Wireless Interference Identification with Convolutional Neural Networks based on the FPGA implementation of the LTE Cell Specific Reference Signal (CRS)

Gianmarco Baldini, European Commission, Joint Research Centre
Ispra, 21027, Italy
Email: gianmarco.baldini@ec.europa.eu
Fausto Bonavitacola, Piksel S.p.A.,
Milan, Italy
Jean-Marc Chareau European Commission, Joint Research Centre
Ispra, 21027, Italy

***Abstract*—Wireless Interference Identification (WII) is an important function in the context of non-cooperative spectrum coexistence management problems because interference may cause loss or degradation of service. Deep Learning (DL) has been recently introduced in spectrum coexistence problems and more specifically in WII where it has demonstrated a superior performance to shallow machine learning algorithms. This paper proposes an advancement in literature by exploiting the existing function in modern cellular networks systems for channel estimation to implement WII. In particular, a channel estimator function based on the LTE Cell Specific Reference Signal (CRS) was implemented in Field Programmable Gate Array (FPGA) by the authors and it was used to generate channel estimates, which are given as an input to a DL algorithm. This study applies Convolutional Neural Networks (CNN) on three different data sets for WII, where the victim is a LTE-plus communication system with 40 MHz bandwidth and the interferences are 1) LTE with 20 MHz bandwidth and FDD modulation, 2) LTE with 20 MHz bandwidth and TDD modulation and 3) WiFi (802.11g). This paper describes the FPGA channel estimator implementation and it performs an extensive analysis of the impact of the parameters of the proposed approach and the CNN architecture. The results show that the proposed approach outperforms other approaches based on DL and constellation diagrams or shallow machine learning algorithms.**

***Index Terms*—Deep Learning, wireless interference, channel estimates, cellular networks.**

I. Introduction

The acquisition of the radio frequency channel state information is a fundamental task that a wireless communication receiver has to perform prior to information symbols extraction from the received radio frequency signal. Channel identification can facilitate channel equalization as well as maximum likelihood sequence detection. In the context of cognitive radio, the knowledge of the channel state information contributes to the radio frequency spectrum awareness, which is an important function for cognitive radios [1]. In this paper, the channel state information is used to detect and identify the presence of wireless interferences. The detection of wireless interferences in the radio frequency spectrum is another essential function needed in cognitive radio and dynamic spectrum management to support the coexistence of different wireless services. Historically, wireless interference detection and mitigation has been implemented using interference models, but an alternative approach is to evaluate the impact of the interference as described in [2], where interference detection can be classified into two large groups: (1) models that describe the characteristics of the interference signal itself, and (2) models that describe the effects of interference. This paper proposes an approach, which belongs to the second category. In recent times, authors have proposed the application of Machine Learning (ML) techniques to the problem of detection of wireless interference. This is part of a more general trend on the application of ML to wireless communication for a variety of functions. In particular DL, based on neural networks, has been applied with significant success to many wireless communication problems as described in the recent survey [3]. More specifically on the problem of wireless interference, DL and CNN have been applied to the detection of WiFi interference for the first time in [4], where it has demonstrated to attain a higher identification performance in comparison to 'shallow machine learning' algorithms (e.g., Support Vector Machine). Additional details on the recent application of ML and DL to the problem of wireless interference detection and mitigation are provided in Section II.

Our contribution: This paper provides an approach based on DL and more specifically on CNN to the problem of adjacent-channel wireless interference identification where the LTE wireless communication service is the victim and signals based on WiFi 802.11g, LTE TDD and LTE FDD standards and modulations are the interferers. The approach is based on the novel exploitation (for this context) of the channel estimates from the LTE Cell Specific Reference Signal (CRS) channel estimator (which is generally available in LTE equipment), which is fed in input to the CNN. For this study, a custom LTE-plus channel estimator was designed and implemented in a Field-Programmable Gate Array (FPGA) where the LTE-plus refers to the FPGA implementation of a 40 MHz bandwidth communication transmission channel and related front-end components (e.g., modulators and demodulators) in addition to the channel estimator itself. The approach was evaluated in a radio frequency test laboratory where signals were generated for three different types of interferer at different values of power and spectral distance from the LTE-plus victim service. The proposed approach based on CNN is compared to approaches proposed in literature based on the application of ML algorithms and False Negative (IQ) constellation diagrams.

From a potential practical implementation of the proposed approach, the idea is to exploit the channel estimators to mitigate the risk of wireless interference (especially adjacent wireless interference) in radio frequency spectrum management scenarios for cellular networks. The need for fair coexistence among LTE and WiFi systems is an important area both from the research and deployment points of view [5], [6]. The detection and more importantly the classification of the interference as proposed in this paper can improve the spectrum awareness of wireless communication systems and improve the overall quality of service. From the practical point of view, the advantage of this approach is that it uses already existing functions in wireless receivers even if they are designed for a different purpose (i.e., channel estimation).

Note: To support a comprehensive analysis of three potential interference services (i.e., based on WiFi 802.11g, LTE TDD and LTE FDD standards) to the LTE-plus victim service for adjacent band interference, the study incorporates a deviation from the current RF spectrum regulation (in particular for the spectrum allocation of the systems based on 802.11g standard). This study was done in the context of pre-normative research to address future RF spectrum allocations where this risk of adjacent band interference may materialize. Additional details on this aspect are provided in Section III. We would also like to highlight that the approach for interference detection is implemented in baseband rather than in the operating carrier frequency of the wireless services and thus the choice of the operating frequency has a limited or negligible impact.

The structure of this paper is following. Section II provides a literature review on WII and on the recent application of DL to the wireless communication domain with a focus on spectrum coexistence. Section III provides an extensive description of the data sets generated by the authors to evaluate the approach and the test bed used to generate the data. It also includes the description of the FPGA implementation of the LTE-plus CRS channel estimator (also called channel estimation function in the rest of this paper). Section IV describes the methodology of the proposed approach with the related work-flow, the CNN architecture used in our study and the related CNN hyper-parameters to be tuned. In addition, section IV also explains the evaluation metrics and the computing platform used in this study. Section V provides the results with an analysis of the impact of the parameters of the approach of the CNN and a comparison with other approaches commonly used in the research literature. Finally, Section VI states the conclusions and indicates potential future developments. A list of the most significant acronyms used in this paper is shown in Table I.

TABLE I: Acronyms used in this paper.

Acronym	Definition
AUC	Area Under Curve
BER	Bit Error Rate
CP	Cyclic Prefix
CNN	Convolutional Neural Network
CRS	Cell Specific Reference Signal
DL	Deep Learning
FDD	Frequency Division Duplexing
FFT	Fast Fourier Transform
FIFO	First In First Out
FPGA	Field Programmable Gate Array
KNN	K Nearest Neighbor
LQI	Link Quality Indicators
LTE	Long Term Evolution
ML	Machine Learning
OAI	Open Air Interface
OFDM	Orthogonal Frequency-Division Multiplexing
PSS	Primary Synchronization Signal
RB	Resource Block
ROC	Receiver Operative Characteristics
RSSI	Received Signal Strength Indication
SISO	Single-Input Single-Output
SSS	Secondary Synchronization Signal
TDD	Time Division Duplexing
UERS	UE Specific Reference Signal
USRP	Universal Software Radio Peripheral
WII	Wireless Interference Identification

II. Related Work

This section reports briefly on the application of DL to wireless communication and on the detection/identification of wireless interference in particular. The section is structured in the sub-section II-A, which describes the application of ML for the identification of wireless interferences, in the sub-section II-B, which provides an overview on the application of DL and particularly CNN to wireless communication problems and to wireless interference identification and sub-section II-C, which summarizes the specific aspects of this study in comparison to the identified research literature.

A. Application of ML to wireless interference identification

Some initial studies used ’shallow’ machine learning algorithms to identify wireless interferences. For example, the authors in [7] have applied Support Vector Machine (SVM) to the detection of wireless interference in an industrial setting where signals based on the IEEE 802.11 family of standards are present (802.11 signals are also considered in this study). A SVM classifier is used to process signal features extracted from Received Signal Strength Indication (RSSI) traces to identify the source of external interference. In a similar way, [8] proposes an approach for wireless interference detection using spectral features and supervised machine learning with Decision Tree, Random Forest and SVM. Random Forest achieved the best performance in most of the scenarios. In [9], the authors have used already available indicators of channel estimation like Link Quality Indicators (LQI) and RSSI to detect the presence of IEEE 802.11 interference. Another study, which uses LQI for wireless interference detection, is [10], where the authors have investigated the use of the Random Forest algorithm to detect jamming signals to a IEEE 802.11 communication links in an industry settings. This study proposes an approach similar to these previous studies using observables derived from existing functions in wireless communication equipment (e.g., RSSI and LQI), but it is based on LTE-plus communication systems and it is based on a DL implementation with CNN.

B. Application of DL to wireless interference identification

We focuses now on the application of DL to this type of problem.

A comprehensive survey on the application of DL to wireless communication is reported [3] and the physical layer is one of the main categories where various algorithms of DL like CNN or autoencoders have been used. In particular, the authors have used DL for channel coding in [11] managing to improve the Bit Error Rate (BER) in comparison to conventional means. In [12], the authors have applied CNN to implement modulation classification, which is another important function for spectrum management, signal identification and electronic warfare. Wireless device classification (also called RF fingerprinting) using CNN has also been achieved with considerable improvement to ML in [13] and [14].

In relation to the specific application of DL to wireless interference detection and identification, the few reported studies are relatively recent (from 2017 onwards). One of the first studies to apply DL to the detection of wireless interference is [4], where the authors have applied CNN to a problem of coexistence among IEEE 802.11 b/g, IEEE 802.15.4 and IEEE 802.15.1 signals with different modulation schemes. Then, a total of 15 classes are used for the identification problem. The data set was created using laboratory equipment as in this study. The CNN was applied both to the original time representation of the signal (expressed in IQ format) and the frequency representation (i.e., through Fast Fourier Transform (FFT)) with the frequency representation achieving a better performance. This paper uses a similar CNN architecture to the one used in [4]. Another study for WiFi interference detection using CNN was performed in [15], where for cross-technology interference the approach manages to achieve an accuracy above 90 %. In a similar way, the authors of [16] have applied CNN to a problem of WII where both WiFi and LTE standards were used with 5 classes of identification. As in [4] and in this paper, the data set is created by the authors using a laboratory settings with Universal Software Radio Peripheral (USRP) and WiFi devices using a conducted setting.

The application of CNN to WII has also been successfully attempted in [17], where the authors investigate the impact of low-precision weights as well as activations in CNN and they proposed a number of mitigation techniques to address this issue, which can become important for the practical implementation of WII using CNN in commercial hardware platforms. The authors manage to demonstrate that the proposed approach using 3-bit and 4-bit precision networks is not only more time efficient but it also leads to performance improvements as compared to their full precision counterparts with standard network architecture.

