Deep Learning-aided Robust Integrated Sensing and Communications with OTFS and Superimposed Training

Méndez-Monsanto Suárez, Lianet; Chen Hu, Kun; Fernández-Getino García, Maria Julia; Armada, Ana Garcia

Published in:
2023 IEEE International Mediterranean Conference on Communications and Networking, MeditCom 2023

DOI (link to publication from Publisher):
10.1109/MeditCom58224.2023.10266634

Publication date:
2023

Document Version
Accepted author manuscript, peer reviewed version

Deep Learning-aided Robust Integrated Sensing and Communications with OTFS and Superimposed Training

Lianet Méndez-Monsanto Suárez[1], Kun Chen-Hu[1,2], M. Julia Fernández-Getino García[1], Ana García Armada[1]
[1]Department of Signal Theory and Communications, Universidad Carlos III de Madrid, Spain
[2]Department of Electronic Systems, Aalborg University, Denmark
E-mails: {lianet, kchen, mjulia, agarcia}@tsc.uc3m.es

***Abstract*—One of the most awaited services promised by the sixth generation (6G) of mobile communication technologies is integrated sensing and communications (ISAC), which will offer a wealth of new benefits and applications. One of the most widely used waveforms for this purpose is orthogonal time-frequency space (OTFS), which together with superimposed training (ST) allows communication and target localization in the delay-Doppler (DD) domain with zero overhead. The conventional technique uses a threshold for the possible detection of a target. However, the optimal choice of this threshold requires statistical analysis of the received signal and prior knowledge of the channel, which is initially unknown. To address this problem, this paper proposes a target detection technique based on deep learning (DL), which is able to assist in the detection of targets without the need of a prior computation of any statistical parameter. The proposed deep neural network (DNN) outperforms the conventional threshold decision method and approaches the ideal performance.**

***Index Terms*—OTFS, ISAC, deep learning, superimposed-training, channel estimation**

I. Introduction

The future Sixth Generation (6G) communications network [1] will open up a wide range of possibilities by developing innovative services in a variety of demanding environments. In one of the most anticipated services, intelligent target detection and broadband communications are synergistically combined in what is called integrated sensing and communications (ISAC). This integration will be essential in next-generation high-mobility scenarios, such as vehicles, drones, etc., as it will not only be able to transmit and receive data signals, but also efficiently provide real-time localization information of the surroundings using the available communication resources [2].

One of the most interesting waveforms for high mobility scenarios is orthogonal time-frequency space (OTFS) [3], [4]. Unlike orthogonal frequency division multiplexing (OFDM), it allows to circumvent the time-varying effect caused by the Doppler spread, as the data symbols are represented in the delay-Doppler (DD) domain. For this reason, OTFS is frequently considered for ISAC in the literature [5], [6]. However, in practical implementations, OTFS requires a large pilot overhead, since each pilot must have a set of guard symbols (zeros) surrounding it to avoid self-interference caused by the data, what leads to a significant deterioration of the data rate [7].

To address this pilot overhead problem, the use of superimposed-training (ST) with OTFS, i.e. ST-OTFS, has been recently proposed in the literature [8], [9]. ST consists in superimposing the pilot signal over the data signal, and hence, no resources are dedicated exclusively to the pilots, producing a zero overhead. However, a self-interference induced by the data symbols in the superposition process must be considered at the receiver. The works [8] and [9] introduce innovative methodologies for channel estimation in ST-OTFS systems. These methodologies incorporate self-interference as an additional interference and noise, which is partially eliminated through iterative techniques that demand significant computational resources. Consequently, the accuracy of the channel and sensing parameter estimates is confined to scenarios with high signal-to-noise ratios (SNR).

As a possible solution to this, in our earlier work [10] we propose an averaging technique in the DD domain, which allows an effective filtering of the interference with an affordable computational cost. First, the normalized delay and Doppler shifts are estimated by performing a block averaging of the received DD signal followed by a bank of correlators. Then, the channel complex gains are accurately obtained by using a two-step computationally efficient method. Thanks to the filtering, this technique is more robust to noise and interference, which results in better estimates.

