronization parameters, the gNB transmits the RAR message as **Msg2** through the narrowband physical downlink control channel (NPDCCH) and allocates radio resources for Msg3. The RAR carries the Timing Advance (TA) command, UL grants, and the temporary identity of the UE (T-RNTI). The header of Msg2 is constituted by the Random Access Preamble Identifier (RAPID) which identifies the first random subcarrier used to send the preamble. Therefore, the user chooses the one that matches the first subcarrier of the preamble sent. It is worth highlighting that Msg1 is not user dependent and collisions will occur if the same preamble is chosen by more than one UE *i.e.*, they all will have to look for a RAR with the same RAPID header.

Upon reception of a successful RAR, and an UL grant, the UE through **Msg3**, replies by sending his own temporary identity (T-CRNTI) to the network using the scheduled resource. It transmits also the data volume status and power head-

room, to facilitate the BS scheduling and power allocation decision for the following transmissions [7].

In the last step of this handshake procedure, the network sends a contention resolution message as **Msg4** to solve the contentions due to multiple UEs transmitting the same random access preamble in the first step. *i.e.*, if more than one UE use the same Msg1, they will transmit the Msg3 on the same resources, since the RAR header is equal for all colliding preambles. Among the colliding devices, only one can conclude the procedure successfully; the remaining users re-attempt the procedures after a Back-off time indicated by their RAR.

I.1.2 2-step RACH

Conventional and existing RA Procedure consist of a 4-step handshake, used in both terrestrial and NTN scenarios. An alternatively proposed access protocol for terrestrial networks reduces the number of handshake steps to 2 by combining, in pairs, the 4 communicated messages between gNB and UE, known as 2-step Random Access Channel (RACH). Since one of the most challenging characteristics of an NTN channel is the introduced RTD due to the high altitude of the flying platform, 2-step RACH can be supremely beneficial as it performs the same task from half the conventionally required time to maximum the same time as of the parent 4-step. And despite the latency improvement, signal overhead and power consumption is also reduced.

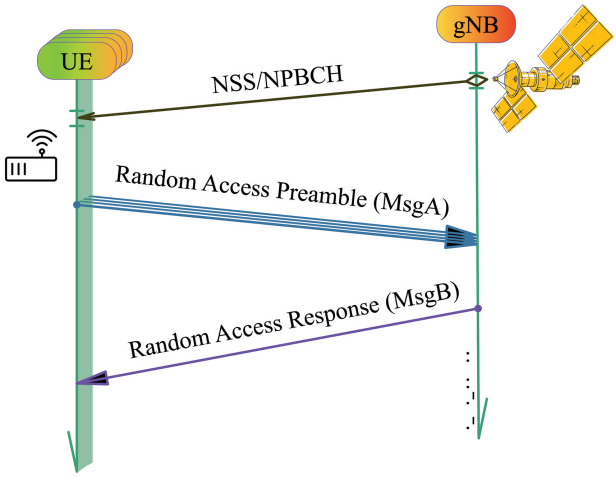


Figure I.2: 2-step RACH, NTN-IoT.

2-step RACH, as illustrated in fig. I.2, can be regarded as a simplified random

access process, where in the first step **MsgA** contains the Msg1 preamble and Msg3 payload, and in the second step **MsgB** contains the Msg2 and Msg4. So there is only one interaction between BS and UE before the establishment of radio resource control (RRC)_CONNECTED mode from RRC_IDLE of the terminal.

The **MsgA**, as illustrated on an example grid in fig. I.3, includes a preamble transmitted on NPRACH, and an additional payload transmitted in narrowband physical uplink shared channel (NPUSCH) which includes at least the equivalent content of Msg3 in 4-step RACH, and potentially also carrying some small data. The time and frequency resource of MsgA payload is preconfigured by the broadcast SI blocks known as random access occasion (RAO) and PUSCH occasion *i.e.*, again different UEs may choose the same access occasions, causing collision.

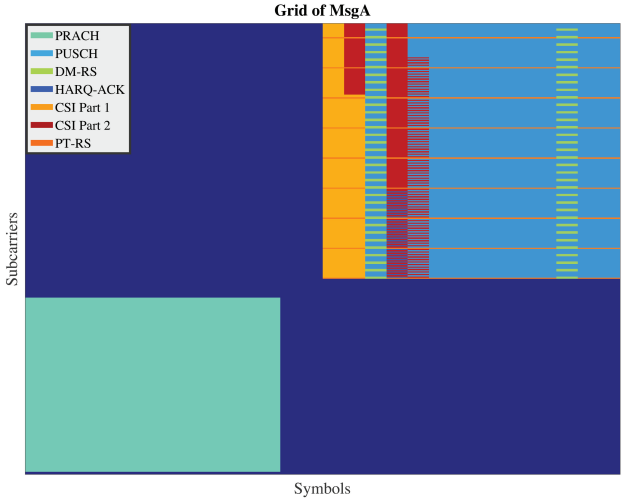


Figure I.3: MsgA: Preamble on a RAO + Payload on a PUSCH occasion

If MsgA preamble is correctly detected, BS will transmit **MsgB** which can be divided into Success RAR and Fallback RAR depending on whether MsgA NPUSCH is correctly decoded. Fallback RAR will instruct UE to fall back from 2-step RACH to 4-step RACH by re-transmitting Msg3 instead of MsgA. With this fallback mechanism, it can be guaranteed that the latency performance is no-worse than 4-step RACH.

I.2 NB-IoT Preamble

The NB-IoT is designed to accommodate a massive number of low-rate, low-cost, and delay-tolerant IoT devices. It adopts Long-Term Evolution (LTE)-like trans-

mission technology of SC-FDMA, with a BW defined as only 180 kHz for low-rate applications. To support extended coverage, massive connections, and long battery lifetime, the legacy LTE RA preamble based on Zadoff-Chu sequences is not appropriate due to its high peak-to-average power ratio (PAPR). This has resulted in the NPRACH being defined as a single-tone frequency-hopping preamble [8]. The use of single tone transmission is particularly interesting for satellite communications. Because of the required transmitted power, optimizing the efficiency of the power amplifier is necessary. The low PAPR of single tone transmission is therefore particularly adapted. However, there is no free lunch specially in such limited BW where the time duration of a packet could be very important, and hence the necessary energy to transmit the packet. With NB-IoT, increasing the bandwidth by using the multitone transmission is possible. Nevertheless, it comes at the price of an increase of the PAPR and therefore a loss on power amplifier efficiency.

I.2.1 Time and Frequency Structure

The physical layer random access preamble is based on single-subcarrier frequency-hopping symbol group (SG)s. A symbol group is illustrated in Figure fig. I.4, consisting of a cyclic prefix (CP) of length T_{CP} and a sequence of N identical symbols with total length T_{SEQ}. The CP is designed such that it is long enough to cover the maximum RTD to suppress inter-symbol interference (ISI). Recall that the RTD *i.e.*, twice the propagation time, thus twice the ToA, is to be estimated by the BS. And the value of N is kept small enough such that the effect of channel variation is negligible within $T_{CP} + N \cdot T_{SEQ}$ samples which altogether are defined as a *Symbol Group* [9]. There are three possible CP lengths specified for FDD mode, each defining a preamble format. CP lengths of 66.7μs (for format 0), 266.67μs (for format 1), and 800μs (for format 2) are designed to support cell radius of up to 10 km, 40 km, and 120 km, respectively. In terms of propagation delay, the three formats account for maximum RTDs of 66.67μs, 266.67μs, and 800μs, respectively [6]. The total number of time-contiguous SGs which shall be transmitted in a preamble repetition unit N_{rep}^{NPRACH} is denoted by P, and the number of time-contiguous SGs is given by G. The parameter values for frame structures 1 are listed in Table I.1[4]. Moreover, the basic time unit for LTE is defined as $T_s = 1/(\Delta f_{ref} \cdot N_{f,ref})$ with $\Delta f_{ref} = 15 \cdot 10^3$ Hz and $N_{f,ref} = 2048$ [10].

P SGs can be repeated up to $N_{rep} = 2^j, j \in \{0, 1, ..., 7\}$ times for coverage extension. Accordingly, the length of a preamble equals $P \cdot N_{rep}$ SGs. The NPRACH

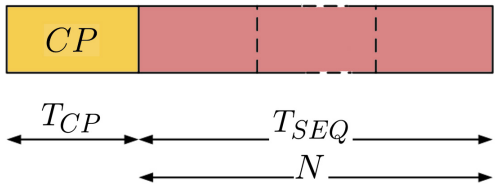


Figure I.4: Random access symbol group

Table I.1: Random access preamble parameters for frame structure type 1.

Preamble format	G	P	N	T_{CP}	T_{SEQ}
0	4	4	5	$2048T_s$	$5 \cdot 8192T_s$
1	4	4	5	$8192T_s$	$5 \cdot 8192T_s$
2	6	6	3	$24576T_s$	$3 \cdot 24576T_s$

transmission supports either a 3.75kHz or a 1.25kHz sub-carrier spacing (SCS) *i.e.*, Δf, with a fixed hopping pattern. SGs in preamble format 0 and 1 (with 3.75kHz SCS) hop by one or six subcarriers in frequency (equivalently 3.75kHz or 22.5kHz, respectively), whereas SGs in format 2 (with 1.25kHz SCS) hop by one, three, or eighteen subcarriers in frequency as defined in [4]. Since the hopping pattern is deterministic, several users choosing the same initial subcarrier will thus collide for the entirety of the NPRACH preamble sequence. The number of orthogonal preamble sequences is therefore the number of allocated NPRACH subcarriers.

The frequency location of each repetition of the NPRACH transmission is constrained within the size of narrowband RA resource $N_{sc}^{RA} = 12$ or for format2 $N_{sc}^{RA} = 36$ contiguous subcarriers. The frequency hopping scheme used within the N_{sc}^{RA} is configured with frequency location of the i^{th} symbol group as given by $n_{sc}^{RA}(i) = n_{start} + \tilde{n}_{sc}^{RA}(i)$ where $n_{start} = N_{scoffset}^{NPRACH} + \lfloor n_{init}/N_{sc}^{RA} \rfloor \cdot N_{sc}^{RA}$ with $N_{scoffset}^{NPRACH}$ frequency location of the first subcarrier allocated to NPRACH (*nprach-SubcarrierOffset*) and n_{init} being the subcarrier selected by the Medium Access Control (MAC) layer from $\{0, 1, \ldots, N_{sc}^{NPRACH} - 1\}$, and the quantity $\tilde{n}_{sc}^{RA}(i)$ (subcarrier occupied by NPRACH resource considered) is frame structure type dependent, and are detailed for also different P, and G in [4]. The channel hopping scheme used in this work is defined in specifications as frame structure type 1, preamble format 0 and 1, with G=4, and P=4, hopping as:

$$\tilde{n}_{sc}^{RA}(i) = \begin{cases} \left(\tilde{n}_{sc}^{RA}(0) + f(i/4)\right) \mod N_{sc}^{RA} & i \mod 4 = 0 \text{ and } i > 0 \\ \tilde{n}_{sc}^{RA}(i-1) + 1 & i \mod 4 = 1,3 \text{ and } \tilde{n}_{sc}^{RA}(i-1) \mod 2 = 0 \\ \tilde{n}_{sc}^{RA}(i-1) - 1 & i \mod 4 = 1,3 \text{ and } \tilde{n}_{sc}^{RA}(i-1) \mod 2 = 1 \\ \tilde{n}_{sc}^{RA}(i-1) + 6 & i \mod 4 = 2 \text{ and } \tilde{n}_{sc}^{RA}(i-1) < 6 \\ \tilde{n}_{sc}^{RA}(i-1) - 6 & i \mod 4 = 2 \text{ and } \tilde{n}_{sc}^{RA}(i-1) \geq 6 \end{cases}$$

$$f(t) = \left[f(t-1) + \left[\sum_{n=10t+1}^{10t+9} c(n) 2^{n-(10t+1)} \right] \mod (N_{sc}^{RA}) + 1 \right] \mod N_{sc}^{RA}, \quad f(-1) = 0$$

where $\tilde{n}_{sc}^{RA}(0) = (n_{init} \mod N_{sc}^{RA})$ is the frequency location of the i^{th} SG, and the m-sequence-based pseudo random sequence generator $c(n)$ is initialized with N_{ID}^{Ncell}. The above specification means there are two pattern of hopping 1 or 6 tones as also illustrated in fig. I.5. Within a repetition (N_{rep}) of P=4 SGs, each including a CP and N=5 symbols, the hopping distance is 1 between SGs at index 0 and 1, and between index 2 and 3. The hopping distance is always 6 between indices 1 and 2. Note that if preamble format1 is chosen, the CP size equals a symbol size of $8192T_s$ or 266.67μs (Table I.1), all together making a SG of length 1.6ms occupying 6 symbols in time transmitted in the same subcarrier, and repeated for $P \times N_{rep}$.