Another improvement in the application of CNN to WII was performed in [18] where a combination of transformers and CNN was used to enhance the performance of the WII. The new approach (called BGCNN) combines the advantages of CNNs in extracting low-level features and the merits of transformers in establishing long-range dependencies and it managed to provide a higher performance than the basis application of CNN. An additional enhancement on the application of CNN to WII is proposed in [19] where time frequency representations are combined with CNN to enhance the performance of WII. The authors demonstrate that the proposed approach has a higher identification accuracy in comparison to the application of CNN to the original time representation of the signal. In a similar way, the authors of [20] have used CNN to the analysis of spectrograms expressed as red green blue (RGB) images where the interference is present together with the victim signal. Another application of CNN to WII is presented in [21], where the identification problem is related to Bluetooth, ZigBee and WiFi all operating in the 2.4 GHz band for a total of 7 classes. The LSTM and ResNet DL algorithms have been also used for comparison, with the CNN achieving a competitive performance. Moreover, the authors have attempted to optimize the application of CNN by selecting specific spectral bands and obtaining a higher identification accuracy with smaller computational time.

The authors proposed in [22] a semi-supervised WII approach, which combines temporal ensemble technique with a CNN network to exploit unlabeled data to improve the identification performance. The approach is applied to the same data set created by the authors of [4] achieving a better identification performance and mitigating the problem of the unlabeled data.

C. Summary of the contributions of this study in comparison to the research literature

As shown in the literature above, CNN has been applied with considerable success to the problem of WII with various improvements related to the selection of the frequency bands and the optimization for a practical deployment in computing platform. The approach proposed in this study is based on a similar application of CNN to WII where the data set is mostly related to the interference between LTE and WiFi signals. The most significant difference of this study and the ones proposed in literature is that the CNN is applied not directly to the IQ data collected directly at the receiver side, but by exploiting the channel estimator, which is always implemented in wireless communication systems. In particular a customized LTE-plus channel estimator is implemented with a FPGA to generate the estimates on channel conditions, which are then fed to a CNN. To the knowledge of the authors, this is a novel approach in comparison to the literature and it is much more efficient than processing the original signal representation. It follows a similar reasoning to the use of LQI and RSSI shown in [9], [10] but the channel estimator output is more sophisticated.

III. Test bed for data set generation

This section is structured in four sub-sections. Sub-section III-A describe the overall radio frequency laboratory configuration including the LTE-plus victim system implemented with the FPGA and the signal generators used to generate the interference. Sub-section III-B describes the main concepts of the LTE channel estimator, while sub-section III-C describes the actual FPGA implementation. Finally, sub-section III-D describes the specific data sets created with this radio frequency laboratory configuration and introduces the data sets notations used in the rest of this paper.

A. Architecture of the radio frequency laboratory set-up

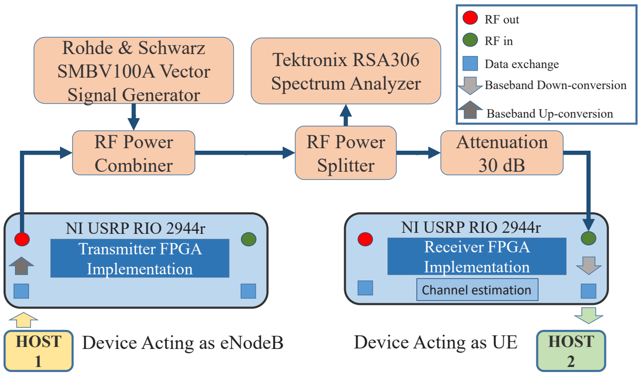
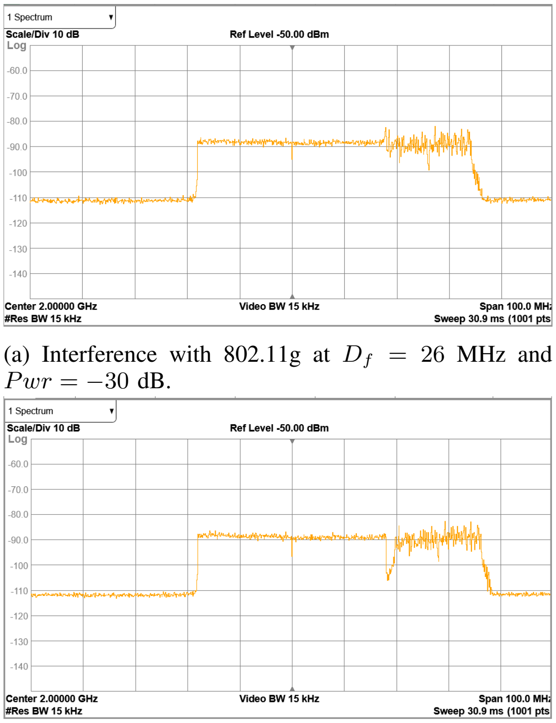


Fig. 1: Schema of the test bed.

The schema of the RF testbed used to generate the adjacent wireless interference data sets used in our study is presented in Figure 1. This is a conducted (i.e., the RF signal is transmitted through cables) testbed where the two USRPs (i.e., NI USRP RIO 2944r devices with 160 MHz of bandwidth) are connected to two different host computers, called Host 1 and Host 2 to implement the LTE-plus victim service with a center frequency at 2 GHz and 40 MHz bandwidth. The interference signals are generated by the signal generator SMBV100A on the basis of three different standards (i.e., 802.11g, LTE TDD and LTE FDD). Each interference signal has a 20 MHz bandwidth and it is transmitted with a center frequency slightly greater (the spectral distance parameter is called D_f in the rest of this paper) than the center frequency of the LTE-plus victim service to generate adjacent channel interference as some of the RF power transmitted by the interference signal is present in the operating RF spectrum of the victim service. The power of the generated interference signal is controlled by another parameter, which is called Pwr in the rest of this paper. In the receiver USRP (i.e., device acting as UE), the RF signal is down-converted to baseband from the operational carrier frequency (e.g., 2 GHz) used in this study. Figures 2a gives a visual representation of the 802.11g interferer positioned at a spectral distance D_f respectively of 26 MHz and 28 MHz from center frequency of the victim service as it is captured by the spectrum analyzer present in the test bed. Note: The use of the center frequency of the LTE-plus victim service at 2 GHz was adopted because LTE operates in this frequency band in most of the regions of the world. On the other side, we also wanted to explore and analyze the impact and detection of wireless interference among different wireless communication standards (notably 802.11g and LTE FDD and TDD) as it is often investigated in literature [16], even if the spectrum allocation of 802.11g (e.g., operating in the 2.4 GHz) is different from the considered spectrum allocation in this study. Then, to take in consideration all the three wireless interference scenarios at the same time, we have considered a context, which is not fully supported by the current spectrum regulation, even if it may not be excluded that such spectrum allocation could be defined in the future creating risks of adjacent cell interference like the ones investigated in this paper. In-fact, this study was carried out in the context of pre-normative research, which investigates RF spectrum problems not directly associated to existing RF spectrum regulations, but which serves the purpose of anticipating and proposing mitigation techniques for such potential future problems. We would also like to highlight that the FPGA implementation of the proposed approach using the channel estimator is in base-band and the information on the operating carrier frequency is not used by the channel estimator.

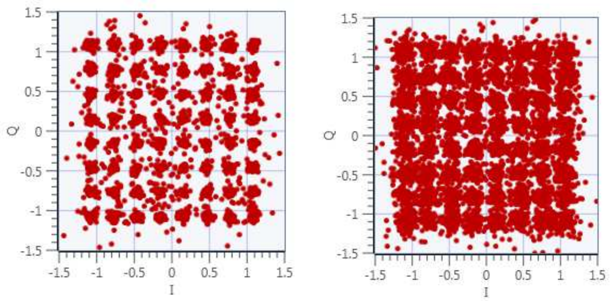


(a) Interference with 802.11g at $D_f = 26$ MHz and $Pwr = -30$ dB.

(b) Interference with 802.11g at $D_f = 28$ MHz and $Pwr = -30$ dB.

Fig. 2: Spectrum analyzer captures of the different interference conditions.

The impact of the adjacent channel interference is to lower



(a) Interference with 802.11g at $D_f = 26$ MHz and $Pwr = -30$ dB. (b) Interference with LTE FDD at $D_f = 26$ MHz and $Pwr = -30$ dB.

Fig. 3: Constellation diagrams for different interference conditions.

the quality of the processed signal by the receiver. Apart from the channel estimator, which is exploited in this study, the impact of the interference can be visually seen by the constellation diagrams, which are the result of the demodulation process by the LTE receiver. Figures 3a and 3b below show respectively the impact of the 802.11g interferer and the LTE FDD interferer for $D_f = 26$ MHz and $Pwr = -30$ dB. It can be noted that the different types of interferer have a different impact on the quality of the processed signal even if the D_f and Pwr have the same values. This justifies the analysis using different types of interferer. Then, we have three different dimensions to characterize the interferer: wireless standard, spectral distance D_f and power Pwr in dB. The whole data set is described in detail in section III-D.

In the test bed, it was used the modified version of the LTE downlink configuration with a bandwidth of 40 MHz and a subcarrier spacing of 30 kHz ($\mu = 1$). In comparison to the LTE numerology (subcarrier spacing and symbol length), the most outstanding difference is that LTE New Radio (NR) supports multiple different types of subcarrier spacing while in LTE there is only one type of subcarrier spacing with 15 kHz. The original LTE Application Framework implements parts of the 3GPP-LTE release 10 downlink and uplink physical layer transmitter and receiver components. To reduce the complexity of this application framework, the following settings are fixed and can only be changed by modifying the FPGA and the host code design:

1) 20 MHz bandwidth. The authors modified the subcarrier spacing from 15 to 30 kHz to increase the bandwidth up to 40 MHz. For this reason, this modified release of the basic LTE is called LTE-plus in this paper.
2) For Time Division Duplex (TDD) operation: uplink/downlink configuration 5, special subframe configuration 5.
3) Normal cyclic prefix.
4) Resource mapping for two transmitting antennas (only the first antenna is used).
5) For the downlink configuration, the Primary Synchronization Signals (PSS) is transmitted only once per radio frame (10 ms periodicity instead of 5 ms periodicity).
6) The SSS (Secondary Synchronization Signals) is not implemented in our framework. Thus the PSS only is utilized for synchronization purpose.

B. Channel estimator model

In wireless communication systems, the receiver needs to remove or at least mitigate the effects of the distortions in the wireless propagation channel. This task can benefit by the process of *channel estimation* to characterize the channel features. This is implemented using the reference signals, since they are made up of data known to both the transmitter and the receiver. Then the receiver can figure out how the communication channel distorts the data by comparing the decoded received reference signals and the predefined reference signals, and it can use the result of this comparison to equalize (in real-time) the received user data and perform the channel estimation. There are many different ways for implementing a channel estimator, but the fundamental concepts and steps are similar: a) Set a mathematical model to correlate the “transmitted signal” and the “received signal” using a “channel matrix”. b) Transmit a known signal (we normally called this as “reference signal” or “pilot signal”) and detect the received signal. c) By comparing the transmitted signal and the received signal, the elements of the related channel matrix can be calculated. In this study a Single Input Single Output (SISO) model is used.