In the latter approach, following the bank of correlators, targets are detected utilizing a decision threshold, which plays a pivotal role in determining the system's overall performance. Selecting an optimal threshold necessitates a statistical analysis of the problem and is contingent upon prior knowledge about the channel, which remains unknown at the beginning. Consequently, in this paper, we enhance the findings presented in [10] by incorporating a deep neural network (DNN), which detects the targets and estimates their respective delay and Doppler shifts, eliminating the requirement for prior threshold decisions.

Numerous studies in the field have shown the effectiveness of incorporating deep learning (DL) into OTFS-based architectures [11]–[14] for various applications. These investigations

have consistently revealed that employing DL-aided OTFS schemes often surpasses the performance of traditional methods across multiple scenarios. In this paper, through exposure to diverse scenarios and multiple channels during training, the DNN acquires robustness against fluctuations in channel parameters, thereby reducing the necessity for repeated training. The results indicate that the DNN successfully detects targets without depending on prior assumptions, performing closely to the ideal threshold case. We present the outcomes in terms of detection probability and false alarm rates.

The remainder of the paper is organized as follows. Section II presents the OTFS system model and Section III introduces the channel estimation with ST-OTFS for ISAC proposed in our earlier work [10]. Section IV describes the proposed DNN technique for target detection and justifies its use as an alternative to traditional detection methods in the radar literature. In Section V, we present some results to evaluate the performance of the DNN and compare it to the use of the conventional threshold. Finally, we conclude the work in Section VI.

Notation: in this work, matrices, vectors and scalar quantities are represented by boldface uppercase, boldface lowercase, and normal letters, respectively. $[\mathbf{A}]_{m,n}$ denotes the element in the m-th row and n-th column of $\mathbf{A}$. $[\mathbf{a}]_n$ represents the n-th element of the vector $\mathbf{a}$. $[\mathbf{a}]^{(n)}$ denotes the resulting vector from right-shifting n samples of vector $\mathbf{a}$. $\mathbf{I}_M$ is the identity matrix of size $(M \times M)$. $\mathbf{1}_{(M\times N)}$ denotes a matrix of ones of size $(M \times N)$. $\mathbf{A} = \text{diag}(\mathbf{a})$ is a diagonal matrix whose diagonal elements are formed by the elements of vector $\mathbf{a}$. The superscripts $(\cdot)^T$ and $(\cdot)^H$ denote transpose and hermitian operations, respectively. $\otimes$ is the Kronecker product. $\odot$ denotes the Hadamard product. $\mathbb{E}\{\cdot\}$ represents the expected value. $\mathcal{CN}(0, \sigma^2)$ represents the circularly-symmetric and zero-mean complex normal distribution with variance σ^2. $|\cdot|$ represents the absolute value of a complex number. $\mathbb{C}^K$ and $\mathbb{Z}^K$ are K-dimensional complex and integer spaces, respectively, and $\mathbb{C}^{K\times K}$ is the $K\times K$-dimensional complex space. Finally, circ$[\cdot]$ denotes the circulant matrix.

II. System Model

The OTFS system model [4] maps the complex data and pilot symbols onto a resource grid in the DD domain. These symbols are then converted into delay-time domain samples using the inverse discrete Zak transform (IDZT). In particular, zero-padding OTFS (ZP-OTFS) is chosen since its appends a ZP at the end of each multi-carrier symbol to absorb the intersymbol interference (ISI) of the channel. The signal is affected by a frequency-selective and time-varying channel, with the following expression in the DD domain

$$h(\tau, \nu) = \sum_{i=1}^{L_p} g_i e^{-j2\pi\nu_i\tau_i}\delta(\tau - \tau_i)\delta(\nu - \nu_i) \tag{1}$$

where L_p is the number of propagation paths and g_i denotes the channel complex gains. τ_i and ν_i are the delay and Doppler shifts, respectively, with the following expressions