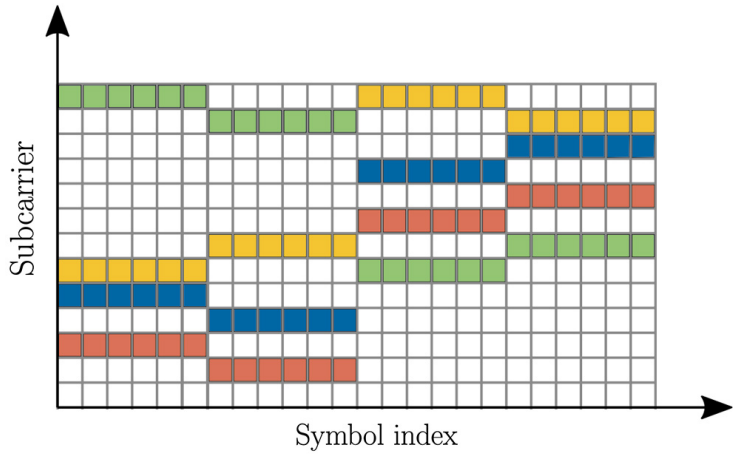


Figure I.5: Hopping pattern

Using the LTE-compatible sampling rate $f_s = 1.92$ M Sample/sec and a SCS of $\Delta f = 3.75kHz$, the Discrete Fourier Transform (DFT) length for preamble formats 0 and 1 is L = 512 (*i.e.*, one symbol is L samples long in time; in frequency, a DFT

of 512 points is used to have a SCS of 3.75 kHz with that sampling frequency.), and the CP length is $L_{CP}^{format0} = 128$, and $L_{CP}^{format1} = 512$. Theoretically, up to 48 UEs can simultaneously send their NPRACH preambles within the NB-IoT bandwidth of 180 kHz. However, the IoT device might be limited to fewer number of starting points to choose from $N_{sc} \in \{12, 24, 36, 48\}$. The j^{th} UE is identified by its $n_{init}(j)$ parameter in the range $\{0, \ldots, N_{sc} - 1\}$, which is used to generate the preamble hopping pattern for consecutive single-tone SGs.

The channel hopping procedure aids in the estimation of ToA and also reduces inter- and intra-cell interference. The ToA should be estimated by the BS for successful UL signal decoding and it further enables device positioning. Error in the ToA estimation results in the user not being able to receive the RAR. ToA estimation therefore has a great impact on performance in NB-IoT.

Chapter II

Non-Terrestrial Networks

NTNs are one of the enabling technologies for the realization of the global 6G system [11]. The NTN component fully integrated into the 6G infrastructure is not only capable of integrating and extending terrestrial networks, both in densely populated and rural areas, but also providing increased resilience, improved sustainability, high spectrum availability, and greater flexibility. To this aim, the 6G NTN network leverages multi-dimensional, multi-layer architecture, consisting of both space and airborne flying nodes, and encompasses novel enablers, such as the artificial intelligence (AI) [12].

3GPP has started integrating NTN into the NR architecture from Rel. 17 and the NTN journey in the 5G ecosystem continues in Rel. 18 and beyond. At the same time, several scientific publications have addressed various aspects of this integration processes: [13] proposes an overview of the possible challenges; the effects of the high Doppler and delay induced by non-geostationary satellites and the possible countermeasures are elaborated in details in [14], [15]. And A thorough link budget analysis for NB-IoT applications over NTN, following the 3GPP specifications, is presented in [16].

II.1 Architecture

An NTN refers to a network, or segment of networks using radio frequency (RF) resources on board a satellite (or Unmanned Aircraft System (UAS) platform) as defined in [TS 38.821] [5]. Figure fig. II.1 depicts a heterogeneous architecture proposed in 2009 as integrated space infrastructure for global communication (ISICOM) [17]. An NTN architecture can be divided into i) space segment: a single or constel-

lation of flying platforms GEO Synchronous Orbit (GSO)/GEO, or NGSO from very large platform to nano-satellites; ii) ground segment: with Gateway (GW)s feeding the platforms and keeping the satellites connected to the core network for overall system management; iii) user terminals; being served by the beams generated by the antenna of the flying platform.

A satellite may implement either a transparent or a regenerative (with on board processing) payload. A transparent payload performs RF filtering, frequency conversion and amplification, resulting in an unchanged waveform signal repeated by the payload. A regenerative payload would also allow to implement gNB functionalities on-board the satellite, thus, more advanced system configurations, such as amplification as well as demodulation/decoding, switch and/or routing, coding/modulation, functional split and Inter-Satellite Links (ISL), are implemented at higher cost.

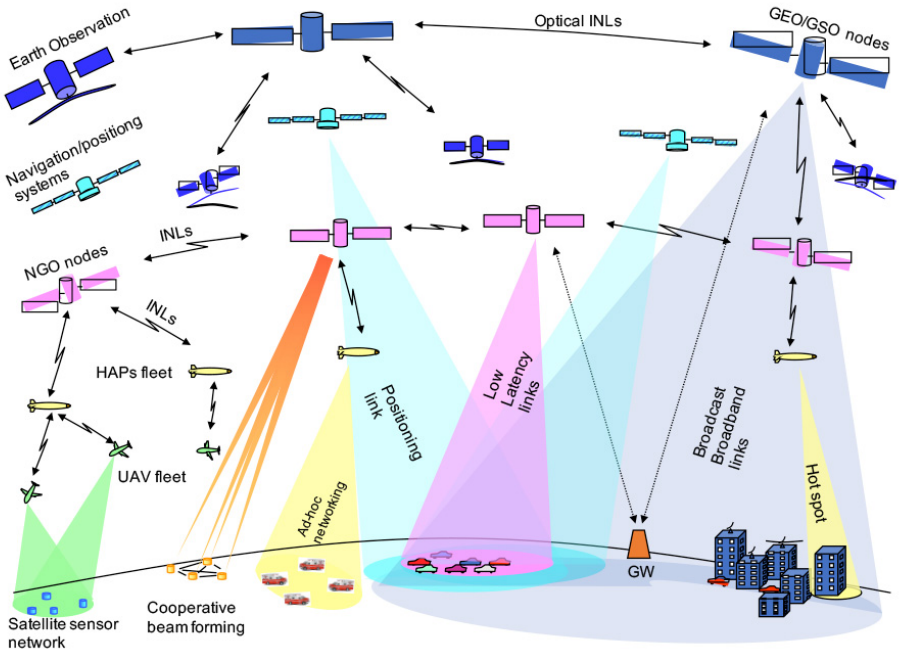


Figure II.1: NTN architecture; once a vision, now a reality. [17]

NTN platform types vary depending on the orbit and the altitude of the flying platforms. Low Earth Orbit (LEO), and Medium Earth Orbit (MEO) satellites circularly orbit the Earth from 600-1500 km and 7000-25000 km, respectively (very Low Earth Orbit (vLEO) satellites at 300-600 km are also defined and have gathered industry's interest); with beam footprint size ranging from 100 to 1000 km used to provide services in both Northern and Southern hemispheres with constellations

possible to provide global coverage including polar regions – given that appropriate orbit inclination, sufficient beams are considered, and inter-satellite links (ISL)s are provided. GEO satellites orbiting at an exact altitude of 35768 km above the earth with notional station keeping position fixed in terms of elevation/azimuth with respect to a given Earth point, with beam footprint size of 200-3500 km. A High Elliptical Orbit (HEO) satellite can elliptically orbit the Earth with a perigee of 400km altitude up to apogee of 50000 km altitude with a beam footprint size of 200-3500 km. Another type of platform defined as UAS (including High-altitude Platform Systems (HAPS)) fly at an altitude of 8-50 km having 5-200 km of typical beam footprint size.

Considered NTN reference scenarios in 3GPP specifications, including circularly and notional station keeping platforms, the highest impairment constraints (*e.g.*, RTD and Doppler), type of payload, and fixed or steerable beams, as given by Table II.1. In the latter, the on-board antenna keeps serving the same on-ground area while the satellite moves on its orbit (steerable antennas). In the fixed beam case, the served on-ground area is moving together with the satellite.

Table II.1: NTN reference scenarios

	Transparent Sat	Regenerative Sat
GEO based NTN	Scenario A	Scenario B
LEO based NTN: steerable beams	Scenario C1	Scenario D1
LEO based NTN: moving with the Sat beams	Scenario C2	Scenario D2

II.2 Channel Impairments

High speed of the platforms (more than 7 Km/sec for a LEO satellite, relative to earth) translates into Doppler, and the high altitude extends the propagation delay in order of mili-seconds (10s for LEO up to 100s for GEO satellites). In this section, typical satellite channel impairments such as large Doppler shifts and propagation delays are discussed.

Delay

Different types of delay are involved in satellite communications. Among them, the propagation delay is not only the predominant one, but it also reaches values

much larger than those of terrestrial networks, being a bottleneck for NR 5G/6G communications [13]. In satellite communication where distances are very large, such delay that yields a misalignment between UL and DL frames, as shown in fig. II.2 is critical to be coped with. The gNB calculates the TA needed for the UL frame i that needs to start advanced in time with respect to the corresponding DL frame number i for transmission from the UE. The initial Timing Advance Command (TAC) is signalled via RAR in Msg2. At the UE side T_{TA} is controlled by the MAC layer and implemented by the PHY layer to synchronize UL and DL frames at the gNB.

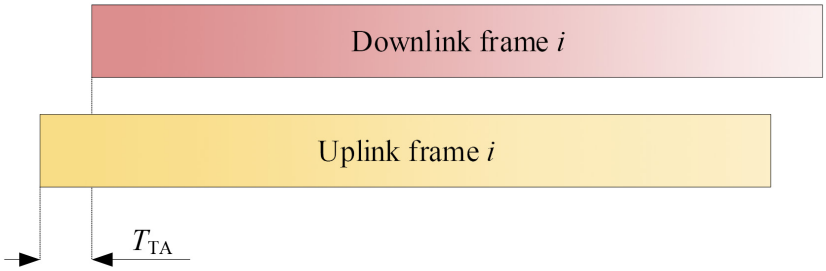


Figure II.2: UL-DL timing relation.