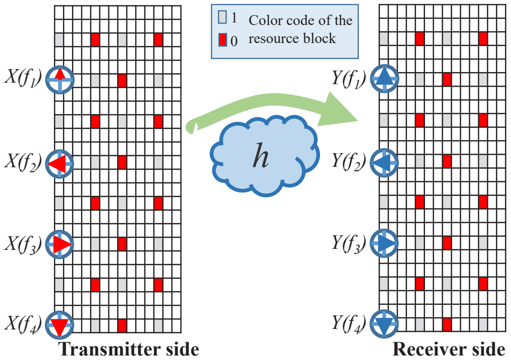


Fig. 4: Resource maps used for the channel estimator.

In this study, the authors have implemented in a FPGA a customized version of the CRS present in LTE receiver equipment as described in sub-section III-C. The customization does not modify the basic functionality of the CRS in LTE but it allows to collect the output of the CRS so that it can be given as input to the CNN. The CRSs were defined in LTE from the first releases as to provide a stable, dense, and always present downlink resource, which the UE uses for multiple and widely different purposes. These purposes are fine time and frequency synchronization, estimation of large-scale channel parameters, measurements for CSI feedback, as well as serving as demodulation reference for receiving the downlink physical channels PBCH, PDSCH, PDCCH, PCFICH, and PHICH [23]. Then, they could also be used to detect and recognize the presence of wireless interference as it was done in this study. Because the CRS is present in all downlink subframes and covers the entire served cell of a LTE base station, the UE can continuously track the channel, leading to excellent channel estimation performance and robustness for the CRS-based transmission schemes. On the other side, the presence of this "always on" CRS transmission is not energy efficient because energy is consumed even there is no provision of services to UEs in the cell. In addition, the CRS from neighboring cells may generate interference to PDSCH and PDCCH if the CRS has a subcarrier offset compared to the CRS of the serving cell, even if the interfering cell is not transmitting any data (this was also one of the motivation for this study, because it provides an useful tool to detect mutual LTE interference). Due to these energy consumption problems 3GPP LTE release 10 introduced the concept of CSI-RS with the addition of up to 8-layer spatial multiplexing where the CRS are not transmitted continuously. Then, in 5G NR for similar reasons, there are no CRS-like signals, but the same concept of CSI-RS is reused and extended in NR to provide support for beam management and mobility in the connected mode. Then, the concepts presented in this study can be readily applied to 5G NR as well to detect and identify wireless interferences.

This study uses the CRS instead of another popular reference signal UE-specific Reference Signal (UERS) for various reasons. The first reason (as described above) is that CRS is always configured in the LTE network while UERS may not be activated in the network as UERS are transmitted only on the resource blocks upon which the corresponding PDSCH is mapped [24]. The second reason is that UERS were introduced for more flexible and advanced multi-antenna transmission techniques [25] and they are suited for MIMO while this study is focused on SISO. The third reason (related to the first) is that UERS could be configured in the different antenna points (e.g., AP7 and AP9) where is the number of layers used for transmission of the PDSCH, while the CRS are transmitted over the entire bandwidth in all downlink subframes [26]. This can make the implementation of the approach more complex in the FPGA because of the variability in the UERS configuration and location. In addition, in a full configuration, there could a large number of UERS to be processed with a significant greater computing effort. On the other side, future developments of this study related to MIMO configurations could take in consideration the UERS if the challenges mentioned above are addressed (see Section VI).

Figure 4 shows the resource maps of the SISO reference signals for the LTE system implemented in the radio frequency laboratory used in this study. The vertical axis in the resource map represents the frequency domain. So each reference signal is indexed with f_1, f_2, $f_3 ... f_n$. Each reference symbol is a complex number (IQ data). Each reference symbol on the left (transmission side) is modified (distorted) by the wireless propagation channel to each corresponding symbol on the right (receiver side). Channel estimation is the process of finding the correlation between the array of complex numbers on the left and the array of complex numbers on the right. Since there is only one antenna (i.e., SISO), the system model for each transmitted reference signal and the received reference signal can be formalized as follows: Y() represents the array of received reference signals, X() represents the array of the transmitted reference signals and h() represents the array of the channel coefficients (i=1..n) (n= 200 in this study).

$$Y(f_i) = H(f_i) \cdot X(f_i), i = 1..4. \tag{1}$$

X() is known because it is configured in the transmitter side and Y() is measured/detected from the receiver. With this information, we can easily calculate the coefficient's array as shown in the equations below:

$$H(f_i) = Y(f_i) \cdot X^H(f_i), i = 1..4. \tag{2}$$

Where $X^H(f_n)$ is the Hermitian of $X(f_n)$. In mathematics, a Hermitian matrix (or self-adjoint matrix) is a complex square matrix that is equal to its own conjugate transpose, that is, the element in the i-th row and j-th column is equal to the complex conjugate of the element in the j-th row and i-th column, for all indices i and j. In this SISO case, because the $X(f_n)$ matrix is actually a 1x1 matrix (single complex number) the Hermitian is simply the conjugate of that complex number. Then, all the channel coefficients for the locations where the reference signals are located, are known. But we need the channel coefficients of all the locations including those points where there are no reference signals. This means that we need to figure out the channel coefficients for those locations with no reference signals. The most common way to do this is to interpolate the measured coefficient's array. At this point, knowing the channel estimates, it is also possible to evaluate the statistical noise properties of the channel, which in this study are used to extract the information on the presence of wireless interferences.

In the following paragraphs, we describe the LTE implementation of the equations identified above, while sub-section III-C describes the FPGA implementation of the channel estimator performed by the authors.

LTE supports multiple numerologies and consequently there are multiple possible implementations of the resource maps (also called resource grids). There is one resource grid for each numerology and carrier. There is a set of resource grids per transmission direction (uplink or downlink). There is one resource grid for a given antenna port p, subcarrier spacing configuration μ, and transmission direction (downlink or uplink).

LTE defines a frame to be 10 milliseconds (ms) in duration. In LTE, each frame is divided into 10 subframes of 1 ms each and every 1 ms subframe is divided into two slots of 0.5 ms (each slot containing 7 symbols at normal CP). In 5G NR, the slot definition is slightly different and this may generate confusion. In our study, the number of symbols for each slot is fixed at 14 (normal CP) and the frame duration is held constant at 10 ms for any μ value and the slot duration changes. In the modified LTE framework used (i.e., LTE-plus) in this study, we set $\mu = 1$ and the nominal bandwidth is set to 40 MHz because we have 100 Resource Block (RB)s and a carrier spacing of 30 KHz. The numerology table extracted from 3GPP TS 38.211 [27] with the values of μ and the corresponding numbers of slots is shown in Table II:

Each RB is composed by 12 sub-carriers, thus we have in total 1200 sub-carriers in the resource map (30 KHz * 1200 = 36 MHz). The frame duration is always 10 ms and we have 20 slots composed by 14 symbols (Normal CP)

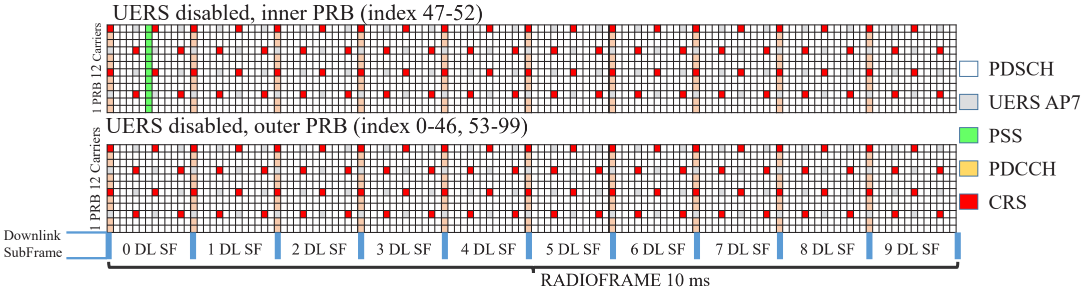


Fig. 5: Supported LTE DL Resource Grid for Frame Structure Type 1 (LTE with 20 MHz bandwidth and FDD).

TABLE II: Numerology extracted from 3GPP TS 38.211. Table 4.3.2-1

μ	N^{slot}_{symb}	$N^{frame,\mu}_{slot}$	$N^{subframe,\mu}_{slot}$
0	14	10	1
1	14	20	2
2	14	40	4
3	14	80	8
4	14	160	16

and every sub-frame contains two slots. Figure 5 shows the resulting LTE downlink resource grid for frame structure type 1 (for the Frequency Division Duplexing (FDD) case) with the supported physical downlink channels and signals for our LTE customized framework used in our study. In Figure 5, we note the main LTE channels: Primary Synchronization Signal (PSS) for synchronization, the reference signal Cell Specific Reference Signal (CRS), the data channel Physical Data Shared Channel (PDSCH), the control channel Physical Control Channel (PDCCH). The SSS (Secondary Synchronization Signal) is not implemented in our framework. Instead the UERS (User Specific Reference Signals) are implemented but not utilized in this study for the reasons described above.

C. Design and implementation of the FPGA channel estimator

This study uses a specific FPGA implementation of the LTE channel estimator developed by some of the authors. This subsection has the purpose to describe the FPGA implementation. The customized FPGA exchanges the base-band data with the RF interface using target-scoped FIFOs. The processing on the FPGA has advantages in comparison to a full software implementation as the Open Air Interface (OAI) streaming project where the digital baseband data is sent to or received from the host which is then responsible for all channel encoding and decoding. In fact, the FPGA implementation provides lower latency than OAI and therefore enables real-time physical layer processing.

With reference to the Figure 6, the transmitter loop in the FPGA implementation receives the payload data from the Host 1 via a Direct Memory Access (DMA) First Input First Output (FIFO), performs channel encoding and generates the transmitted (TX) baseband signal, which is passed to the RF output loop for the up-conversion on the first USRP. The RF input loop is instantiated on the second USRP. It performs the down-conversion of the received (RX) baseband signal that is passed to the receiver loop for channel decoding. The decoded transport blocks are sent to the Host 2 using a DMA FIFO. The study presented in this paper is mostly focused on the identification of wireless interferences using the DL RX IQ processing. Then, the specific focus is the “DL RX PHY” block which implements, among other things, the channel estimation. This module reads the radio-frame aligned signal in time domain and outputs the channel-equalized subcarriers that are associated to the physical channels. As shown in Figure 7, it includes the functional blocks described in the following paragraphs.