$$\tau_i = \frac{l_i}{M\Delta f} \leq \tau_{max} = \frac{l_{max}}{M\Delta f}, \tag{2}$$

$$\nu_i = \frac{k_i}{NT} \quad |\nu_i| \leq \nu_{max} \tag{3}$$

where M and N corresponds to the number of delay and Doppler bins, respectively. $l_i, k_i \in \mathbb{Z}$ are the normalized delay and Doppler shifts, respectively, which are assumed to be integers. The channel is assumed to be underspread [7]–[9], so that $\tau_{max}\nu_{max} \ll 1$ and $T\Delta f = 1$. The normalized delay and Doppler shifts are constrained to $l_{max} < M$ and $-N/2 \leq k_i < N/2$. The ZP is then selected to have length $L_{ZP} \geq l_{max}$, to absorb the ISI.

At the end, the received signal is again converted to the DD domain by using the discrete Zak transform (DZT) [4], [15]. The equivalent channel model in the DD domain can be given as [7]–[9]

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{z}, \tag{4}$$

where $\mathbf{y} \in \mathbb{C}^{MN\times 1}$ is the received signal in the DD domain, $\mathbf{x} \in \mathbb{C}^{MN\times 1}$ is the transmitted data vector, composed of quadrature amplitude modulation (QAM) symbols, with unit energy ($\mathbb{E}\left\{\left|[\mathbf{x}]^i\right|^2\right\} = 1$, $1 \leq i \leq MN$). $\mathbf{z} \in \mathbb{C}^{MN\times 1}$ denotes the additive white Gaussian noise (AWGN), with $\mathcal{CN}\left(0, \sigma_z^2\right)$, and $\mathbf{H} \in \mathbb{C}^{MN\times MN}$ is the equivalent channel matrix in the DD domain, which is modeled as [8], [16], [17]

$$\mathbf{H} = (\mathbf{F}_N \otimes \mathbf{I}_M)\left(\sum_{i=1}^{L_p} g_i e^{-j2\pi\nu_i\tau_i}\mathbf{\Pi}^{l_i}\mathbf{\Delta}^{k_i}\right)(\mathbf{F}_N^H \otimes \mathbf{I}_M). \tag{5}$$

where $\mathbf{F}_N \in \mathbb{C}^{N\times N}$ denotes the normalized Fourier transform (DFT) matrix, $\mathbf{\Pi} \in \mathbb{C}^{MN\times MN}$ is the forward cyclic-shift (permutation) matrix and $\mathbf{\Delta} = \text{diag}\left(\left[1, e^{j2\pi\frac{1}{MN}}, ..., e^{j2\pi\frac{MN-1}{MN}}\right]\right)$.

III. Channel Estimation for ISAC with ST

In this section, we briefly describe the OTFS channel estimation and sensing scheme based on ST proposed in [10], where the pilot symbols are superimposed on the data ones. We start by defining the pilot design that will allow us to later perform the averaging method in the DD domain. The pilot design covers the entire lattice, and its pattern is as follows. We define a pilot sequence $\mathbf{p} \in \mathbb{C}^{M\times 1}$ as

$$\mathbf{p} = [p_0, ..., p_{M-1}]^T = [e^{j2\pi\phi_{p_0}}, ..., e^{j2\pi\phi_{p_{M-1}}}]^T \tag{6}$$

where ϕ_{p_b} denotes the phase of the pilot and $b = 0, 1, ..., M-1$. This pilot sequence is replicated for each multi-carrier symbol as

$$\mathbf{x}_p = \mathbf{1}_{(N\times 1)} \otimes \mathbf{p} \in \mathbb{C}^{MN\times 1} \tag{7}$$

where $\mathbf{x}_p$ is the pilot signal that will be superimposed on the data as

$$\mathbf{x} = \sqrt{\beta}\mathbf{x}_d + \sqrt{1-\beta}\mathbf{x}_p, \tag{8}$$

where β is the power assigned to the data and $\mathbf{x}_d$ is the data signal. The received signal (4) can be expressed as

$$\mathbf{y} = \mathbf{H}\mathbf{x}_d + \mathbf{H}\mathbf{x}_p + \mathbf{z} \in \mathbb{C}^{MN\times 1}. \tag{9}$$