The RTD is approximated by twice the propagation delay between the transmitter and the receiver with negligible signal processing time in this context, and the propagation delay depends on the satellite orbit, type of payload, and elevation angle. In order to estimate the propagation delay in the considered scenarios, we further assume to be in a pessimistic scenario in which the transmitter and the receiver are not perfectly aligned and, thus, they have different elevation angles. The overall RTD can thus be computed as [13]:

$$RTD \approx 2 \times T_{1way} = 2\frac{d_{GW-SAT}(\theta_{GW}) + d_{SAT-Rx}(\theta_{Rx})}{c} \tag{II.1}$$

where T_{1way} is the one-way propagation delay, d_{GW-SAT} is the distance between the GW and the satellite as a function of its elevation angle θ_{GW}, and the distance between the satellite and the receiver is given by d_{SAT-Rx} as a function of its elevation angle θ_{Rx}, and c the speed of light (299792458 m/s).

The worst case NTN scenarios to be considered for the delay constraint are given by [TS 38.821] [5] and are summarized in the table II.2. The maximum and minimum propagation delay are given as their contribution to the RTD on the radio interface between the gNB and the UE. Also, the Maximum RTD variation as seen by the

UE measures how fast the RTD (function of UE-Sat-NTN GW distance) varies over time. The worst case considered for the calculations in the table is when the satellite moves towards/away from the UE at 10° elevation angle, and also assuming the UE speed higher than of a commercial jet aircraft cruise speed, 1200 km/h! However, general design specifications of on-ground receivers consider UE speed of maximum 500 km/h, as of a bullet train.

Table II.2: NTN reference scenarios vs. delay constraints

NTN scenarios	A	B	C1-C1	D1-D2
Relative speed of the Satellite	negligible		7.56 km/s	
Min elevation	10° for service link and 10° for feeder link			
Maximum RTD	541.46 ms (worst case)	270.73 ms	25.77 ms	12.89 ms
Minimum RTD	477.48 ms	238.74 ms	8 ms	4 ms
Maximum RTD variation	negligible		up to ±93.0 $\mu s/sec$ (worst case)	up to ±47.6 $\mu s/sec$

Considering UEs within a beam footprint, there is a **common delay** defined at the center of the beam experienced by every user in the same manner. Satellite ephemeris information broadcast in the concerned beam easily compensate most of this common delay and the offsets introduced depending on the prediction accuracy can be handled by protocols. As shown in fig. II.3, the UEs located elsewhere of the beam but the center, experience an additional delay known as the **differential delay**. Compensating this UE-specific delay via protocols needs further study, however in current 3GPP (up to Rel 18) specification Global Navigation Satellite Systems (GNSS) capability of the terminals are responsible for compensating such delay by means of a priori known location of the UE, at the price of the introduced complexity to the terminal which translates into power consumption.

It can be noticed, as expected, for the mMTC or particularly NB-IoT scenario, the propagation delay might be an issue for all procedures and steps, since it is several orders of magnitude above typical terrestrial networks delays. The impact of the delays shall be evaluated on a case-by-case basis also taking into account the type of communication that the considered procedure is requesting.

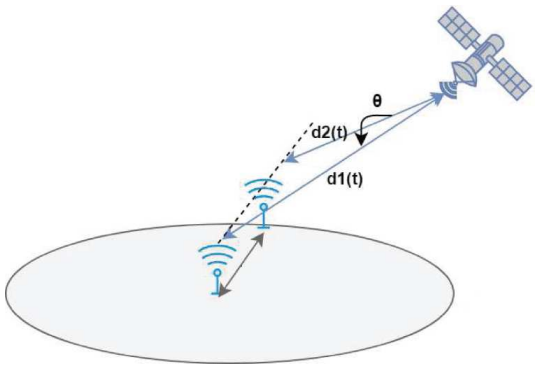


Figure II.3: Common and differential delay/Doppler. [14]

Doppler Shift

The Doppler shift consists in the change in the carrier frequency due to the relative motion between the satellite and the user terminal *i.e.*, carrier frequency offset (CFO). The Doppler shift that the UE experiences due to its relative motion to the relay node can be calculated as $f_d = (\nu \cdot f_c/c) \cdot \cos\theta$ [18], which in an NB-IoT sense can be assumed negligible due to the fixed position of the device. When considering a satellite communication channel, the Doppler shift only caused by the satellite movement on its orbit is provided by authors of [15] in a closed-form expression as a function of the satellite orbital velocity (relative to the user terminal) and the elevation angle:

$$f_d(t) = \frac{f_c \cdot \omega_{SAT} \cdot R_E \cdot \cos\left(\theta_{UE}(t)\right)}{c} \tag{II.2}$$

where $\omega_{SAT} = \sqrt{GM_E/(R_E + h_{SAT})^3}$ is the satellite orbital velocity, $\theta_{UE}(t)$ is the elevation angle of the UE at a fixed time, R_E the Earth radius, $G = 6.67 \cdot 10^{-11}\mathrm{Nm}^2/\mathrm{kg}^2$ the Gravitational constant, and $M_E = 5.98 \cdot 10^{24} kg$ is the Earth's mass. It shall be noticed that, in GEO systems the Doppler shift can be assumed to be negligible. On the other hand, in LEO satellites the Doppler shift can introduce significant frequency shifts with respect to those expected in terrestrial systems as seen in fig. II.4. This could deeply impact the frequency synchronisation of the resources used to transmit through the air interface. Furthermore, the received signals are impacted by a significant Doppler rate as well, thus leading to additional degradation of the detection performance in case of longer preamble/data packets.

The Doppler experienced by the i^{th} user in the DL channel and, vice-versa, on the satellite with respect to the i^{th} user in the UL channel can be described as $f_{Doppler} =$

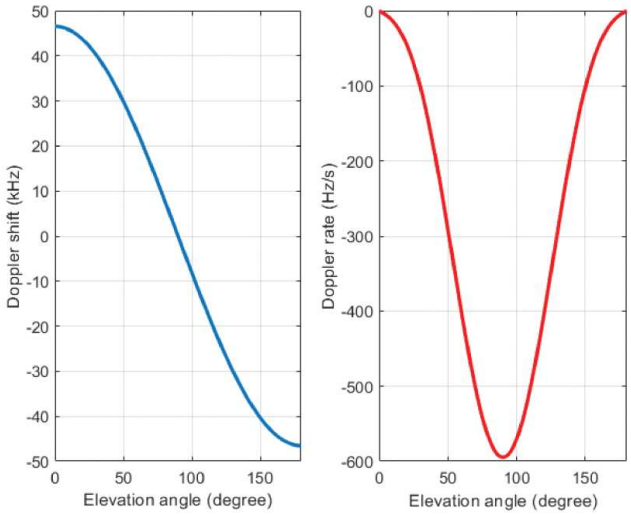


Figure II.4: Doppler variation depending on the elevation angle [6]

$f_{D_{common}} + \Delta f_{D_{diff}}$, where $f_{D_{common}}$ is the **common Doppler** experienced in the same way by all UEs within a beam footprint, while $\Delta f_{D_{diff}}$, the **differential Doppler**, depends on the relative positions of users in the footprint [13]. A UE with GNSS capability performs the pre-compensation of Doppler shifts by itself thanks to the knowledge of satellite ephemeris and available UE location. And for a UE without GNSS capability, the satellite performs the pre-compensation of the Doppler shift at the center of the beam on the ground and broadcasts the common Doppler to all UEs inside the concerned beam to be taken into account for the uplink transmission.

II.3 Random Access Adaptations

Pre-/Post- compensation techniques for such impairments are given by 3GPP specifications [5] and include broadcasting satellite ephemeris information as SI messages to aid the UE in compensating the common delay, Doppler shift, and variations in NTN. Moreover, the design of a NB-IoT receiver in the presence of Doppler effects at the GW side is addressed in [19]; the assessment and the evaluation of the RA procedure for a satellite-based NB-IoT system, with a particular emphasis on the effect of the large RTD typical of the NTN, is proposed in [20, 21, 22, 23]. However, only the last two works consider the congestion generated during the message flows of the RA procedure due to the short visibility window of the satellite.

Since IoT communications are mainly characterized by sporadic UL data report-

ing, non-continuous satellite coverage, provided by cost efficient incomplete constellations, is the baseline approach for most of the foreseen IoT-NTN architectures. Thus, all the terminals within the satellite beam, simultaneously competing to access the network, must be served within the short visibility period of the flying platform, generating congestion. For the moment being, 3GPP in its Rel. 18 recognizes the congestion has a problem [24], since it may cause the deterioration of system performance or even the unavailability of the service. However, solutions will be most likely proposed in Rel. 19, where more sophisticated architectures are considered also for IoT scenario. A possible counter measure to this issue is represented by the application of Non-Orthogonal Multiple Access (NOMA) schemes to the Msg3 of the RA protocol, [25]. However, in order to fully benefit from the advantages brought by NOMA, it is of paramount importance to identify and count the terminals transmitting on the same time/frequency resources.

Chapter III

Neural Network Algorithms

The areas of intervention in 6G key technologies include extensive use of AI and Machine Learning (ML) to facilitate innovative services. The processing of data is rather complex where AI and ML are expected to play a crucial role. The intertwining of communication and computation algorithms requires a communication/computation co-design that falls under the scope of in-network AI governance (incl. AI-based allocation and instantiation of network functions; management of collaborative AI components across the network; design of distributed AI mechanisms based on *e.g.*, Federated Learning (FL); or the deployment of AI-output/decision justification mechanisms leading to explainable AI). In 6G intelligent context-aware networks, deployment, operation, and energy usage will be minimized subject to the information gathered without human intervention, [11]. Despite the effective assistance AI/ML offers in communication systems, they cannot be treated as a black box without verifying the undergone process to train a usable model. This chapter is responsible for illuminating such black box by describing Neural Networks (NN) Algorithms to make sense of the offered AI/ML assisted solutions.

III.1 Artificial Intelligence and Machine Learning

AI encompasses several approaches and paradigms. ML, DL, Reinforcement Learning (RL) and their intersections are all parts of AI. Thus, a major part of AI follows the learning approach, although approaches without any learning aspects are also included. Overall, research into AI aims to make the machine smarter (*i.e.*, the ability to accomplish complex intellectual tasks normally necessitating a human), either by following some rules or by facilitating guided learning. The former refers

to symbolic AI, which implements sophisticated handcrafted commands as an extensive set of rules that encompasses the humans' expertise; the latter refers to ML which, in contrast to symbolic AI, requires a learning approach. Thus, rather than giving the rules to solve a problem, the machine is provided with the context to learn the rules by itself to solve the issue *i.e.*, accomplish a complex task such as classification, regression, clustering, detection, recognition, segmentation, planning, scheduling, or decision making.

An ML system is trained rather than programmed with explicit rules. The learning process requires data to extract patterns and hidden structures; the focus is on finding optimal representations of the data to get closer to the expected result by searching within a predefined space of possibilities using guidance from a feedback signal, where representations of the data refer to different ways to look at or encode the data. Supervised, unsupervised and semi-supervised learning are all ML approaches that can be employed to solve a broad variety of problems.

During supervised learning, all of the training data is labeled, i.e., tagged with the correct answer. The algorithm is thus fully supervised, as it can check its predictions are right or wrong at any point in the training process. The supervised model learns the patterns from the training data to then be able to predict labels for non-labeled data during inferencing. Supervised learning has been applied for classification and regression tasks. As labeling can be impossible due to a lack of information or infeasible due to high costs, unsupervised learning employs an unlabeled data set during training. Using unlabeled data, the model can extract hidden patterns or structures in the data that may be useful to understand a certain phenomenon or its output could be used as an input for other models. Unsupervised learning has been commonly used for clustering, anomaly detection, association and autoencoders. As a middle ground between supervised and unsupervised learning, semi-supervised learning allows a mixture of non-labelled and labaled portions of training data.