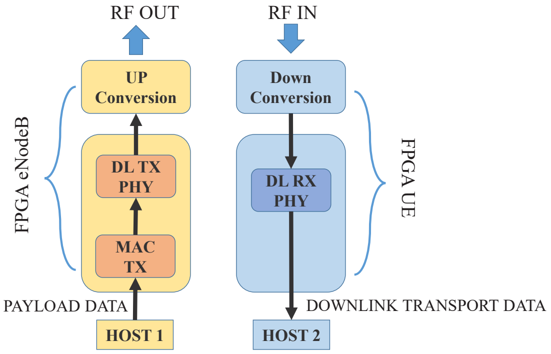


Fig. 6: High level view of the FPGA implementation of the channel estimator.

An internal FIFO memory is used to decouple the incoming samples from the rest of the processing chain. The throttle control module waits until enough samples for one complete Orthogonal Frequency-Division Multiplexing (OFDM) symbol (FFT size + CP) are available before it passes them as a consecutive stream to the next modules. The subsequent module is the Cyclic Prefix (CP) removal, which removes the valid flag from the samples belonging to the CP. The 2048 remaining samples are sent to a Xilinx FFT module. The output of the FFT are the 2048 subcarriers in the frequency domain. The resource mapper first selects the 1200 allocated subcarriers by removing the surrounding whitespaces and the DC carrier in the center. After-wards, it generates the timing information for each sample and the resource grid by marking each sample for its corresponding channel by using a Boolean cluster. The resource mapping is based on a fixed frame structure configuration described in the LTE specifications. All subsequent modules use this boolean cluster, with elements for each LTE channel, to determine if this sample is relevant. The

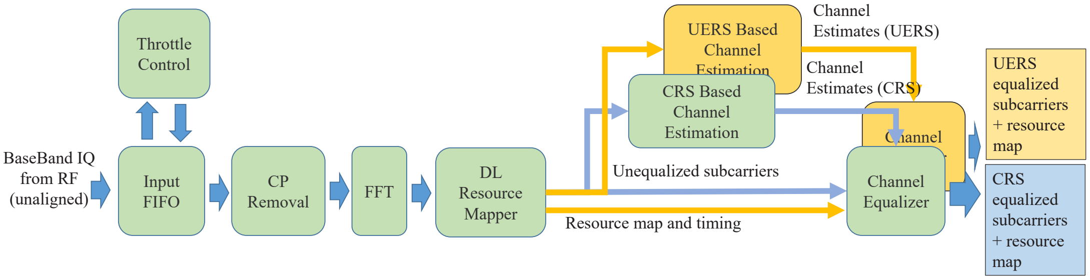


Fig. 7: FPGA implementation of the channel estimator.

FFT output data is fed into two separate channel estimation blocks running in parallel. The first channel estimation block is based on the CRS. The channel estimate values are calculated by conjugate complex multiplications. A linear interpolation is applied in the frequency domain between adjacent reference symbols. The second channel estimation block is based on the UERS, but they are not used in this paper.

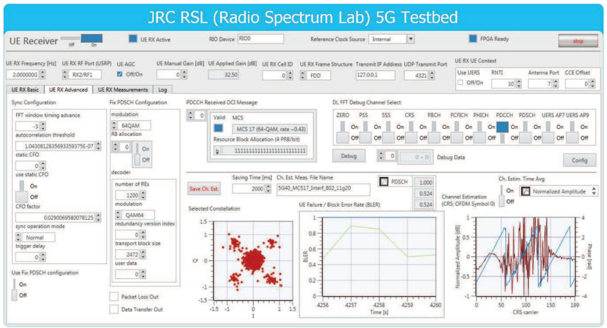
On the edges of the symbol, the nearest estimated value is replicated (zero order hold). OFDM symbols not containing CRS sequences rely on the last channel estimation (zero order hold in time). We implemented in addition a linear interpolation in the time domain of the channel estimate for OFDM symbols not containing CRS sequences. This option can be activated from the host instead of the zero order hold in time. The channel estimates are delivered sample by sample to the channel equalization modules parallel to the data. The channel equalization determines the result from the data sample d and the channel estimate e by using the following equation:

$$\frac{d \cdot e^*}{|e|^2} = \frac{d}{\sqrt{e \cdot e^*}} \cdot \frac{e^*}{\sqrt{e \cdot e^*}} \tag{3}$$

In conclusion, going back to Figure 5, it can be seen that the CRS, represented by square red dots in the resource grid are 200 (over 1200 carriers) for each symbol in which they are present.

As mentioned before, the channel estimate values are calculated in real-time by the FPGA and are not normally available at the host level in the format needed for this study. The solution is to insert in the FPGA a so-called *probe* to retrieve the desired data and pass them to the host. On the other side, implementing a probe in the FPGA is not so easy because it is mandatory to satisfy the following conditions:

- As this is actually a modification of the FPGA source code, the probe must not perturb the original behavior of the FPGA. A probe is normally based on a so-called Target-to-Host FIFO. If it is not placed in the proper way in the source code, especially in a time-critical part as the channel estimation function, it may conduct to timing violation errors at complication level and/or incorrect behaviour of the FPGA at run-time.
- The throughput of the data must be limited in some way, because the amount of information produced by the FPGA per unit of time is enormously greater than the host consuming speed and this can create FIFO overflow errors. In this study, it was decided to send to the host only the CRS values for the symbol 0 of each slot. Remembering that the frame duration is always 10 ms and we have 20 slots composed by 14 symbols (Normal CP, $\mu = 1$) and for each symbol we have 200 CRSs, the throughput is: (200 CRS * 20 Slots) / 0.01 sec = 400000 Samples/Second. The total acquisition duration is 2 seconds for each step.


JRC RSL (Radio Spectrum Lab) 5G Testbed

UE Receiver Off On
UE RX Active
RIO Device RIO0
Reference Clock Source Internal
FPGA Ready
stop

UE RX Frequency [Hz] 2.000000G
UE RX RF Port (USRP) RX2/RF1
UE AGC ☑ Off/On
UE Manual Gain [dB] 0
UE Applied Gain [dB] 32.50
UE RX Cell ID 0
UE RX Frame Structure FDD
Transmit IP Address 127.0.0.1
UDP Transmit Port 4321
UE RX UE Context: Use UERS □ Off/On; RNTI 10; Antenna Port 7; CCE Offset 0

UE RX Basic | UE RX Advanced | UE RX Measurements | Log

Sync Configuration
FFT window timing advance -3
autocorrelation threshold 1.0430812835693359375E-07
static CFO 0
use static CFO On Off
CFO factor 0.0290069580078125
sync operation mode Normal
trigger delay 0
Use Fix PDSCH configuration On Off

Fix PDSCH Configuration
modulation 64QAM
RB allocation 0 On Off
decoder
number of REs 1200
modulation QAM64
redundancy version index 0
transport block size 2472
user data 0
□ Packet Loss Out
□ Data Transfer Out

PDCCH Received DCI Message 0
Valid MCS: MCS 17 (64-QAM, rate ~0.43)
Resource Block Allocation (4 PRB/bit) 1111111111111111111111111

DL FFT Debug Channel Select
ZERO PSS SSS CRS PBCH PCFICH PHICH PDCCH PDSCH UERS AP7 UERS AP9
Debug 0 0 + 0i Debug Data
Config

Save Ch. Est.
Saving Time [ms] 2000
Ch. Est. Meas. File Name 5G40_MCS17_Interf_802_11g20
PDSCH 1.000 0.524 0.524
Ch. Estim. Time Avg On Off
Channel Estimation (CRS; OFDM Symbol 0) Normalized Amplitude

Fig. 8: Front panel of downlink receiver (DL RX) with an interference signal – UE (User Equipment) RX Advanced Tab.

The visual representation of the front panel of the downlink receiver (DL RX) with the presence of an interference signal is shown in Figure 8, where it can be seen an example of the CRS channel estimate.

We summarize again in this paragraph the configuration of the downlink transmitter and receiver: the victim system is implemented with the NI USRP RIO 2944r set to transmit a LTE plus signal with a carrier frequency of 2 GHz, 40 MHz bandwidth with Frequency Division Duplex (FDD). The Modulation and Coding Scheme (MCS) was set to a value of 17 (i.e., 64 QAM) with a rate of 0.43 and all the resource blocks activated: Resource Block Allocation (RBA) set to 100 PRBs. The receiver side where the channel estimation function is executed is also implemented with a NI USRP RIO 2944r.

An example of the final output of the CRSs provided by the channel estimator is shown in Figure 9 for the normalized amplitude and the phase component (in radians). Basically, the output consists in two time series (amplitude and phase) of length 200. The analysis conducted by the authors have shown that the amplitude component provides much better classification results than the phase component or the combination of the amplitude and the phase. For this reason, only the amplitude component will be used in the rest of this paper.

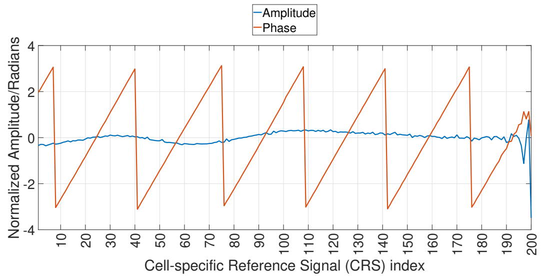


Fig. 9: Amplitude and phase output from the channel estimator.

D. Data set structure and composition

Three different data sets were used in this study with different types of wireless adjacent band interference. As described before, each data set is composed of 4000 CRS channel estimates of length 200 (one for each CRS) for a total acquisition time of 2 seconds for each type of wireless interference for a total of 48000 samples in each data set, because there are 12 different types of wireless interference in each data set. As mentioned in the previous sub-section III-A, we selected a frequency band for the LTE-plus victim service (i.e., 2 GHz), which is not currently allocated (in the current spectrum allocation) for the operation of 802.11g. This deviation from the current spectrum regulations was done to investigate potential future scenarios where all the three interference services (e.g., 802.11g, LTE-FDD, LTE-TDD) may be present in the same frequency bands in a spectrum allocation where adjacent band interference to the LTE-plus victim service may be generated. This study was done in the context of pre-normative research as it may not be excluded that such spectrum allocation could be defined in the future and such pre-normative analysis could be useful to propose detection and mitigation techniques for this future case.

Then, Table III identifies the different types of wireless interferences for the first data set where WiFi 802.11g is the interferer. In Table III, BW is the BandWidth of interferer, Df is the distance in the spectral domain of the interferer signal to the main LTE victim signal operating at a center frequency of 2GHz (i.e., to generate adjacent band interference), Pwr is the power level of the interferer. In a similar way, Table IV describes the list of LTE TDD interferers of the second data set and Table V describes the list of LTE TDD interferers of the third data set. Then, each data set is used for a supervised classification problem with 12 classes and 4000 samples for each class.