The channel estimation and sensing scheme [10] is based on four steps: 1) a pre-processing of the received signal to average it in the DD domain, 2) the estimation of the normalized delay and Doppler shifts, 3) the computation of the channel complex gains, in this order. Finally, 4) the symbols are equalized to recover the transmitted communications signal and the sensing parameters are used to compute the range and velocity of the targets. The main idea of the scheme is to perform an averaging technique on the signal. Making use of the fact that the data and noise samples are zero-mean random variables [8] ($\mathbb{E}\{\mathbf{x}_d\} = \mathbb{E}\{\mathbf{z}\} = 0$), the averaging reduces both data-induced self-interference and noise effects.

1) *Pre-processing*: The averaging in the DD domain of the received signal is done by multiplying the received signal $\mathbf{y}$ by an averaging matrix $\mathbf{W}$ of size $M \times MN$,

$$\overline{\mathbf{y}} = \frac{1}{N}\mathbf{W}\mathbf{y} = \frac{1}{N}\left(\mathbf{1}_{(1\times M)} \otimes \mathbf{I}_N\right)\mathbf{y}, \tag{10}$$

where $\overline{\mathbf{y}} \in \mathbb{C}^{M\times 1}$ is the averaged received vector $\mathbf{y}$, whose expression is

$$[\overline{\mathbf{y}}]_m = \sum_{i=1}^{m}\sum_{n=1}^{L_p^{(l_i)}} g_n p_{m-i+1} e^{j2\pi\frac{k_n(m-i-l_n)}{MN}} + x'_d + z', \tag{11}$$

with $L_p^{(l_i)}$ being the number of propagation paths whose delay is l_i, $i = 1, ..., m$, where m denotes the index position of $\overline{\mathbf{y}}$. The terms x'_d and z' denote the residual self-interference from the data and the residual noise, respectively, with variances $\sigma^2_{x'} = \frac{\beta}{N}$, $\sigma^2_{z'} = \frac{\sigma_z^2}{N}$. Note that the averaging reduces both variances by a factor of N.

2) *Estimation of the delay and Doppler shifts*: Given that the possible options for the values of the normalized delay and Doppler shifts are bounded by $l_i < M$ and $-N/2 \le k_i < N/2$, we use a bank of correlators to infer their values from the received signal. We define the reference vector of Doppler exponentials $\boldsymbol{\kappa}_{k_i}$ as

$$\boldsymbol{\kappa}_{k_i} = e^{j2\pi\frac{k_i q}{MN}} \in \mathbb{C}^{MN\times 1} \text{ with } q = 0, ..., MN-1, \tag{12}$$

$$\frac{1}{MN}\boldsymbol{\kappa}_{k_i}\boldsymbol{\kappa}_{k_j}^H = \begin{cases} 1, \text{ if } i = j, \\ 0, \text{ otherwise.} \end{cases} \tag{13}$$

And we define $\boldsymbol{\kappa}_{k,p}$ as the reference Doppler vector $\boldsymbol{\kappa}_{k_i}$ multiplied by the pilots

$$\boldsymbol{\kappa}_{k,p} = \boldsymbol{\kappa}_{k_i}(M) \odot \mathbf{p} = \left[p_0 e^{j2\pi\frac{0}{MN}}, ..., p_{M-1} e^{j2\pi\frac{k_i(M-1)}{MN}}\right]^T, \tag{14}$$

where the term (M) in $\boldsymbol{\kappa}_k(M)$ refers to the first M samples of the signal. We correlate shifted copies of the vector $\boldsymbol{\kappa}_{k,p}$ with the received averaged signal $\overline{\mathbf{y}}$, so that correlation peaks will be obtained whenever the Doppler is present in the received signal. The position of the correlation peak gives the information of the delay shift, since the delay of the tap is directly related to the position of the tap in the averaged received vector.