III.1.1 Supervised Learning

In a supervised learning setting, the learning algorithm experiences a data set containing examples and their respective labels or targets. An example will typically be denoted as x and its label, or target, as y. Together, we have training examples $(x, y) \in \mathbb{D}$ existing in our data set $\mathbb{D}$. In supervised learning problems, we attempt to learn to predict the label y from the example x, or equivalently, estimate

the conditional distribution $p(y|x)$. Taking this approach, we will want to obtain a model of this conditional distribution and we will denote the parameters of such a model as θ. Assuming a set of independent and identically distributed (i.i.d) data $\mathbb{D} = \{x_1, x_2, \ldots, x_n\}$ drawn from the true but unknown data-generating distribution $p_{data}(x)$. We let $p_{model}(x;\theta)$ be a parametric family of probability distributions over the same space indexed by θ. In other words, $p_{model}(x;\theta)$ maps any configuration x to a real number estimating the true probability $p_{data}(x;\theta)$. The maximum likelihood estimator for θ is then defined as [26]:

$$\theta_{ML} = \arg\max_{\theta} p_{model}(\mathbb{D};\theta) = \arg\max_{\theta} \prod_{i=0}^{n} p_{model}(x_i;\theta) \tag{III.1}$$

where P_{model} is a function space of probability distributions over the parameters θ. A convenient product into sum transform makes the above more computationally appealing, with equivalent optimization problem; we can also take the logarithm on both sides, which gives us:

$$\theta_{ML} = \arg\max_{\theta} \sum_{i=0}^{n} \log(p_{model}(x_i;\theta)) \tag{III.2}$$

Additionally, we can divide the right hand side of the equation by n, as rescaling the cost function does not change the arg max, and we obtain the expectation of the log-probability of the model over the empirical data-generating distribution $\hat{p}_{data}$; *i.e.*, we can quantify the amount of uncertainty in an entire probability distribution using the Shannon entropy which is the expected amount of information in an event drawn from that distribution:

$$\theta_{ML} = \arg\max_{\theta} \mathbb{E}_{x\sim\hat{p}_{data}} \log(p_{model}(x;\theta)) \tag{III.3}$$

One way to interpret maximum likelihood estimation is to view it as minimizing the dissimilarity between the empirical distribution $\hat{p}_{data}$, defined by the training set and the model distribution, with the degree of dissimilarity between the two measured by the Kullback-Leibler (KL) divergence. The KL divergence of these distributions is defined as:

$$D_{KL}(\hat{p}_{data}||p_{model}) = \mathbb{E}_{x\sim\hat{p}_{data}}[\log \hat{p}_{data}(x) - \log p_{model}(x)] \tag{III.4}$$

The term on the left is a function only of the data-generating process, not the model. This means when we train the model to minimize the KL divergence, we need only minimize:

$$-\mathbb{E}_{x\sim\hat{p}_{data}} \log(p_{model}(x)) \tag{III.5}$$

which is exactly equivalent to the maximization problem stated in the maximum likelihood formulation (III.3). The above is referred to as the negative log-likelihood of the model distribution and minimizing it results in the minimization of the cross-entropy between the data-generating distribution and the model distribution. The significance of this is two-fold. Firstly, the terms cross entropy and negative log-likelihood are often used in literature to describe the loss functions that are being used to evaluate a given ML model and the above minimization problem is what is being referred to. Secondly, this gives rise to the narrative that the model associated with the maximum likelihood estimate is, in fact, the same model that most closely resembles the empirical data distribution. This is important considering what we want our model to do, namely, produce correct labels or targets for data drawn from the data-generating distribution that the model has not seen before [27].

For completeness, the maximum likelihood estimator for the conditional distribution, which provides a label's probability for x_i i.i.d examples, is given as:

$$\theta_{ML} = \arg\max_{\theta} \sum_{i=0}^{n} \log(p_{model}(y_i|x_i;\theta)) \tag{III.6}$$

Often times, regularization on the parameters of the model is desirable, as regularization can lead to better generalization of the model. This is most frequently seen in the different types of neural network models. Building on the maximum likelihood perspective of the loss function, we can show that adding a regularization function to our optimization function can be seen as inducing a prior over the model parameters and subsequently changing our estimator to the maximum a posteriori (MAP) point estimate. Inducing a prior probability on the model parameter results in the following optimization problem [27],

$$\theta_{MAP} = \arg\max_{\theta} p(\theta|D) = \arg\max_{\theta} \log(p(D;\theta)) + \log(p(\theta)) \tag{III.7}$$

where, Bayes' Rule, the properties of logarithm, and the fact that the optimization problem does not depend on the data-generating distribution have been made use of.

III.1.2 Convolutional Neural Networks

Convolutional Neural Networks (CNN)s are a specialized kind of neural network for processing data that has a known grid-like topology. Examples include time-series data, which can be thought of as a 1D grid taking samples at regular time intervals, and image data, as an example shown in Fig. III.1 a 2D grid of pixels. In CNN terminology, the first argument of the convolution operation is often referred to as the **input**, and the second argument as **kernel**, which both in ML applications usually are a multidimensional array of data and parameters, respectively; called **tensors**. And the output is sometimes referred to as the **feature map**. The sparse weights in CNN is accomplished by making the kernel smaller than the input. This means that we need to store fewer parameters, which both reduces the memory requirements of the model and improves its statistical efficiency. It also means that computing the output requires fewer operations.

Neural Networks can be used for regression by predicting continuous values or for classification by predicting probabilities for each class. A typical layer of a convolutional network consists of three stages. In the first stage, the layer performs several convolutions in parallel to produce a set of linear activations. In the second stage, sometimes referred to as the **detector stage** each linear activation is run through a non-linear activation function, such as the rectified linear activation function (ReLu). A nonlinear activation function is generally chosen to add more complexity to the model by eliminating the linearities. In the third stage, a pooling function is used to modify the output of the layer further. A **pooling** function replaces the output of the net at a certain location with a summary statistic of the nearby outputs. For example, the **max-pooling** operation reports the maximum output within a rectangular neighborhood. Other popular pooling functions include the average of a rectangular neighborhood, the L^2 norm of a rectangular neighborhood, or a weighted average based on the distance from the central pixel. In all cases, pooling helps to make the representation approximately invariant to small translations of the input. Invariance to translation means that if we translate the input by a small amount, the values of most of the pooled outputs do not change. Invariance to local translation can be a useful property if we care more about whether a feature is present than exactly where it is [26].

Because pooling summarizes the responses over a whole neighborhood, it is possible to use fewer pooling units than detector units, by reporting summary statistics for pooling regions spaced k pixels apart rather than 1 pixel apart. This improves

the computational efficiency of the network because the next layer has roughly k times fewer inputs to process. When the number of parameters in the next layer is a function of its input size (such as when the next layer is fully connected and based on matrix multiplication), this reduction in the input size can also result in improved statistical efficiency and reduced memory requirements for storing the parameters.

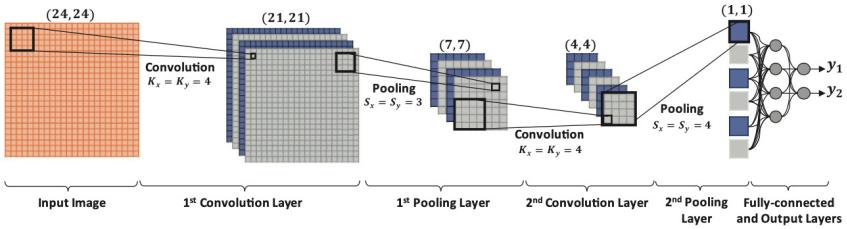


Figure III.1: The illustration of a sample CNN with 2 convolution and one fully-connected layers. [28]

Fig. III.1 illustrates the basic blocks of a sample 2D CNN configuration that classifies a 24 × 24-pixel grayscale image into two categories. This sample network consists of two convolution and two pooling layers with 4 and 6 neurons, respectively. The output of the last pooling layer is processed by a single fully-connected layer and followed by the output layer that produces the classification output. The interconnections feeding the convolutional layers are assigned by weighting filters ω having a kernel size of (K_x, K_y). The convolution takes place within the boundaries; therefore, the feature map dimension is reduced by the $(K_x - 1, K_y - 1)$ pixels from the width and height, respectively. The sub-sampling factors (S_x, S_y) are set in advance in the pooling layers. In the sample illustration above, the kernel sizes corresponding to the two convolution layers were set to $K_x = K_y = 4$, while the sub-sampling factors are set as $S_x = S_y = 3$ for the first pooling layer and $S_x = S_y = 4$ for the second one. Note that these values were deliberately selected so that the outputs of the last pooling layer (*i.e.*, the input to the fully-connected layer) are scalars (1x1). The output layer consists of two fully-connected neurons corresponding to the number of classes to which the image is categorized. The complete forward-propagation process steps of this example are described in [28].

CNNs are predominantly trained in a supervised manner by the so-called back propagation (BP) algorithm. During each iteration of the BP, the gradient magnitude (or sensitivity) of each network parameter such as the weights of the convolution and fully-connected layers is computed. The parameter sensitivities are then used to iteratively update the CNN parameters until a certain stopping criterion is achieved.

The training of Neural Networks is all about finding the best weights; to do so, a loss function is designed to compare the output of the model and the ground truth for each output, to find the weights that minimize that loss function. There are several gradient-descent optimization methods that can be used in BP such as Stochastic Gradient Descent, AdaGrad, RMSProp, and adaptive momentum (ADAM).

There is a significant difference in terms of computational complexities of 1D and 2D convolutions, *i.e.*, an image with $N \times N$ dimensions convolve with $K \times K$ kernel will have a computational complexity $\sim O(N^2K^2)$ while in the corresponding 1D convolution (with the same dimensions, N and K) this is $\sim O(NK)$. This means that under equivalent conditions (same configuration, network, and hyperparameters) the computational complexity of a 1D CNN is significantly lower than the 2D CNN. Also, a general observation shows, most of the 1D CNN applications have used compact (with 1-2 hidden CNN layers) configurations with networks having < 10 K parameters whereas almost all 2D CNN applications have used "deep" architectures with more than 1 M (usually above 10 M) parameters. Obviously, networks with shallow architectures are much easier to train and implement. As a result, training deep 2D CNNs requires a special hardware setup (*e.g.*, Cloud computing or GPU farms). On the other hand, any CPU implementation over a standard computer is feasible and relatively fast for training compact 1D CNNs with few hidden layers (*e.g.*2 or less) and neurons (*e.g.*< 50). Thus, compact 1D CNNs, due to their low computational requirements, are well-suited for real-time and low-cost applications, especially on mobile or hand-held devices.

An example of 1D CNN is illustrated in Fig III.2, two distinct layer types are proposed in such network: 1) the so-called "CNN-layers" where both 1D convolutions, activation function, and sub-sampling (pooling) occur, and 2) Fully-connected (dense) layers that are identical to the layers of a typical Multi-layer Perceptron (MLP) [28]. The configuration of a 1D-CNN is formed by the following hyper-parameters:

1. Number of hidden CNN and dense layers/neurons (in the illustrated sample 1D CNN, there are 3 and 2 hidden CNN and MLP layers, respectively).