TABLE III: Data Set 1: LTE with 802.11g interferer

Id	Description of the type of wireless interference
DS1RFI1	802.11g with 20 MHz BW, Df=26 MHz, Pwr=-30 dB
DS1RFI2	802.11g with 20 MHz BW, Df=26 MHz, Pwr=-35 dB
DS1RFI3	802.11g with 20 MHz BW, Df=26 MHz, Pwr=-40 dB
DS1RFI4	802.11g with 20 MHz BW, Df=26 MHz, Pwr=-45 dB
DS1RFI5	802.11g with 20 MHz BW, Df=28 MHz, Pwr=-30 dB
DS1RFI6	802.11g with 20 MHz BW, Df=28 MHz, Pwr=-35 dB
DS1RFI7	802.11g with 20 MHz BW, Df=28 MHz, Pwr=-40 dB
DS1RFI8	802.11g with 20 MHz BW, Df=28 MHz, Pwr=-45 dB
DS1RFI9	802.11g with 20 MHz BW, Df=30 MHz, Pwr=-30 dB
DS1RFI10	802.11g with 20 MHz BW, Df=30 MHz, Pwr=-35 dB
DS1RFI11	802.11g with 20 MHz BW, Df=30 MHz, Pwr=-40 dB
DS1RFI12	802.11g with 20 MHz BW, Df=30 MHz, Pwr=-45 dB

TABLE IV: Data Set 2: LTE with LTE TDD interferer

Id	Description of the type of wireless interference
DS2RFI1	LTE TDD with 20 MHz BW, Df=26 MHz, Pwr=-30 dB
DS2RFI2	LTE TDD with 20 MHz BW, Df=26 MHz, Pwr=-35 dB
DS2RFI3	LTE TDD with 20 MHz BW, Df=26 MHz, Pwr=-40 dB
DS2RFI4	LTE TDD with 20 MHz BW, Df=26 MHz, Pwr=-45 dB
DS2RFI5	LTE TDD with 20 MHz BW, Df=28 MHz, Pwr=-30 dB
DS2RFI6	LTE TDD with 20 MHz BW, Df=28 MHz, Pwr=-35 dB
DS2RFI7	LTE TDD with 20 MHz BW, Df=28 MHz, Pwr=-40 dB
DS2RFI8	LTE TDD with 20 MHz BW, Df=28 MHz, Pwr=-45 dB
DS2RFI9	LTE TDD with 20 MHz BW, Df=30 MHz, Pwr=-30 dB
DS2RFI10	LTE TDD with 20 MHz BW, Df=30 MHz, Pwr=-35 dB
DS2RFI11	LTE TDD with 20 MHz BW, Df=30 MHz, Pwr=-40 dB
DS2RFI12	LTE TDD with 20 MHz BW, Df=30 MHz, Pwr=-45 dB

TABLE V: Data Set 3: LTE with LTE FDD interferer

Id	Description of the type of wireless interference
DS3RFI1	LTE FDD with 20 MHz BW, Df=26 MHz, Pwr=-30 dB
DS3RFI2	LTE FDD with 20 MHz BW, Df=26 MHz, Pwr=-35 dB
DS3RFI3	LTE FDD with 20 MHz BW, Df=26 MHz, Pwr=-40 dB
DS3RFI4	LTE FDD with 20 MHz BW, Df=26 MHz, Pwr=-45 dB
DS3RFI5	LTE FDD with 20 MHz BW, Df=28 MHz, Pwr=-30 dB
DS3RFI6	LTE FDD with 20 MHz BW, Df=28 MHz, Pwr=-35 dB
DS3RFI7	LTE FDD with 20 MHz BW, Df=28 MHz, Pwr=-40 dB
DS3RFI8	LTE FDD with 20 MHz BW, Df=28 MHz, Pwr=-45 dB
DS3RFI9	LTE FDD with 20 MHz BW, Df=30 MHz, Pwr=-30 dB
DS3RFI10	LTE FDD with 20 MHz BW, Df=30 MHz, Pwr=-35 dB
DS3RFI11	LTE FDD with 20 MHz BW, Df=30 MHz, Pwr=-40 dB
DS3RFI12	LTE FDD with 20 MHz BW, Df=30 MHz, Pwr=-45 dB

IV. METHODOLOGY

This section presents the overall workflow, the CNN architecture, the metrics of evaluation and the computing platform used to execute the CNN algorithms.

A. Workflow

The overall workflow is presented in Figure 10, where the main steps are identified. The workflow is relatively simple: after that the connection is set-up and a videostreaming (a video of 30 minutes duration) is activated to generate traffic, different wireless interference conditions are generated as described in sub section III-D. Then, the channel estimator implemented in the receiver FPGA (as described in subsection III-C) provides the channel estimates for each wireless interference condition. The channel estimates are represented by time series of length 200. Then, the entire data set is used for supervised learning with a CNN described in subsection IV-B (and the other machine learning algorithms for comparison). Section IV-B also describes how the data set is partitioned in one part for testing (a quarter of the entire data set) and one part for training (three quarters of the entire data set including the portion used for validation, which is one tenth of the training data set). The results obtained for accuracy and F-score are averaged to obtain the final results presented in Section V.

An alternative approach is also considered for comparison of the main proposed approach. In this alternative approach, the channel estimation is based on the constellation diagrams created with the IQ values which are provided as an output of the FPGA demodulator. This approach has been proposed

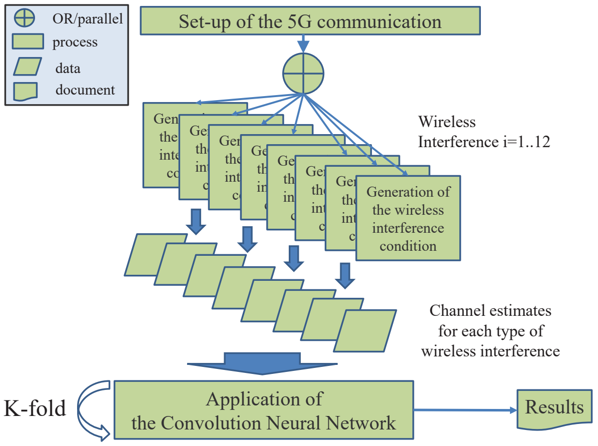

OR/parallel
process
data
document
Set-up of the 5G communication
Wireless
Interference i=1..12
Generation of the wireless interference condition
Channel estimates for each type of wireless interference
K-fold
Application of the Convolution Neural Network
Results

Fig. 10: Overall workflow.

recently in various papers like [28] where the objective is actually the modulation classification using CNN in presence of wireless interference and where constellation diagrams are used as in input to the CNN in a similar way to this paper. Then, the objective of [28] is complementary to the objective in this paper. In this alternative approach based on the constellation diagrams the challenge is to reduce the size of the IQ constellation diagrams to make more efficient the application of CNN (the larger the size of the image and the longer is the processing time of the CNN classification). Different sub-approaches have been used in literature for such dimensionality reduction. In this paper, the alternative approach is based on the method used in [29] and [30], where the constellation diagrams are transformed to 2D histograms, which greatly reduce the size of the images to the number of bins used in the 2D histogram definition. In this method, the number of bins is obviously a parameter to be investigated and the comparison results are presented in Section V. A simple visual representation of the overall schema is provided in Figure 11, where it is shown that the IQ constellation diagrams are transformed with the application of the 2D histograms and a defined number of bins (in the example this value was set to 64) and then given as an input to the CNN. This alternative approach is called CNN-CONST in the rest of this paper.

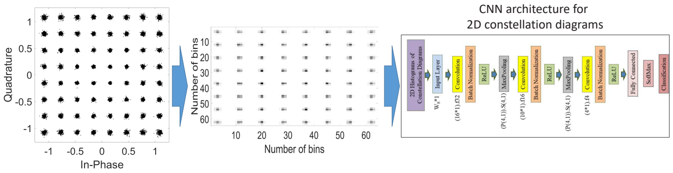


Fig. 11: Alternative approach based on the constellation diagrams output by the FPGA demodulator.

B. Convolutional Neural Network architecture

A three layers CNN was used in the study, whose architecture is shown in Figure 12. Because the input data is a structure of size 1×200, a 1-D CNN was used. The optimization of the

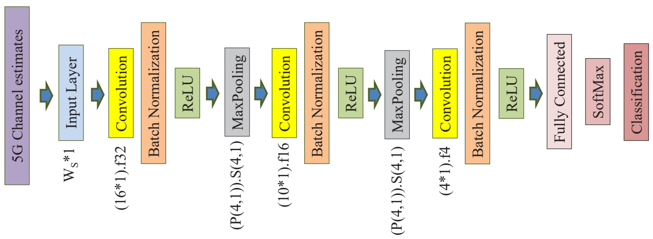


Fig. 12: Architecture of the Convolutional Neural Network used in this approach proposed in this study.

CNN architecture is based on a set of hyper-parameters, which are defined in detail in Table VI. For each hyper-parameter identified in the first column, the range of values on which the optimization was performed is provided in the second column, while in the third column the optimal value is selected. A grid approach was used to perform the optimization. Even if this may seem cumbersome, the grid approach is exhaustive and the number of hyper-parameters and ranges used in this study is anyway limited. Other parameters of the CNN architecture were set on the basis of the values of the hyper-parameters already defined. For example, the height of the filter was set to 16 in the first convolutional layer, was set to 10 in the second convolutional layer and 4 in the last convolutional layer. The width of the filter was set to 1 in all the three layers because it is a 1-D CNN. For nonlinearity, ReLU is chosen as an activation function in each layer. Two max pooling layers were used instead of the average pooling layer for better performance. The pool size was set to [4,1] and the stride to [4,1] in both layers. The number of filters in the second convolutional layer was set to half of the number of the first convolutional layer and the number of filters in the third convolutional layer was set to half of the number of filters in the second layer. The entire data set was divided in 4 portions using a 4-fold approach where the testing portion was 1/4 of the entire data set. As written before, the training data set was further divided in ten portions with one portion (i.e., 10% of the training data set) allocated for the validation task. The classification process was then repeated 10 times with different set of folds (then 4 folds per 10 is equal to 40 classification trials) and the results (Accuracy and F-score) were averaged.

A note on the potential practical implementation of the approach proposed in this paper. In this study, the channel estimator is fully implemented in the FPGA while the CNN is implemented offline in a MATLAB environment (see Subsecction IV-D). An implementation of the CNN directly in the FPGA would be quite complex, even if there have been recent proposals for in this context in research literature [31]. In any case, this aspect is out of the scope of the present study. On the other side, in a possible future deployment of the proposed approach, the CNN can be hosted on the same computer, which is hosting the FPGA hardware and which is collecting the output from the FPGA, so that the approach can be fully implemented on the same computing platform. This is to say that there are no specific constraints in the present

TABLE VI: Hyper-parameters of CNN.

Hyper-parameter	**Range of hyper-parameter values**	**Optimal Value DS1**	**Optimal Value DS2**	**Optimal Value DS3**
Solver	(RmsProp, SGD,Adam)	Adam	RmsProp	Adam
Maximum Number of epochs	[40,80,120,160]	80	60	80
Window size W_S in the first layer	[16,20,24,28, 32,36,40,44, 48,52,56,60, 64,68,72,76]	52	60	56
Number of filters in the first layer	[8,16,32,64]	32	32	64
Type of pooling	[Average Pooling, Max Pooling]	Max Pooling	Max Pooling	Max Pooling
Number of convolutional layers	[1,2,3,4,5]	3	3	3
Initial Learning Rate (IRL)	[0.0005,0.001, 0.005,0.01, 0.05,0.1]	0.001	0.005	0.005

study for future deployments on the same computer or receiver system.