To store the results of the correlations, we define a matrix $\mathbf{C}^{(\text{abs})} \in \mathbb{R}^{N\times M}$, with each of its positions in row index k and column index l as

$$[\mathbf{C}^{(\text{abs})}]_{k,l} = \frac{1}{M}\left|\,\overline{\mathbf{y}}[\boldsymbol{\kappa}_{k,p}^H]^{(l)}\right| \tag{15}$$

with $l = 0, ..., M-1$ and $-N/2 \le k < N/2$. The estimates of the normalized delay and Doppler shifts will be given by

$$\left(\tilde{k}_i, \tilde{l}_i\right) = \arg\max_{k,l}\left[\mathbf{C}^{(\text{abs})}\right]_{k,l}, \tag{16}$$

where a target will be detected if the correlation peak is higher than a certain threshold γ

$$[\mathbf{C}^{(\text{abs})}]_{k,l} \underset{\tilde{l}_i\neq l, \tilde{k}_i\neq k}{\overset{\tilde{l}_i=l, \tilde{k}_i=k}{\gtrless}} \gamma. \tag{17}$$

3) *Computation of the channel complex gains*: once the estimated delay and Doppler vectors are known $\tilde{l}_i$ and $\tilde{k}_i$, the averaged received vector is used to compute the estimated channel complex gains $\tilde{g}_i$. The method accurately yields $\tilde{g}_i$ in a two-step procedure with a low level of computational complexity.

4) *Data detection and sensing*: The estimated channel matrix is built from the estimated $\tilde{l}_i$, $\tilde{k}_i$ and $\tilde{g}_i$ and a minimum mean squared error (MMSE) equalization is used to compute the estimated data symbols as

$$\hat{\mathbf{x}}_{\text{MMSE}} = \left(\tilde{\mathbf{H}}^{\text{H}}\tilde{\mathbf{H}} + \sigma_z^2\mathbf{I}_{MN}\right)^{-1}\tilde{\mathbf{H}}^{\text{H}}\mathbf{y}. \tag{18}$$

We consider passive sensing, in which the receiver infers the sensing parameters of the targets from a received communications signal. Some examples of passive sensing scenarios can be found in [9] and [18], where the velocity and range of the targets are derived by relating a reflected non-line of sight (NLoS) tap with a reference LoS tap. The delay and Doppler shifts are proportional to the range and relative velocity of the targets, as $r \propto \tau c$ and $v \propto \nu c/f_c$ where c is the speed of light and f_c is the carrier frequency.

IV. Proposed DNN for Target Detection

The choice of the threshold γ in (17) determines the overall performance of the system. This threshold determines if a target is present or not, based on the values of the correlation matrix $\mathbf{C}^{(\text{abs})}$. A poor threshold decision could result in suboptimal operations. For this reason, a scheme based on DNNs is proposed that is capable of detecting taps in the matrix $\mathbf{C}^{(\text{abs})}$ without the need for defining a threshold or any prior statistical analysis. The architecture is illustrated in Fig. 1. The input of the DNN tap detector is the correlation matrix $\mathbf{C}^{(\text{abs})}$, and the output is the estimate of the normalized delay and Doppler shifts $\left(\tilde{k}_i, \tilde{l}_i\right)$. Note that the input matrix does not necessarily have to be $N \times M$, it can be reduced to $N \times L_{ZP}$, where $L_{ZP} \ge l_{max}$, since the maximum delay spread appearing in the matrix will be given by this value.