2. Filter (kernel) size in each CNN layer.

3. Sub-sampling factor in each CNN layer.

4. The choice of pooling and activation functions.

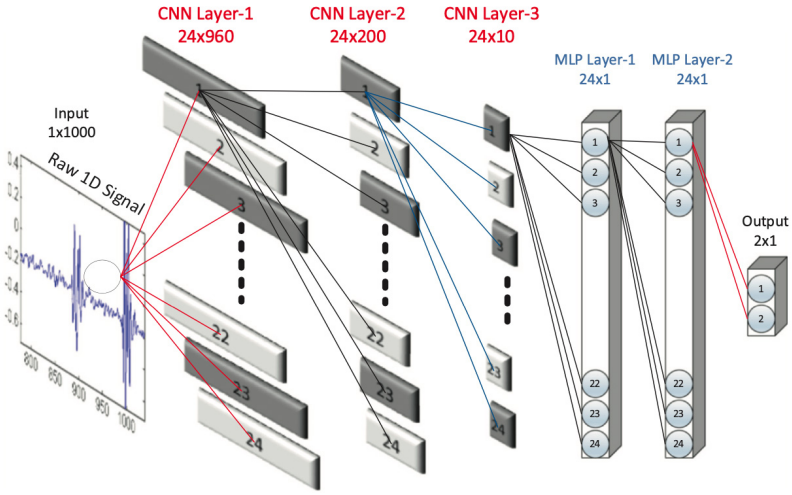


Figure III.2: A sample 1D CNN configuration with 3 CNN and 2 MLP (dense) layers. [28]

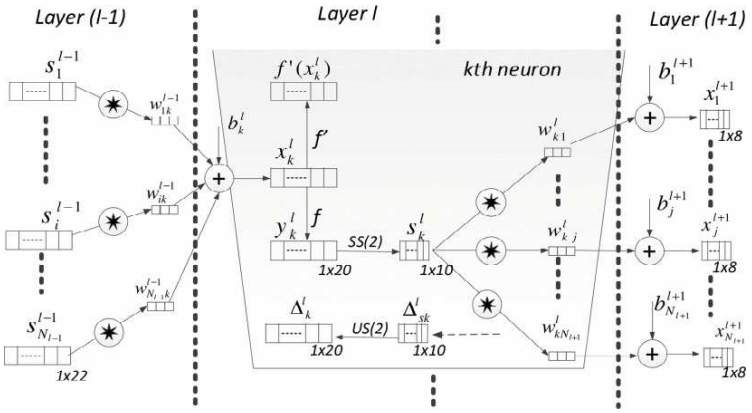


Figure III.3: Three consecutive hidden CNN layers of a 1D CNN. [29]

As in the conventional 2D CNNs, the input layer is a passive layer that receives the raw 1D signal and the output layer is a fully-connected MLP layer with the number of neurons equal to the number of classes. Three consecutive CNN layers of a 1D CNN are presented in Fig. III.3. As shown in this figure, the 1D filter kernels have size 3 and the sub-sampling factor is 2 where the k^{th} neuron in the hidden CNN layer l, first performs a sequence of convolutions, the sum of which is passed through the activation function f, followed by the sub-sampling operation. This is indeed the main difference between 1D and 2D CNNs, where 1D arrays replace 2D matrices for both kernels and feature maps. As a next step, the CNN layers process the raw 1D data and "learn to extract" such features which are used in the

classification/regression task performed by the fully connected layers. As a consequence, both feature extraction and classification/regression operations are fused into one process that can be optimized to maximize the classification performance. This is the major advantage of 1D CNNs which can also result in a low computational complexity since the only operation with a significant cost is a sequence of 1D convolutions which are simply linear weighted sums.

III.2 State-of-the-Art

The demonstration of successful applications of AI in healthcare, finance, business, industries, robotics, autonomous cars and wireless communication including satellites has led it to become a subject of high interest in the research community, industries, and media. Many researchers have discussed AI and its applications to wireless communication in general [30, 31, 32, 33]. Others have focused on the application of AI to one aspect of wireless communication, such as wireless communications in the IoT [27], network management [34], emerging robotics communication [35], antenna design [36] and Unmanned Aerial Vehicle (UAV) networks [36, 37]. Moreover, [38] briefly discussed some promising use cases of AI for satellite communication, whereas [39] discussed the use of AI for space-air-integrated networks.

AI can assist a wide variety of satellite communication aspects including beamhopping, anti-jamming, network traffic forecasting, channel modeling, telemetry mining, ionospheric scintillation detecting, interference managing, remote sensing, behavior modeling, space-air-ground integrating, and energy managing. [40] discusses several challenges facing diverse aspects of satellite communication systems, and their proposed and potential AI-based solutions. Another comprehensive survey of the state-of-the-art in the application of ML techniques to address key problems in IoT wireless communications with an emphasis on its ad hoc networking aspect are given by [28].

The application of ML techniques to make the PRACH receiver more robust to false peaks, which are responsible for performance degradation in the extension of the 4G technique to 5G, are discussed in [41], where authors proposed a hybrid method based on the combination of two ML algorithms, which are k-NN and naïve Bayes. Although even with the hybrid ML method they were not able to satisfy the 3GPP requirements, a considerable improvement is observed. In addition, [42] proposes ML-based solution for enhancing the NB-IoT coverage and to reduce the energy

consumption, where instead of employing a random spectrum access procedure, dynamic spectrum access is used which reduces the number of required repetitions, increases the coverage, and reduces the energy consumption.

Regarding the RA repetition, an ML-based adaptive repetition scheme for a 3GPP NB-IoT system is proposed in [43] to improve overall network transmission efficiency. In [44] a comparison regarding the path loss prediction of two ML methods, namely the Artificial Neural Networks and the Random Forests, was carried out for the two NB-IoT bands at 900 MHz and 1800 MHz. The authors have also discussed how the input data information plays a key role in predicting path loss via ML methods. Also, [45] leverages ML techniques to design a system that is able to estimate the collision multiplicity and thus gather information about how many LTE-M (NB-IoT) devices chose the same preamble, which is later used by the eNB to resolve collisions, increase the supported system load and reduce transmission latency.

Chapter IV

Results and Simulations

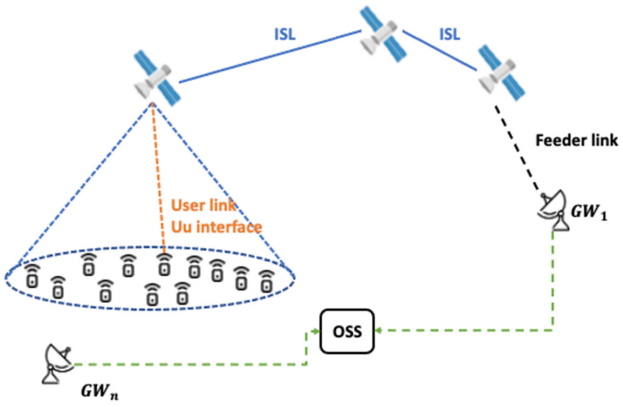


Figure IV.1: High-level NTN architecture

With reference to fig. IV.1, the main elements of the high level system architecture used in this work are: *i)* a plethora of UE, which are fixed position NB-IoT sensors; *ii)* satellites (LEO and vLEO platforms and HAPS) providing connectivity to the UE through the user link (Uu/air interface); *iii)* ground segment, *i.e.*, the GW. The 5G-Advanced Core Network manages the overall system and provides inter-connectivity within the constellation with the Operations Support Systems (OSS) entity. This work assumes the following system configuration: direct access, *i.e.*, the user is directly connected to the flying platform, transparent payload, LEO single-beam platform operating in S-band, with moving beam. In addition, we make the following assumptions: *i)* the feeder link is assumed ideal; *ii)* a standalone NB-IoT deployment is considered; *iii)* the UEs are equipped with a GNSS receiver so as to pre-compensate the common Doppler shift and the propagation delay.

IV.1 Baseband NPRACH Signal

The transmitted SG in the time-domain is expressed in *eq.* (IV.1) [6]:

$$x_{p,n,j}(l) = \sum_{p=0}^{P-1} X_{p,n,j}[N_{sc}(p)] e^{j2\pi \frac{N_{sc}(p)}{L} l} \tag{IV.1}$$

where $x_{p,n,j}(l)$ represents l^{th} sample of the n^{th} symbol in the p^{th} SG of the j^{th} user and $X_{p,n,j}[N_{sc}(p)]$ is the n^{th} unitary symbol on $N_{sc}(p)$ subcarrier where SG p is transmitted; $l = [L_{p,n} - L_{CP}, \ldots, L_{p,n} + L - 1]$, $n = [0, \ldots, N-1]$, and $L_{p,n} = pL_{SG} + nL$, where $L_{SG} = L_{CP} + NL$ is the duration of one SG, L_{CP} is the length of the CP, while one symbol is L samples long.

Considering one user, the received signal after CP removal and DFT, assuming a negligible inter-carrier interference, is expressed as *eq.* (IV.2) [6]

$$Y_{p,n} = h_p e^{j2\pi\eta(L_{p,n} - \tau)} \times e^{-j2\pi n_{sc}^{RA}(p) \frac{\tau}{L}} \left(\frac{1 - e^{j2\pi\eta L}}{1 - e^{j2\pi\eta}} \right) + Z_{p,n} \tag{IV.2}$$

where η is the CFO normalized by the sampling frequency, τ is the RTD normalized by the symbol duration, h_p is the channel coefficient for the p^{th} SG, $n_{sc}^{RA}(p)$ is the subcarrier occupied by the p^{th} SG, and $Z_{p,n}$ is the noise term. Combining the symbols within the same p^{th} SG, we get the so-called SG-sum (SG-S) as follows:

$$Y_p = \sum_{n=0}^{N-1} Y_{p,n} = h_p e^{j2\pi\eta(pL_{SG} - \tau)} \times e^{-j2\pi n_{sc}^{RA}(p) \frac{\tau}{L}} \left(\frac{1 - e^{j2\pi\eta NL}}{1 - e^{j2\pi\eta}} \right) + Z_p \tag{IV.3}$$

The received signal at the BS as a superposition of signals from multiple users can be given as:

$$Y = \sum_{j=0}^{J-1} a_j Y_j + Z \tag{IV.4}$$

where J is the maximum number of concurrent users, and $a_j \in \{0, 1\}$ indicates whether the j^{th} user is active or not.

IV.1.1 Traditional Preamble Detection and ToA Estimation

From the received NPRACH (Msg1) the BS must detect active users and perform synchronization parameter estimation; different algorithms for this process has been proposed in literature.

As a benchmark for the detection of the number of users, an amplitude-based estimator is considered. The mean amplitude of the received signal for different number of colliding users is compared to the amplitude of the received signal. The closest match then yields an estimate of the number of colliding users present in the received signal.

A traditional estimation algorithm is given by [6] and is used in this work for comparison; after the CP removal and applying the DFT, a differential processing is performed by the p^{th} SG-S with the complex conjugated $(p+1)^{th}$ SG-S of the same repetition. Defining $\Delta(p) = n_{sc}^{RA}(p+1) - n_{sc}^{RA}(m)$ as the hopping step between the p^{th} and $(p+1)^{th}$ SGs, resulting in:

$$Z_{p,1} = Y_p Y_{p+1}^* \propto e^{-j2\pi\eta L_{SG}} e^{j2\pi\Delta(m)\frac{\tau}{L}} \tag{IV.5}$$

Then, the symbols in the same SG are summed to obtain the SG-S, This differential processing is performed for all the SGs and their outputs, *i.e.*, the symbols $Z_{p,1}$, are collected in an array ν_u . Within the vector ν_u, the symbols are ordered based on their hopping difference, *i.e.*, $\Delta(p) = N_{sc}(p+1) - N_{sc}(p)$. Then, the Rife & Boorstyn (R&B) algorithm is applied to each array element considering 512 points N_{DFT}:

$$V_u[k] = \sum_{n=0}^{N_{DFT}-1} \nu_u[n] e^{-j2\pi k \frac{n}{N_{DFT}}} \tag{IV.6}$$

where k refers to the subcarrier index. Since the number of arrays ν_u is equal to N_{rep}, $V_u[k]$ is non-coherently combined over N_{rep}:

$$X[k] = \sum_{u=0}^{N_{rep}-1} |V_u[k]|^2. \tag{IV.7}$$

With $X_{max} = X[k_{max}]$ compared to a threshold, presence of a preamble is declared, where $k_{max} = \arg\max_k X[k]$. And finally, the ToA is computed as:

$$ToA = \frac{k_{max}}{N_{DFT} \cdot \Delta f} \tag{IV.8}$$

One interesting property of the above method is its insensitivity to CFO when estimating the ToA. The term $e^{-j2\pi\eta L_{SG}}$ is common for all $Z_{p,1}$ symbols (*i.e.*, common for all $\nu[n]$ and affects only the phase of $\nu[n]$, not its magnitude. Similarly, since the DFT operation is linear, this factor affects only the phase of $V[k]$, not

its magnitude, such that by taking the absolute of $V[k]$, the impact of CFO is completely eliminated [6].