C. Evaluation Metrics

The metrics used to evaluate the performance of the proposed approach are the classification accuracy, the F-score, the confusion matrices and the ROCs. The accuracy is defined as:

$$Accuracy = \frac{TP + TN}{(TP + FP + FN + TN)} \tag{4}$$

Where TP is the number of True Positives, TN is the number of True Negatives, FP is the number of False Positives and FN is the number of False Negatives. The F-score is defined as:

$$F - score = \frac{2 \times (TP)}{(2TP + FP + FN)} \tag{5}$$

Since this is multi-class problems with 12 classes (the 12 interferers), we have implemented the F-score by macro-averaging (taking all classes as equally important).

To complete the accuracy metric, confusion matrices are also provided to assess the predicted values against the true values. In the confusion matrices presented in this paper, each row of the matrix represents the instances in a true class while each column represents the instances in an predicted class.

Finally, the Receiver Operative Characteristics (ROC)s were used for binary classification to compare how well the proposed approach is able to distinguish between pair of wireless interference cases. The ROC curve is created by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings. In the ROC, the Area Under Curve (AUC) used to evaluate the performance of the binary classifier. The AUC is the area under the ROC curve. The higher is the value of the AUC and the better is the performance of the binary classifier.

D. Computing platform

The computing platform used in this study is Windows workstation with MATLAB scientific computing environment where the Deep Learning toolbox was used for the implementation of the CNN. The workstation is based on a Intel Xeon Silver 4214Y with a clock speed of 2.2 Ghz and 40 Gigabytes of RAM.

V. Results

A. Parameters optimization

In an initial phase of the study, we performed an optimization of the hyper-parameters of the CNN, which are described in Table VI. As mentioned before, a grid approach was performed with the resulting optimal values presented in Table VI. The aim of this subsection is provide an overview of the impact of the different hyper-parameters. The analysis presented here is not meant to be exhaustive, because of reasons of space, but just to show how important is the tuning of some hyper-parameters. In the following figures, only one hyper-parameter is changed while all the other hyper-parameters are set to the optimal values shown in Table VI.

The optimization of the window size parameter for the input to the CNN is shown in Figure 13 for all the three data sets (DS1,DS2 and DS3). The bar graphs show the accuracy for different values of W_s for each data set, while the line plots show the accuracy for the optimal value of W_s. It can be seen from Figure 13 that this parameter has a substantial impact on the overall accuracy, because not appropriate values of the parameter W_s decrease substantially the accuracy. It can also be seen that the optimal values of W_s are well within the range of values investigated in this study.

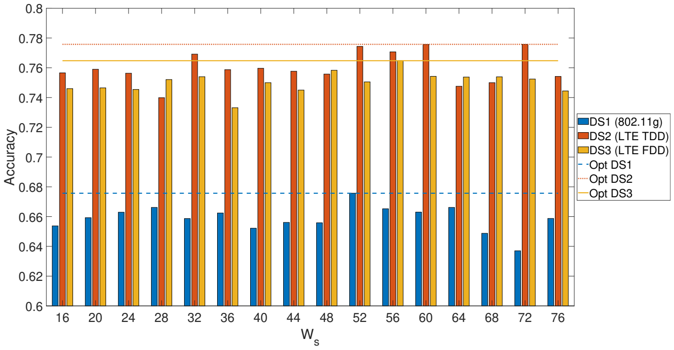


Fig. 13: Optimization of input window size W_s in CNN.

Another important hyper-parameter is the Initial Learning Rate (IRL), which controls how much the model should be changed in response to the estimated error each time the model weights are updated. There is a trade-off in choosing the learning, because a value too small may result in a long training process, whereas a value too large may result in learning a sub-optimal set of weights or in an unstable training process. The results for all the three data sets are shown in Figure 14, where the optimal values are chosen. The bar graphs show the accuracy for different values of IRL for each data set, while the line plots show the accuracy for the optimal values of IRL. It can be seen that larger values of the learning rates indeed provide a sub-optimal set of weights in a consistent way across all the three data sets. As in the previous hyper-parameter, the optimal values are identified well within the range of adopted values for this hyper-parameter.

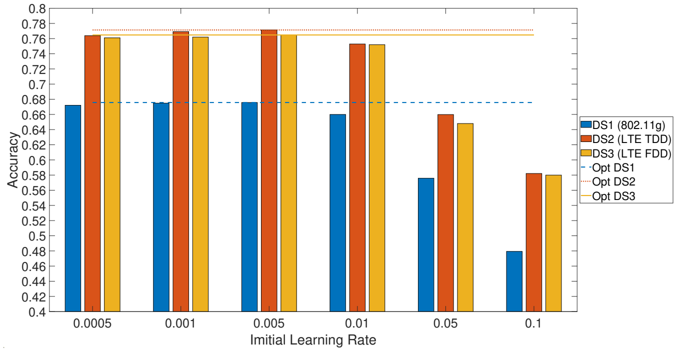


Fig. 14: Optimization of the initial learning rate (IRL) in CNN.

An evaluation of the impact of the pooling layer type (i.e., average or max pooling layer) is shown in the following Table VII. The results justify the choice of the max pooling layer type as optimal layer.

TABLE VII: Accuracy comparison of pooling layer type.

Type of layer	DS1	DS2	DS3
Max Pooling Layer	0.675	0.764	0.761
Average Pooling Layer	0.668	0.755	0.752

Even if it is not an hyper-parameter, it is interesting to evaluate how the accuracy is impacted for different sizes of the data set. A subset of the data was chosen from the initial size of 4000 samples for class (e.g., type of interference). Smaller data sets for the range of values [1000,1500,2000,2500,3000,3500] where generated and the accuracy was calculated using the CNN with the optimal values for the enture data set. To support generalization of the results, the samples were chosen randomly from the full data set and the classification process was repeated 10 times. Then, the accuracy results were averaged. Figure 15 shows the trends of the accuracy metrics using CNN with the optimal values of the hyper-parameters (with the full data set) for different sizes of the data set. It can be seen that the general trend is that the accuracy decreases with smaller data sets because the CNN needs a larger data set to generate a more accurate model, even if it would require a longer training time.

B. Approach evaluation and comparison

This subsection presents the final results of the study using the optimal values identified in the previous sub-section and shows a comparison with the application of 'shallow' ML algorithms and the alternative approach based on the constellation diagrams CNN-CONST, which is described in Section IV-A

For the ML algorithms, a grid optimization process was performed for the other ML algorithms as well on the basis of the following parameters: for the Random Forest and Decision Tree (DT) the maximal number of decision splits, for SVM the kernel and the penalty factor C and for K Nearest Neighbor (KNN) the parameter K. For the CNN-CONST approach, the hyper-parameter is the number of bins in the 2D-histogram. The results for the optimization of CNN-CONST for the data set DS1 and DS2 (the results for DS3 are similar and they are not shown for reasons of space) are shown respectively in Figures 16a and 16b, where it is shown the impact of the number of bins, but overall the classification accuracy is significantly less than with this proposed approach.

The numeric results are provided in Table VIII where it can be seen that CNN provides the highest performance accuracy and F-score in relation to the other adopted algorithms and CNN-CONST. Among the ML algorithms, the SVM (with Poylnomial kernel of third degree) obtains the best results for the data sets DS1 and DS2 (reaching almost the accuracy of the CNN) and Random Forest obtains the best results for the DS3 data set (where the performance is close to the CNN). This result is consistent with the findings from literature and in particular [10] where Random Forest and SVM managed to achieve remarkable results.

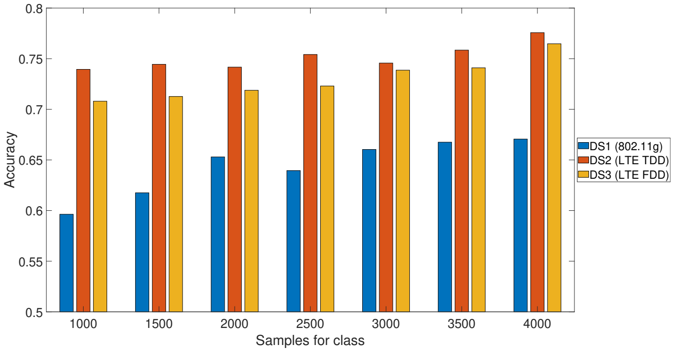
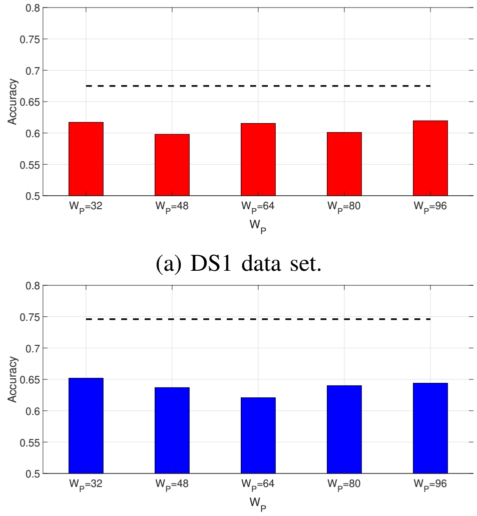


Fig. 15: Accuracy trends with CNN for different values of the number of samples for class.



(a) DS1 data set.

(b) DS2 data set.

Fig. 16: Comparison with the alternative approach based on constellation diagrams CNN-CONST acquired from the FPGA demodulator. The dotted line represents the accuracy obtained with our approach for the related data set.

These findings confirm the superior performance of CNN algorithm and the justification of its use for this classification problem.

TABLE VIII: Comparison of this approach with ML algorithms and CNN-CONST.