All in all, it should be noted that with traditional methods it is possible to estimate only a single value of ToA, regardless of the number of interferes.

IV.1.2 Channel Characteristics

In the considered coverage area, we assume U users uniformly distributed. Within the beam, each user is described by a complex channel coefficient representing the channel between the j^{th} user and the satellite, given by:

$$h_j = \frac{\sqrt{G_{Tx} G_{Rx} \Omega(\theta_j^s)}}{4\pi \frac{d_j}{\lambda} \sqrt{A_{loss,j} P_z}} e^{-j\frac{2\pi d_j}{\lambda}} \tag{IV.9}$$

where *i)* G_{Tx} represents the transmission gain of the device antenna, which is equivalent for all users. For the purpose of the work it is assumed that the NB-IoT terminals are quipped with an omnidirectional antenna; *ii)* G_{Rx} is the maximum receiving gain of the satellite; *iii)* d_j is the slant range between the j^{th} user in the beam and the satellite; *iv)* λ is the carrier wavelength; *v)* P_z is the noise power, which depends on the satellite antenna equivalent noise temperature, T, and the user bandwidth, B. *vi)* A_{loss} represents the additional losses. The latter are computed as $A_{loss,j} = A_{shadowfading} + A_{atmospheric} + A_{scintillation} + A_{clutter}$; these terms vary depending on the user position and are computed as per 3GPP [TR 38.821] [5]. Finally, the term $\Omega(\theta_j^s)$ defines the satellite radiation pattern, which is a function of the angle between the antenna pointing of the j^{th} user and the s^{th} satellite, *i.e.*, θ_j^s, which does not depend on the satellite antenna index due to antenna co-location. In this study, we consider a Bessel radiation pattern [46]. Based on the above assumptions and definitions, the signal received at the satellite can be computed as follows:

$$Y_p = \sum_{j=1}^{J} \sqrt{P_{tx}} h_j s_{p,j} + z_p \tag{IV.10}$$

where J is the number of concurrent users, *i.e.*, the users transmitting the same preamble identified by the sub-carrier p, P_{tx} is the transmitting power, which is assumed to be the same for all users, and z_p is a complex circularly-symmetric Gaussian random variable with zero-mean and unit variance. The latter is a licit

assumption, since the noise power is included in the term P_z in *eq.* (IV.9). Finally, $s_{p,j}$ denotes the signal from the j^{th} user, given by:

$$s_{p,j} = \sum_{p=0}^{P-1} X_{p,n,j}[N_{sc}(p)] e^{j2\pi\eta \frac{l-\tau}{L} N_{sc}(p)} \tag{IV.11}$$

where τ is the ToA due to the RTD and η is the CFO. At the receiver side, the DFT is applied to each symbol of each SG, included the CP.

IV.2 NN Approach to Preamble Detection

In the proposed scenario, two Convolutional Neural Networks are designed to extract the non-linear relationship between the DFT coefficients of each symbol of each SG and their targets; the two CNNs are used in cascade: the first one is dedicated to the collision classification, *i.e.*, it classifies the number of users transmitting the same preamble, and its output is used as input for the ToA estimation network.

IV.2.1 Neural Network Implementation

The CNN architectures for the collision classification and the ToA estimation are depicted in fig. IV.2 and fig. IV.3, respectively. Both the schemes follow the same structure up to the last layer, where they differ to perform the two different tasks, *i.e.*, the classification of the number of users and the regression; the latter aims at yielding ToA values close to the original ones in the Mean Square Error (MSE) sense. With respect to the networks input, both the CNNs take the DFT coefficients of the received symbols and the second network, in charge of the ToA estimation, leverages also the number of colliding users identified by the collision classification network. The first two layers are implemented with convolutional layers followed by max-pooling operations. Thus, the input is processed so as to extract spatially correlated features that are exploited for the task learning. The 1D convolution layer is followed by batch normalization, ReLu activation function, and max-pooling. The latter aims at reducing the output dimension of the CNN and making the network invariant to the small changes in the input and, therefore, to the noise in the training samples. It is worth noting that the ToA and the CFO are assumed to be constant throughout a transmission, therefore the 1D convolution layer is capable of extracting invariant features of the frequency-domain signal. Subsequently, these features are shaped into a single vector that feeds three fully connected layers. The

first two are followed by the ReLu activation (large corrections), while the last one is softmax activation (refining corrections) for the classification network and a linear layer for the regression.

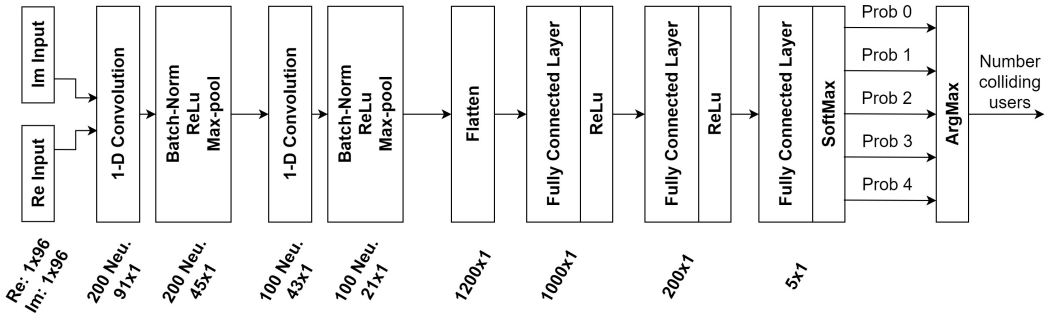


Figure IV.2: Structure of the 1D CNN for collision classification.

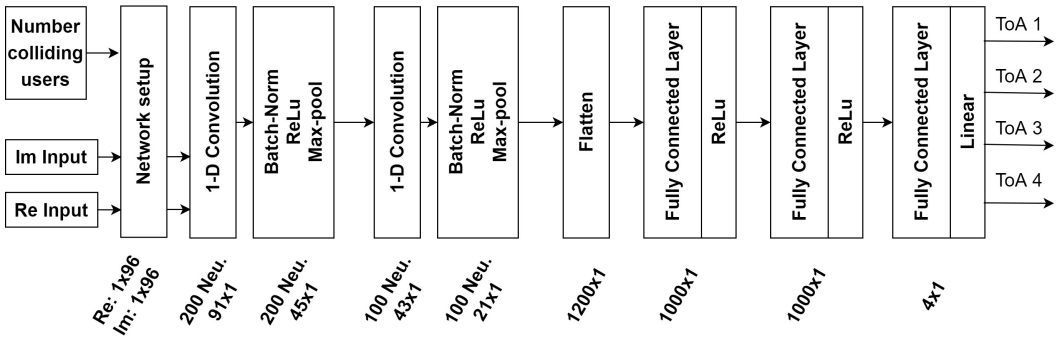


Figure IV.3: Structure of the 1D CNN for ToA estimation.

IV.2.2 Collision Classification

The collision classification is commonly applied to all the available preambles and separates them into several classes. This network accepts $(\mathfrak{Re}(y), \mathfrak{Im}(y))$ of the DFT coefficients of each symbol as an input and it is trained to predict the number of devices transmitting the same preamble. To this aim, the following $U+1$ different classes are considered: *i)* **CLASS 0**: the preamble is unused, meaning there is only noise. *ii)* **CLASS 1**: the preamble is selected by a single UE, *i.e.*, collision-free. *iii)* **CLASS u**: the preamble is chosen by u out of U terminals for $u \in \{2, \ldots, U-1\}$, therefore, there are u collisions. *iv)* **CLASS U**: the same preamble has been selected by U terminals. The label of each $U+1$ class consists of a one-hot vector $t = OneHot(U)$, where the index u in the truth label $t = [t_0, t_1, \ldots, t_U]$ is one

and zero everywhere. Therefore, the number of collided users per preamble is given by the $L0$ norm of the sparse vector t. This network is trained so as to minimize the softmax cross entropy loss function (or negative log-likelihood), given by:

$$L_{cc}(t,p) = -\sum_{u=0}^{U} t_u \log(p_u) \tag{IV.12}$$

where p_u is the output of the softmax function, which computes the probability associated to each category by taking the scores of the Fully Connected layer, as follow:

$$p_u = \frac{e^{a_u}}{\sum_{j=0}^{U} e_{a_u}} \tag{IV.13}$$

where $[a_0, \ldots, a_U]$ are the outputs of the Fully Connected layer. Therefore, the index (u_{max}) associated to the highest probability value, *i.e.*, $p_{max} = \arg\max p$, within the vector $p = [p_0, \ldots, p_U]$ is selected to obtain the number of colliding users.

IV.2.3 Delay Estimator

Following the preamble collision classification, the ToA estimation is performed by the second CNN. For each preamble, the neural network takes as input the $(\mathfrak{Re}(y), \mathfrak{Im}(y))$ of the DFT coefficients of each symbol together with the class value that identifies the number of non-orthogonal users, *i.e.*, u_{max}, and it is trained to estimate the ToA of the colliding terminals in each signal. To this aim, the ToA values are normalized on the highest value, thus enabling the training convergence, and they are collected in the vector $\mathbf{Y} = [\tau_0, \tau_1, \ldots, \tau_{U-1}]$. Therefore, the network finds an estimate $\hat{\mathbf{Y}}$ of $\mathbf{Y}$ so as to minimize the MSE loss function *i.e.*, $\mathbf{E} = ||\mathbf{Y} - \hat{\mathbf{Y}}||^{\mathbf{2}}$.

IV.3 Numerical Results

In this section, we evaluate the performance of the proposed CNN-based Random Access. We first examine the accuracy of the number of colliding users classification and then the ToA estimation performance. For the first network, the learning rate is fixed to 5×10^{-4}, while for the later it is 1×10^{-4}, which guarantee the convergence of both the CNNs. The dataset consists of 42×10^6 samples generated in Matlab following a Monte Carlo simulation, and organized in 4200 batches with 10000 examples each, where every input is the couple of the real and imaginary part of 96 DFT symbols, trained over 20 epochs in TensorFlow Keras, Python. Each preamble

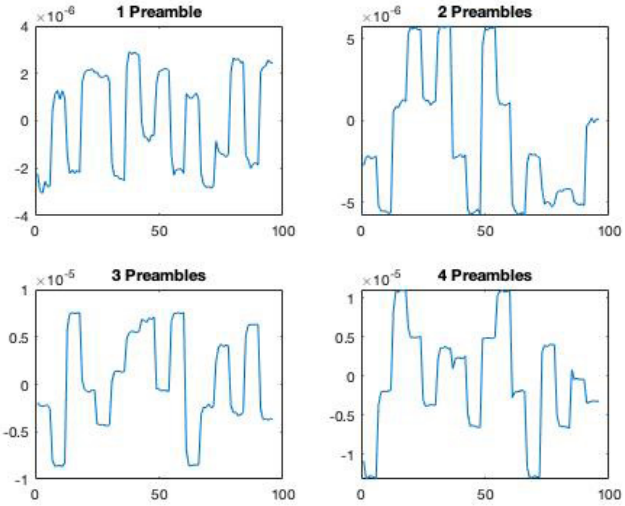


Figure IV.4: Real part time-domain representation of the interfered preambles

consists of four SGs, each one with five symbols plus the CP, whose length over time is the same as a symbol, *i.e.*, 266.7μs. Therefore, considering four repetitions, the total amount of symbols sent is given by $\{6 \times 4 \times 4 = 96\}$. For both the collision classification and the ToA estimation, we generated a dataset with DFT symbols containing from 0 to 4 colliding users per preamble; Real part of the time-domain received preamble signal of four randomly selected examples that are fed to the Neural Networks are given in fig. IV.4; each subplot corresponds to a different number of interfered terminals' signals that have been collided (CLASS 0,1,2,3). 80% of the samples are used for training with the stochastic optimization method based on ADAM, and 20% for testing. The estimation performance of the ToA is evaluated in terms of the Root Mean Square Error (RMSE) (Table IV.3). For each user transmitting the same preamble, the network estimates the ToA and the estimation accuracy is obtained by averaging all the RMSEs up to the value u_{max}.

$$RMSE = \sqrt{\frac{1}{u_{max}} \sum_{u=1}^{u_{max}} \left\| \mathbf{Y_u} - \hat{\mathbf{Y}}_\mathbf{u} \right\|^2} \tag{IV.14}$$

Both the CFO and ToA are randomly selected in the range $[-40, 40]Hz$ and $[-50, 50]\mu s$, respectively [47]. The proposed method is evaluated considering the frequency and the timing residual error after the pre-compensation done with the GNSS. Moreover, all datasets are generated considering the following scenarios: *i)* one coverage enhancement zone, *i.e.*, all the 48 sub-carriers are available for the

preamble transmission; *ii)* NB-IoT transmitting power equal to $P_{tx} = -7$ dB; *iii)* receiver noise power equal to $P_z = -163.8$ dB; *iv)* two different satellite configurations proposed in [48], *i.e.*, set 3 and set 4, whose parameters are shown in Table IV.1; *v)* one propagation environment, *i.e.*, sub-urban, in both Line of Sight (LoS) and Non Line of Sight (NLoS) conditions, whose parameters are available in [49]; *vi)* UEs uniformly deployed in the beam coverage.

Table IV.1: Satellite parameters [48]

Satellite orbit	Set 3 LEO 600 km	Set 4 LEO 600 km
Equivalent satellite antenna aperture	0.4 m	0.097 m
Sat Tx/Rx max Gain	16.2 dBi	11 dBi
3dB beamwidth	22.1 degree	104.7 degree
Sat beam diameter	234 km	1700 km
G/T	-12.8 dB K^{-1}	-18.6 dBK^{-1}

It shall be noticed that, in UL, the interference towards the satellite is clearly depending on the specific interfering UE location. Therefore, both the shadow fading and the interference source location vary through each example, in addition to the subcarrier index of the preamble. The shadow fading is modeled as a log-normal random variable $A_{sh} \sim (0, \sigma^2)$ with the values of σ^2 depending on the UE's elevation angle [46]. In addition, we consider the Carrier to Noise Ratio (CNR) ranges reported in Table IV.2. For each of the proposed scenarios, we trained the network with the minimum achievable CNR and we tested it by randomly choosing users within the beam coverage. As recorded, the CNR variance in NLoS condition is higher than the one in the LoS, due to presence of the clutter loss (CL). Notably, the larger the beam size, the greater the variance of σ^2.

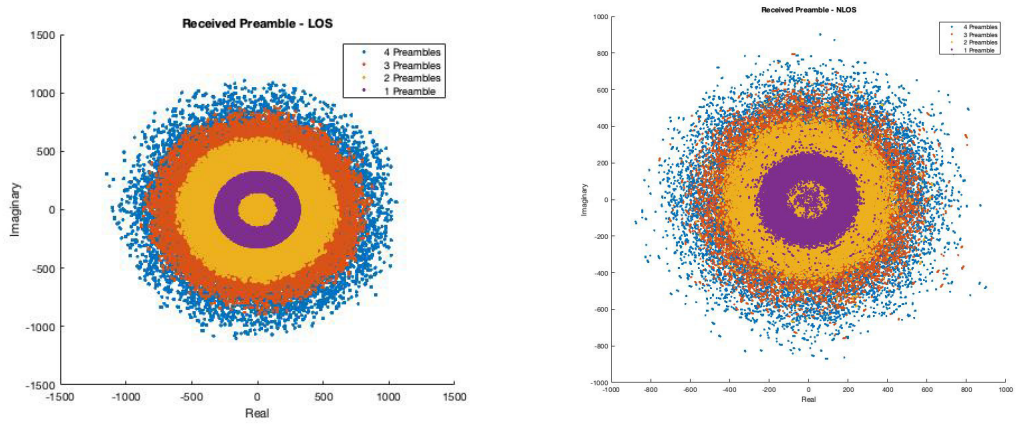
IV.3.1 Collision Classification Performance

Fig. IV.5a and IV.5b depict the inputs of both the CNNs, *i.e.*, the $\{\mathfrak{Re}, \mathfrak{Im}\}$ parts of the DFT symbols in LoS and NLoS conditions varying the number of colliding users per preamble. The datasets are equally distributed among the five classes. It is worth emphasizing that the annulus in the figures, which correspond to the outer edge of an interfered preamble, gains a different dimension based on the number of colliding terminals (shown in different colors).

Indeed, the three outer colored area host symbols that have been constructively interfered by 1,2, or 3 more users in a received signal and the values of the $\{\mathfrak{Re}, \mathfrak{Im}\}$

Table IV.2: Mean and variance of CNR

Satellite Set	Propagation condition	CNR dB
Set 3	LoS	$\mu = 9.3$
		$\sigma = 1.42$
	NLoS	$\mu = -6.9$
		$\sigma = 7.03$
Set 4	LoS	$\mu = 0.1$
		$\sigma = 2.14$
	NLoS	$\mu = -15.05$
		$\sigma = 10.04$



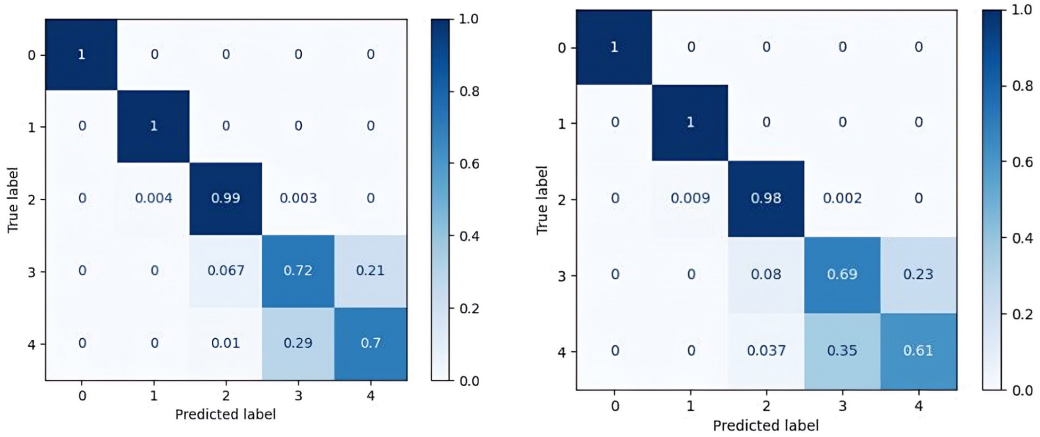
(a) LoS condition

(b) NLoS condition

Figure IV.5: Real and Imaginary part of the DFT symbols. Set 3

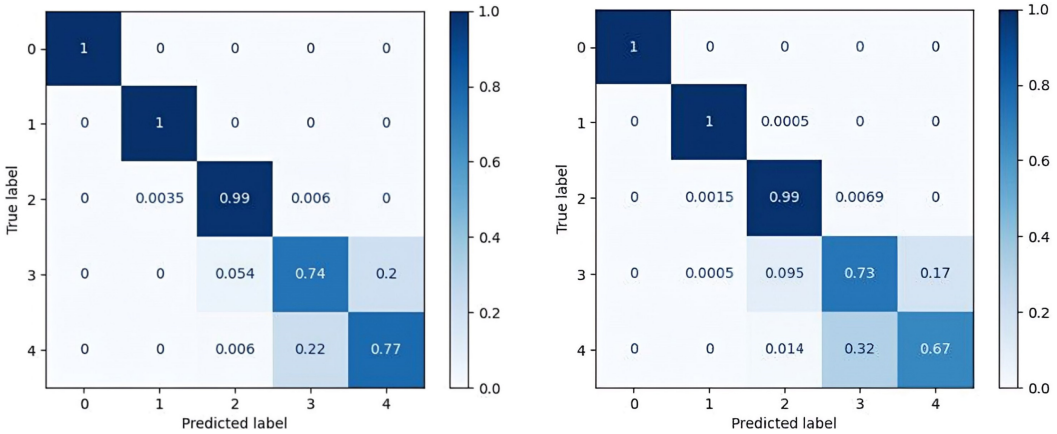
parts are key in the learning process. Given that the symbols of an interfered signal set belong to a circle (radius extending from the center to the outlined edges) and not an annulus, the larger the number of collisions per signal the less symbols will be accommodated in the **Learning Range (LR)** of the given group, *i.e.*, the amplitude of the signal with a constructive interference. This means that destructively interfered symbols (sitting more towards the center of the circle) might cause a misclassification if there are no other symbols in the preamble that have acquired sufficient amplitude to fit within the *LR* annulus of that class.

Thus, with the same number of examples per class, the CNN is able to correctly classify 0, 1, and 2 interferers per preamble, as reported in Figs. IV.6a - IV.7b



(a) LoS condition (b) NLoS condition

Figure IV.6: Outcome of the classification: Set 3



(a) LoS condition (b) NLoS condition

Figure IV.7: Outcome of the classification: Set 4

(the confusion matrices). Indeed, with a maximum of two colliding terminals, the network is able to correctly classify the number of colliding users with a probability of about 0.99. On the other hand, classes 4 and 5 are the most difficult to correctly classify. The classification accuracy decreases due to the fact that the network has fewer examples in the corresponding *LR*.

In NLoS condition, edges of the *LR* have less outlined boundaries compared to the LoS condition, because symbols of a collision class are placed outside the *LR* of the corresponding group. This leads to a degradation of the classification performance. Due to the large beam sizes and, consequently, the low elevation angles, resulting in

large values of CL, the classification of users within Set 4 in NLoS condition shows the worst performance.

IV.3.2 Delay Estimator Performance

Referring to Table IV.3, it is observed that, despite low CNR values and uncompensated CFO, the RMSE of the ToA is around 5 μs in absence of collision, while the performance decreases by increasing the number of superimposed symbols, as expected. It should be noted that, fixed the number of interferers, the estimation accuracy remains constant across the two different satellite sets in both the LoS and NLoS conditions.

Subsequently, to understand the goodness of the CNN based ToA estimation, we compare it with the technique applied in [6] which was described in subsection IV.1.1. The accuracy of the R&B estimation algorithm is measured as follow: $\mathbf{E_{R\&B}} = \mathbf{min}\|\mathbf{Y} - \mathbf{Y}_{\hat{\mathbf{R\&B}}}\|$. Considering only one UE, the proposed CNN approach provides higher accuracy under the NLoS condition with respect to the one achievable with the R&B method, as shown in Table IV.3. Clearly, the CNN based ToA estimation provides advantages in case of collisions, as it is able to estimate the ToA for each colliding users also in presence of CFO as well as with low CNR values.

The proposed method is designed to work with the GNSS but it could also be adopted in case where the UEs do not have GNSS capabilities. On the other hand, to provide more accurate estimation, the network needs as input a large number of examples with different types of channel coefficients. In fact, the ToA estimation in set 4 is slightly better, since the Neural Networks can learn from different types of coefficients due to the wider beam dimension.