Approach	Accuracy	F-score
DS1 802.11g data set		
CNN (this approach)	**0.675**	**0.682**
CNN (CNN-CONST) [29], [30]	0.619	0.623
Random Forest [8], [10]	0.537	0.545
Decision Tree [8]	0.440	0.447
Support Vector Machine [7], [8]	0.667	0.669
KNN	0.568	0.571
DS2 LTE TDD data set		
CNN (this approach)	**0.764**	**0.760**
CNN (CNN-CONST) [29], [30]	0.652	0.654
Random Forest [8], [10]	0.616	0.613
Decision Tree [8]	0.535	0.534
Support Vector Machine [7], [8]	0.756	0.756
KNN	0.647	0.645
DS3 LTE FDD data set		
CNN (this approach)	**0.761**	**0.763**
CNN (CNN-CONST) [29], [30]	0.724	0.728
Random Forest [8], [10]	0.737	0.736
Decision Tree [8]	0.678	0.680
Support Vector Machine [7], [8]	0.727	0.702
KNN	0.660	0.651

C. Confusion matrices

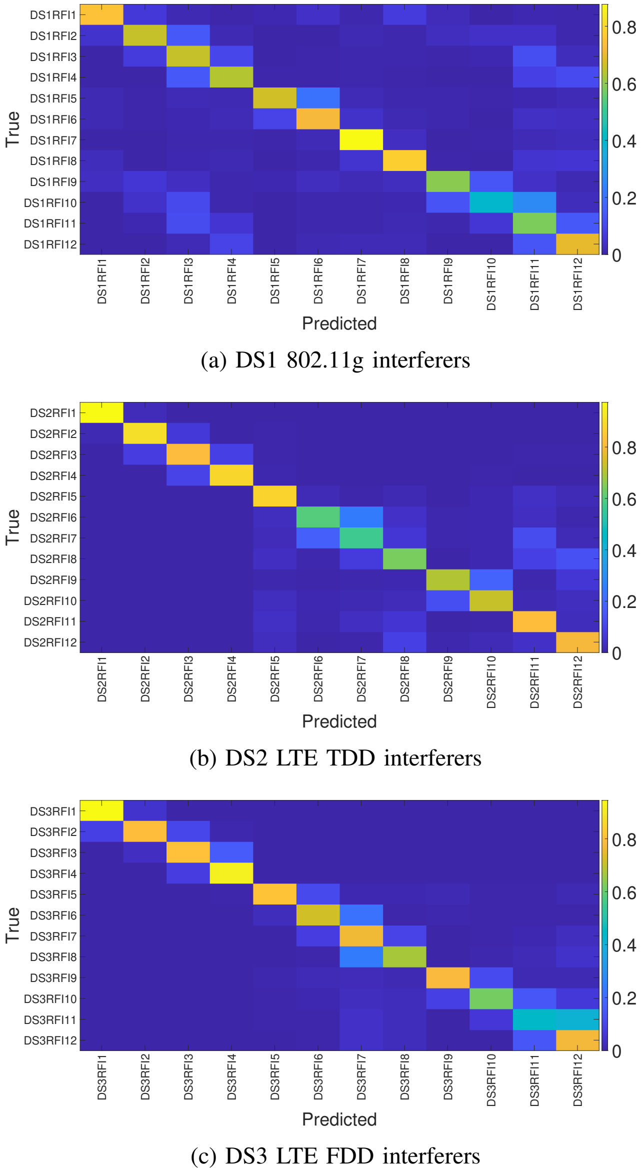
Beyond the accuracy and F-score for all the three data sets presented in Table VIII, we present in the following paragraphs and figures more detailed results using confusion matrices and ROCs.

Figures 17a, 17b and 17c provide the results of the confusion matrices respectively for the data sets DS1, DS2 and DS3. It can be seen that some interference scenarios are more difficult to distinguish than others. In particular, the algorithm has more difficulties in identifying similar scenarios (different only for the power levels and with the same spectral distance). This is to be expected because interference scenarios, which differ only for one parameters (e.g., the spectral distance or power levels) are more difficult to distinguish than interference scenarios, which are different for both parameters and where the correspondent values are the extremes of the ranges.

D. Binary classification among pairs of interfence conditions

Then, it is interesting to evaluate the performance of the proposed approach based on CNN for specific pairs of wireless interference cases rather than taking in consideration the entire data set with 12 classes. The metric used to evaluate this aspect is the ROC and the related parameter AUC which is the area under the curve.

The results from the ROCs show that the binary classification among two specific types of wireless interferer is able to obtain a very accuracy as shown by the relatively high values of the AUC (almost getting to the unit value for interferer which are more distinct on the base of their transmission power). These results are confirmed by the calculation of the accuracy and F-score on the same binary classification problems as shown in the Table IX below, where it can be seen that an accuracy greater of 90% is obtained in all the cases.



(a) DS1 802.11g interferers

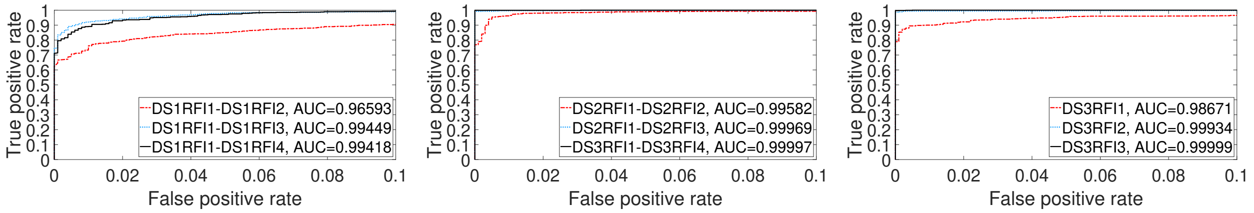
(b) DS2 LTE TDD interferers

(c) DS3 LTE FDD interferers

Fig. 17: Confusion matrices using the optimal hyperparameters values of the CNN.

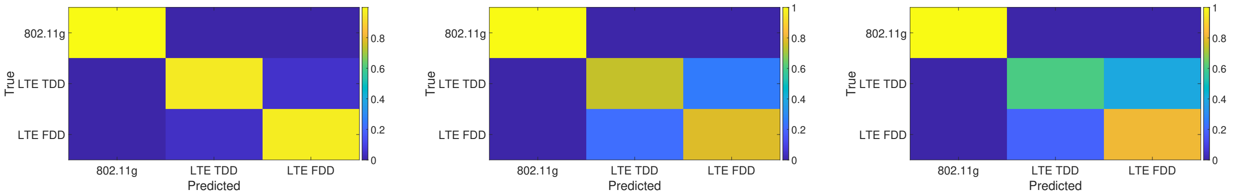
E. Performance evaluation in presence of mixed type of interference

In a practical deployment of the approach proposed in this paper, there could be also scenarios where the three types of wireless interferences signals (i.e., 802.11g, LTE TDD and LTE FDD) may be present in the radio frequency spectrum. Then, we investigated the capability of this approach to distinguish the type of wireless interference by mixing together the three data sets DS1, DS2 and DS3. For reasons of space, we show only specific values of Df and Pwr because the number of combinations can be quite large (i.e., 12^3).



(a) DS1 802.11g interferers (b) DS2 LTE TDD interferers (c) DS3 LTE FDD interferers

Fig. 18: ROCs between DS(X)RFI1 and DS(X)RFI(Y) with X=1,2,3 and Y=2,3,4 (e.g., DS1RFI1 and DS1RFI2). For clarity the values of X-axis (False Positive Rate) are within the range 0...0.1 (instead of the usual range 0...1).



(a) Set composed of DS1RF2,DS2RF2,DS3RF2. Df=26MHz and Pwr=-35 dB.
(b) Set composed of DS1RF6,DS2RF6,DS3RF6. Df=28MHz and Pwr=-35 dB.
(c) Set composed of DS1RF10,DS2RF10,DS3RF10. Df=30MHz and Pwr=-35 dB

Fig. 19: Confusion matrices between three different classes of interferers (DS1RFX,DS2RFX,DS3RFX) for specific values of Df and Pwr (X=2,6,10).

TABLE IX: Accuracy for specific combination of wireless interferences scenarios.

Combination	**Accuracy**	**F-score**
DS1 802.11g data set		
DS1RFI1 vs DS1RFI2	0.9030	0.9020
DS1RFI1 vs DS1RFI3	0.9635	0.9633
DS1RFI1 vs DS1RFI4	0.9575	0.9569
DS2 LTE TDD data set		
DS2RFI1 vs DS2RFI2	0.9790	0.9788
DS2RFI1 vs DS2RFI3	0.9950	0.9950
DS2RFI1 vs DS2RFI4	0.9985	0.9985
DS3 LTE FDD data set		
DS3RFI1 vs DS3RFI2	0.9520	0.9523
DS3RFI1 vs DS3RFI3	0.9925	0.9924
DS3RFI1 vs DS3RFI4	0.9985	0.9985

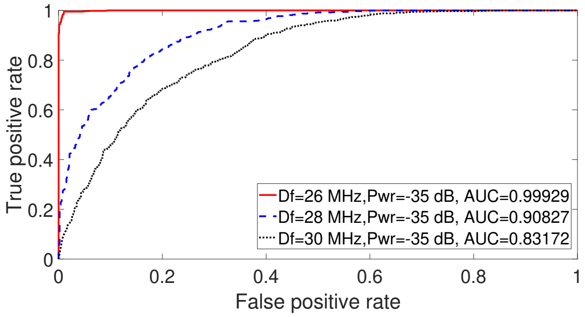


Fig. 20: ROCs between the LTE TDD and LTE FDD interferers for different values of Df and Pwr=-35 dB.

We use the confusion matrices and the ROCs to evaluate the performance in this classification problem. Figures 19a, 19b and 19c show respectively the confusion matrix for Df= 28 MHz, Pwr= -35 dB. We note that the approach is able to distinguish with great accuracy the 802.11g interferer from the LTE TDD and LTE FDD. This is expected due to the significant difference in the signal structure. Instead, it is more challenging to distinguish the LTE TDD interfer from the LTE FDD interferer. In addition, the accuracy obtained by the proposed approach is inversely proportional to the value of Df: the more is the spectral distance of the interfer from the victim system and the less is the classification accuracy. This is also expected because for larger values of Df, the interference impact on the channel emulator is less significant, which reduces the discriminating information that the CNN can exploit to distinguish the different classes. Similar results were obtained for other values of Pwr but they are not shown for reasons of space.

On the basis of the previous considerations based on the confusion matrices that the proposed approach has difficulties in distinguishing the LTE TDD and LTE FDD interfers, we perform a binary classification only among samples from DS2 and DS3 and we use the ROCs to evaluate the performance. The results are shown in Figure 20 for Pwr=-35 dB and different values of Df, where it can be seen that the AUC decreases significantly for higher values of Df, which confirms the previous statements derived from the analysis of the confusion matrices. Similar results were obtained for other values of Pwr but they are not shown for reasons of space.

F. Time complexity and computational complexity aspects

We address in this final subsection a description of the time complexity and computational complexity aspects related to the implementation of the proposed approach for the two main components: the FPGA component used to implement the LTE-plus UE receiver and the embedded channel estimator and the CNN component. Regarding the FPGA component of the proposed approach implementing the channel estimator, we have used a Xilinx Kintex-7 XC7K410T FPGA embedded in the NI-2944R USRPs used both for the transmission and receiving part of the LTE-plus victim system. The typical parameters used to describe the Xilinx FPGA resource usage are [32]: Look-Up Tables (LUT)s, which are used to implement combinational logic in an FPGA, Registers, which are used to store values in a FPGA, Digital Signal Processing (DSP) slices used to perform high-speed digital signal processing operations such as multiplication and addition and Block RAM, which is used to store data in the FPGA. We have used the Xilinx Vivado tool to generate resource utilization reports and we report that the percentage of usage of the different components was respectively: LUTs (28%), DSP (31%), Block RAM (31%), Registers (18%). Regarding the CNN implementation, we report the training time and the testing time for the execution of 1 fold for the DS1 data set (averaged on 40 repetitions) on the basis of the computing platform specifications described in Sub-section IV-D: Training time: 160 seconds, Testing time (prediction): 0.57 seconds.