Table IV.3: RMSE of ToA [μs]

Satellite Set	Prop. condition	N.users	CNN Output	R&B Output
Set 3	LoS	1	5.3 μs	0.27 μs
		2	16.4 μs	26.7 μs
		3	19.8 μs	30.3 μs
		4	22.5 μs	32.7 μs
	NLoS	1	5.7 μs	5.8 μs
		2	17.0 μs	74 μs
		3	20.5 μs	142.2 μs
		4	23.1 μs	165.1 μs
Set 4	LoS	1	4.8 μs	1.4μs
		2	15.3 μs	27.2 μs
		3	18.1 μs	57.3 μs
		4	21.7 μs	112.6 μs
	NLoS	1	4.2 μs	4.89 μs
		2	14.7 μs	27.8 μs
		3	17.3 μs	87.8 μs
		4	20.9 μs	120 μs

Chapter V

Conclusions

In this Thesis we proposed an innovative detection and estimation method in support of NOMA approaches for mMTC. In particular, the collision classification and the ToA estimation are performed by two different CNNs. The latter are trained and tested in real propagation environment, under the NLoS and LoS conditions, considering two different satellite configurations. The extensive simulations show that Deep Learning is a promising tool for these two learning tasks. Thus, future works foresee the exploitation of the information related to number of colliding devices and their ToA to enhance the capacity of the system by designing a Non Orthogonal Random Access scheme, where it is possible to schedule the Msg2 based on the number of colliding devices and their ToA and, then, apply NOMA techniques on Msg3 and, subsequently, on the data. The greater the accuracy of the estimation of these parameters, the greater the network performance in terms of network access time, access probability and, thus, throughput [50].

The study on adaptation of 2-step RACH to NTN(-IoT) has shown great benefits in terms of reducing the whole RA procedure from half the average-access-time to maximum the same time as conventional 4-step guaranteeing latency performance no worse than before, specially when solutions like the one proposed in this work is implemented to also serve the colliding terminals. In particular, in an NB-IoT scenario in which the UE may also transmit its packet of data along with MsgA in an assigned NPUSCH and conclude not only its access but also its mission by means of a protocol called small data transmission (SDT). Future work will focus on adaptation of NR 2-step RACH to NTN with minimal to no modification on current specifications for a **Unified 6G**.

Bibliography

[1] Fortune Business Insights, "Market research report, 2022-2029," March 2022, https://www.fortunebusinessinsights.com/industry-reports/internet-of-things-iot-market-100307.

[2] Ala Al-Fuqaha et al., "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE Communications Surveys Tutorials*, 2015.

[3] 3GPP, "TR36.763, Technical Specification Group Radio Access Network; Study on Narrow-Band Internet of Things (NB-IoT) / enhanced Machine Type Communication (eMTC) support for Non-Terrestrial Networks (NTN); (Release 17), V17.0.0, (2021-06)," .

[4] 3GPP, "TS36.211, Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical channels and modulation; (Release 17), V17.0.0, (2022-03)," .

[5] 3GPP, "TR38.821, Technical Specification Group Radio Access Network; Solutions for NR to support non-terrestrial networks (NTN); (Release 16), V16.1.0, (2021-05)," .

[6] Houcine Chougrani et al., "NB-IoT Random Access for Nonterrestrial Networks: Preamble Detection and Uplink Synchronization," *IEEE Internet of Things Journal*, 2022.

[7] 3GPP, "TS36.213, Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA); (Release 16), V16.4.0, (2020-12)," .

[8] Hwang et al., "Efficient Detection and Synchronization of Superimposed NB-IoT NPRACH Preambles," *IEEE Internet of Things Journal*, 2019.

[9] X. Lin et al., "Random Access Preamble Design and Detection for 3GPP Narrowband IoT Systems," .

[10] 3GPP, "TS38.211, Technical Specification Group Radio Access Network; NR; (Release 17), V17.1.0, (2022-03)," .

[11] "European Vision for the 6G Network Ecosystem," July 2021.

[12] Alessandro Vanelli-Coralli et al., "5G and Beyond 5G Non-Terrestrial Networks: trends and research challenges," in *2020 IEEE 3rd 5G World Forum (5GWF)*, 2020.

[13] Guidotti et al., "Architectures and Key Technical Challenges for 5G Systems Incorporating Satellites," *IEEE Transactions on Vehicular Technology*, 2019.

[14] Oltjon Kodheli et al., "An Uplink UE Group-Based Scheduling Technique for 5G mMTC Systems Over LEO Satellite," *IEEE Access*, 2019.

[15] Conti et al., "Doppler Impact Analysis for NB-IoT and Satellite Systems Integration," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020.

[16] Conti and othres, "NB-IoT over Non-Terrestrial Networks: Link Budget Analysis," in *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, 2020.

[17] Alessandro Vanelli-Coralli, Giovanni E. Corazza, et al., "The ISICOM Architecture," in *2009 International Workshop on Satellite and Space Communications*, 2009.

[18] Kodheli et al., "Integration of Satellites in 5G through LEO Constellations," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, 2017.

[19] Cluzel et al., "3GPP NB-IOT Coverage Extension Using LEO Satellites," in *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*, 2018.

[20] Amatetti et al., "Preamble detection in NB-IoT via Satellite: a Wavelet based approach," in *2021 IEEE Global Communications Conference (GLOBECOM)*, 2021.

[21] Kodheli et al., "Random Access Procedure Over Non-Terrestrial Networks: From Theory to Practice," *IEEE Access*, 2021.

[22] Olof Liberg et al., "Narrowband Internet of Things for Non-Terrestrial Networks," *IEEE Communications Standards Magazine*, 2020.

[23] Petrosino et al., "WIP: An Open-Source Tool for Evaluating System-Level Performance of NB-IoT Non-Terrestrial Networks," in *2021 IEEE 22nd International Symposium on WoWMoM*, 2021.

[24] 3GPP, ," R1-211658, RAN vice-chair, Moderators summary for discussion [RAN93e-R18Prep-08] NTN evolution, RP-93-e meeting.

[25] Piao et al., "Random Power Back-Off for Random Access in 5G Networks," *IEEE Access*, 2021.

[26] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning*, MIT Press, 2016.

[27] J. Jagannath et al., ," Machine learning for wireless communications in the Internet of Things: A comprehensive survey, Ad Hoc Networks, Volume 93, Jan. 2019.

[28] "1D convolutional neural networks and applications: A survey," *Mechanical Systems and Signal Processing*, vol. 151, pp. 107398, 2021.

[29] Serkan Kiranyaz et al., "Real-Time Fault Detection and Identification for MMC Using 1D Convolutional Neural Networks," *IEEE Transactions on Industrial Electronics*, 2019.

[30] O. Simeone, ," A Very Brief Introduction to Machine Learning With Applications to Communication Systems", IEEE Transactions on Cognitive Communications and Networking.

[31] and others M.Chen, ," Artificial Neural Networks-Based Machine Learning for Wireless Networks: A Tutorial, IEEE Communications Surveys Tutorials .

[32] and others Y. Qian, ," Survey on Reinforcement Learning Applications in Communication Networks, Journal of Communications and Information Networks, June 2020.

[33] EC. Strinati et al., ," 6G: The Next Frontier: From Holographic Messaging to Artificial Intelligence Using Subterahertz and Visible Light Communication, IEEE Vehicular Technology Magazine.

[34] G. P. Kumar et al., ," "Artificial intelligence approaches to network management: recent advances and a survey", Computer Communications, Volume 20, Issue 15, Dec. 1997.

[35] S. H. Alsamhi et al., ," "Survey on artificial intelligence based techniques for emerging robotic communication", Telecommunication Systems: Modelling, Analysis, Design and Management,.

[36] H. Misilmani and T. Naous, ," "Machine Learning in Antenna Design: An Overview on Machine Learning Concept and Algorithms", 2019 International Conference on High Performance Computing Simulation (HPCS), Dublin, Ireland, 2019.

[37] P.Bithas et al., ," A survey on machine-learning techniques for UAV-based communications, Sensors 2019, Nov. 2019.

[38] M.A.Va Ìzque et al., ," "On the Use of AI for Satellite Communications", arXiv preprint, 2020.

[39] N. Kato et al., ," "Optimizing Space-Air-Ground Integrated Networks by Artificial Intelligence," IEEE Wireless Communications, August 2019.

[40] Fares Fourati et al., "Artificial intelligence for satellite communication: A review," *Intelligent and Converged Networks*, 2021.

[41] Modina et al., "A machine learning-based design of PRACH receiver in 5G," *Procedia Computer Science*, 2019.

[42] Chafii et al., "Enhancing coverage in narrow band-IoT using machine learning," in *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, 2018.

[43] Chen et al., "Adaptive Repetition Scheme with Machine Learning for 3GPP NB-IoT," in *2018 IEEE 23rd Pacific Rim International Symposium on Dependable Computing (PRDC).*

[44] P. Sotiroudis et al., "Neural networks and random forests: A comparison regarding prediction of propagation path loss for nb-iot networks," in *2019 8th International Conference on Modern Circuits and Systems Technologies (MOCAST)*, 2019.

[45] Magrin et al., "Enabling LTE RACH Collision Multiplicity Detection via Machine Learning," in *2019 International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOPT)*, 2019.

[46] 3GPP, "TR38.811, Technical Specification Group Radio Access Network; Study on New Radio (NR) to support non-terrestrial networks (Release 15), V15.3.0," .

[47] 3GPP, "R1-2009095, Discussion on RAN1 aspects of IoT NTN, RAN1 meeting 103e," .

[48] 3GPP, "TR36.763 Study on Narrow-Band Internet of Things (NB-IoT) / enhanced Machine Type Communication (eMTC) support for Non-Terrestrial Networks (NTN)," June 2021.

[49] 3GPP, "TR38.821, Technical Specification Group Radio Access Network; Solutions for NR to support non-terrestrial networks (NTN) (Release 16), V16.0.0," .

[50] Ali Georganaki Carla Amatetti, Riccardo Campana and Alessandro Vanelli-Coralli, "Neural Network based Non Orthogonal Random Access for 6G NTN-IoT," *IEEE Globecom22*, Dec 2022.

Acknowledgements

I am grateful to the University of Bologna; largely for the top quality education system they have developed for seekers of science, and the ace professors of DEI who philanthropically school engineers, and evaluate their students not on a traditional sit-and-go basis but in a vis-à-vis manner, patiently listening to the way students return the taught elements and decide whether they are ready to take a step forward. My special thanks to the Digicomm research group for providing the necessary tools of research, such as the server used for training the model proposed in this work, and the Marconi Institute for Creativity where ideas are moved.

I would like to extend my sincere gratitude to Professor Alessandro Venelli, the Guru who ignited my passion in SatCom and has curved my career path toward a better good with his excellent mentorship and supervision. Alessandro has kept me sane by overclocking my processing capabilities whenever the complexity struck; with his take-it-easy smile over a cup of coffee in his "office". He has introduced me to two exceptional scientists, my betters, Carla and Riccardo, whom I would like to highlight their intelligence and their propagating efficiency in simultaneous research activities. Dr. Amatetti, Dr. Campana you are a bliss to collaborate with, I wish to be shoulder to shoulder with you when you move borders of science in the upcoming decades.

I am also indebted to Professor Giovanni Corazza for telling me the secrets of climbing the mountain where a virgin land on the top awaits a creative explorer to penetrate its unknown. And my most truthful appreciations to Professor Massimo Ferri and his insights on teaching engineers how to make good use of mathematics in their fields; I have graciously accepted a task given by him, that is to bring the charm of topology into engineering, having in mind a quote that I also learned from him: "What is the only maths you can't ever apply? The one you don't know!"