VI. Conclusions and Future Developments

This paper has described a novel approach to identify wireless adjacent interferences conditions in a custom FPGA LTE communication system (a custom LTE-plus FPGA front end was implemented for this study with a 40 MHz communication bandwidth) using the channel estimator function, where the CRS channel estimates are given in input to a CNN. This approach has the main advantage to exploit an already existing function in the LTE receiver systems, which is used for a different purpose (i.e., fading channel estimation), without the need to implement a specific module to process the raw IQ data as commonly done in research literature. The approach was evaluated using three data sets (based on 802.11g, LTE TDD and LTE FDD standards) created in the JRC Radio Frequency laboratory where 12 different wireless interferences conditions were generated for different spectral distance and power of the interferer from the LTE victim service. These data sets are particularly challenging for classification because some of the wireless interference conditions are quite similar. The results show that the approach is able to effectively distinguish the different wireless interference with a maximum accuracy of 0.764 when all the 12 wireless interferences in the data set are considered. The identification accuracy between pairs of wireless interference cases can be even higher, reaching a maximum value of 0.998 for specific pairs of wireless interferences. A comparison with approaches from literature based on machine learning and with constellation diagrams was also performed, showing the superior performance of the approach proposed in this paper.

Future developments may investigate different types of wireless interferences from the ones evaluated in this study. In addition, unsupervised learning will be attempted to evaluate how well the proposed approach is able to identify unknown cases of interference. We would also like to highlight that the same approach proposed in this paper and based on the available LTE-plus (with 40 MHz bandwidth) FPGA implementation of the CRS can be extended in the future on 5G NR systems with CSI-RS in the future. Another important future development is related to the extension of the approach proposed in this paper to Multiple Input Multiple Output (MIMO). This study was focused specifically on SISO for three different kinds of interferers as the analysis was already quite complex for the SISO case. On the other side, it is also interesting to take in consideration more complex MIMO scenarios. This is out of scope of the present study as it would require a new FPGA implementation and a completely new sets of results, but the authors aims to explore the extension to MIMO in future developments.

Acknowledgments

We acknowledge the contributions and technical support in setting up the test bed by the JRC colleague James Bishop.

References

[1] F. K. Jondral, “Cognitive radio: A communications engineering view,” *IEEE Wireless Communications*, vol. 14, no. 4, pp. 28–33, 2007.

[2] P. Cardieri, “Modeling interference in wireless ad hoc networks,” *IEEE Communications Surveys & Tutorials*, vol. 12, no. 4, pp. 551–572, 2010.

[3] Q. Mao, F. Hu, and Q. Hao, “Deep learning for intelligent wireless networks: A comprehensive survey,” *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2595–2621, 2018.

[4] M. Schmidt, D. Block, and U. Meier, “Wireless interference identification with convolutional neural networks,” in *2017 IEEE 15th International Conference on Industrial Informatics (INDIN)*. IEEE, 2017, pp. 180–185.

[5] S. Zinno, G. Di Stasi, S. Avallone, and G. Ventre, “On a fair coexistence of LTE and wi-fi in the unlicensed spectrum: A survey,” *computer communications*, vol. 115, pp. 35–50, 2018.

[6] A. M. Voicu, L. Simić, and M. Petrova, “Inter-technology coexistence in a spectrum commons: A case study of wi-fi and LTE in the 5-ghz unlicensed band,” *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 11, pp. 3062–3077, 2016.

[7] S. Grimaldi, A. Mahmood, and M. Gidlund, “An SVM-based method for classification of external interference in industrial wireless sensor and actuator networks,” *Journal of Sensor and Actuator Networks*, vol. 6, no. 2, p. 9, 2017.

[8] ——, “Real-time interference identification via supervised learning: Embedding coexistence awareness in iot devices,” *IEEE Access*, vol. 7, pp. 835–850, 2019.

[9] F. Barać, M. Gidlund, and T. Zhang, “Ubiquitous, yet deceptive: Hardware-based channel metrics on interfered wsn links,” *IEEE Transactions on Vehicular Technology*, vol. 64, no. 5, pp. 1766–1778, 2014.

[10] R. Candell, K. Montgomery, M. Kashef, Y. Liu, and S. Foufou, “Wireless interference estimation using machine learning in a robotic force-seeking scenario,” in *2019 IEEE 28th International Symposium on Industrial Electronics (ISIE)*. IEEE, 2019, pp. 1334–1341.

[11] T. Gruber, S. Cammerer, J. Hoydis, and S. ten Brink, “On deep learning-based channel decoding,” in *2017 51st Annual Conference on Information Sciences and Systems (CISS)*. IEEE, 2017, pp. 1–6.

[12] S. Peng, H. Jiang, H. Wang, H. Alwageed, and Y.-D. Yao, “Modulation classification using convolutional neural network based deep learning model,” in *2017 26th Wireless and Optical Communication Conference (WOCC)*. IEEE, 2017, pp. 1–5.

[13] K. Merchant, S. Revay, G. Stantchev, and B. Nousain, “Deep learning for RF device fingerprinting in cognitive communication networks,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 160–167, 2018.

[14] G. Baldini and C. Gentile, “Transient-based internet of things emitter identification using convolutional neural networks and optimized general linear chirplet transform,” *IEEE Communications Letters*, vol. 24, no. 7, pp. 1482–1486, 2020.

[15] S. Grunau, D. Block, and U. Meier, "Multi-label wireless interference classification with convolutional neural networks," in *2018 IEEE 16th International Conference on Industrial Informatics (INDIN)*. IEEE, 2018, pp. 187–192.

[16] V. Maglogiannis, A. Shahid, D. Naudts, E. De Poorter, and I. Moerman, "Enhancing the coexistence of LTE and Wi-Fi in unlicensed spectrum through convolutional neural networks," *IEEE Access*, vol. 7, pp. 28 464–28 477, 2019.

[17] P. Wang, Y. Cheng, Q. Peng, B. Dong, and S. Li, "Low-bitwidth convolutional neural networks for wireless interference identification," *IEEE Transactions on Cognitive Communications and Networking*, 2022.

[18] P. Wang, Y. Cheng, B. Dong, and Q. Peng, "Bring globality into convolutional neural networks for wireless interference classification," *IEEE Wireless Communications Letters*, vol. 11, no. 3, pp. 538–542, 2021.

[19] P. Wang, Y. Cheng, G. Shang, J. Wang, and S. Li, "Time-frequency component-aware convolutional neural network for wireless interference classification," *IEEE Wireless Communications Letters*, 2022.

[20] G. Liu, Z. Xi, and R. Liu, "A novel wireless interference identification and scheduling method based on convolutional neural network," in *2022 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2022, pp. 1–6.

[21] X. Zhang, T. Seyfi, S. Ju, S. Ramjee, A. El Gamal, and Y. C. Eldar, "Deep learning for interference identification: Band, training SNR, and sample selection," in *2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 2019, pp. 1–5.

[22] J. Huang, M. L. Huang, P. H. Tan, Z. Chen, and S. Sun, "Semi-supervised deep learning based wireless interference identification for IIoT networks," in *2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*. IEEE, 2020, pp. 1–5.

[23] H. Asplund, D. Astely, P. von Butovitsch, T. Chapman, M. Frenne, F. Ghasemzadeh, M. Hagström, B. Hogan, G. Jongren, J. Karlsson *et al.*, *Advanced Antenna Systems for 5G Network Deployments: Bridging the Gap Between Theory and Practice*. Academic Press, 2020.

[24] C. Hoymann, D. Larsson, H. Koorapaty, and J.-F. Cheng, "A lean carrier for LTE," *IEEE Communications Magazine*, vol. 51, no. 2, pp. 74–80, 2013.

[25] H. Koorapaty, J.-F. Cheng, J.-C. Guey, and S. Grant, "Reference signals for improved energy efficiency in LTE," in *2013 IEEE 78th Vehicular Technology Conference (VTC Fall)*. IEEE, 2013, pp. 1–5.

[26] F. Sun, L. Lu, and T. B. Sørensen, "Designs of precoding for LTE TDD using cell specific reference signals," in *2010 IEEE Globecom Workshops*. IEEE, 2010, pp. 871–875.

[27] 3GPP, "Ts 38.211: technical specification group radio access network; nr; physical channels and modulation," 2019.

[28] R. M. Al-Makhlasawy, A. A. Hefnawy, M. M. Abd Elnaby, and F. E. Abd El-Samie, "Modulation classification in the presence of adjacent channel interference using convolutional neural networks," *International Journal of Communication Systems*, vol. 33, no. 13, p. e4295, 2020.

[29] V.-S. Doan, T. Huynh-The, C.-H. Hua, Q.-V. Pham, and D.-S. Kim, "Learning constellation map with deep CNN for accurate modulation recognition," in *GLOBECOM 2020-2020 IEEE Global Communications Conference*. IEEE, 2020, pp. 1–6.

[30] G. Oligeri, S. Sciancalepore, S. Raponi, and R. Di Pietro, "PAST-AI: Physical-layer authentication of satellite transmitters via deep learning," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 274–289, 2022.

[31] D. T. Nguyen, T. N. Nguyen, H. Kim, and H.-J. Lee, "A high-throughput and power-efficient FPGA implementation of YOLO CNN for object detection," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 8, pp. 1861–1873, 2019.

[32] xilinx, "Xilinx 7 series product selection guide," *https://www.xilinx.com/content/dam/xilinx/support/documents/selection-guides/7-series-product-selection-guide.pdf*, 2021.

VII. Biography Section

Gianmarco Baldini Gianmarco Baldini is currently a Senior Researcher/Project Manager in the Joint Research Centre of the European Commission. He received the Laurea degree in Electronic Engineering from the University of Rome,Italy in 1993 and his PhD in computer science at the University of Insubria,Italy in 2019. In the JRC, he is working to support European regulatory frameworks in the areas of RF spectrum, AI, and cybersecurity to implement pre-normative research to anticipate future European policies.

Fausto Bonavitacola Fausto Bonavitacola is an expert in data acquisition, FPGA programming, control and simulation devices, hardware-in-the-loop testing, sensors and measure instrumentation, 5G NR, GNSS and in general RF radio/satellite communications. He holds a degree in electronics with a specialization in devices and circuits. He has been involved for the last 25 years as a radio frequency expert and IT expert in many projects and/or research activities at the Joint Research Center of the European Commission.

Jean-Marc Chareau Jean-Marc Chareau is a electronics engineer who has been working at the Joint Research Centre, Ispra, Italy since 2006 on various projects related to radio frequency measurement, electronic passport, RF co-existence testing and efficient use of radio spectrum and spectrum measurements. Previously he worked at CEA, France on fusion plasma diagnostics.