The DNNs used are based on convolutional neural networks (CNNs), which are widely used in the literature for applying DL to the physical layer, since the signals can be easily represented as if they were pixel-based image processing. Each

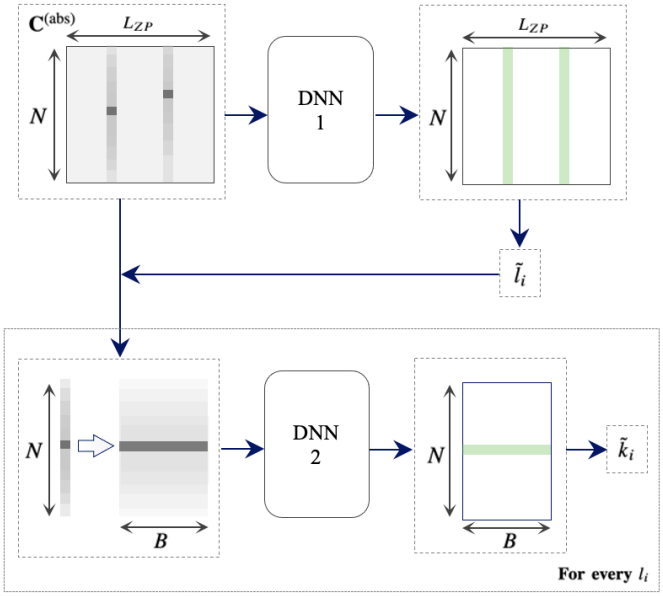

Fig. 1. DNN architecture to estimate the normalized delay and Doppler shifts.

TABLE I
DNN ARCHITECTURE

Layer	Filter	Stride	Shape
Input layer: $\mathbf{C}^{(abs)}$, $R = L_{ZP}$ (DNN1) $\left[\mathbf{C}^{(abs)}\right]_k$, $R = B$ (DNN2)			$(N \times R \times 1)$
2-D Convolutional	$(3, 3)$	$(1, 1)$	$(N \times R \times 32)$
ReLU			$(N \times R \times 32)$
2-D Max Pooling		$(2, 2)$	$(N/2 \times R/2 \times 32)$
2-D Convolutional	$(3, 3)$	$(1, 1)$	$(N/2 \times R/2 \times 32)$
ReLU			$(N/2 \times R/2 \times 32)$
2-D Max Pooling		$(2, 2)$	$(N/4 \times R/4 \times 32)$
2-D Transposed Convolution	$(4, 4)$	$(2, 2)$	$(N/2 \times R/2 \times 32)$
ReLU			$(N/2 \times R/2 \times 32)$
2-D Transposed Convolution	$(4, 4)$	$(2, 2)$	$(N \times R \times 32)$
ReLU			$(N \times R \times 32)$
2-D Convolutional	$(1, 1)$		$(N \times R \times 2)$
Softmax			
Dice pixel classification			

element of the matrix is managed as if it were a pixel. First, the matrix $\mathbf{C}^{(abs)}$ is fed into a first neural network, called DNN1, which is responsible for discriminating targets based on delay. Each output delay estimated by DNN1 is fed to DNN2 to obtain the normalized Doppler shifts associated with that tap. The architecture of both networks is the same and it is shown in Table I. The DNNs are semantic segmentation networks, which means that they assign a class label to each individual pixel of the input image. The architecture uses first CNNs and max pooling layers to perform downsampling, followed by transposed CNN layers for upsampling. The activation function used is the rectified linear unit (ReLU).

DNN1 is responsible for assigning a '1' or a '0' to every element of $\mathbf{C}^{(abs)}$ if it considers that there is a tap or not, respectively. In this first step, taps are distinguished only by their delay. Based on the delays found by DNN1, the indexes of the $\mathbf{C}^{(abs)}$ columns where there is a tap are extracted. Each column (for each delay l_i) is transformed to logarithmic (for a better resolution) and is given to DNN2, which is responsible for finding the Doppler shifts. Before feeding the column into DNN2, it is expanded to obtain a matrix of B replicas of the column, as shown in Fig. 1. This is done to combat the class imbalance problem. This problem is due to the disproportion between the number of classes of type '1' (tap) and type '0' (non-tap). The number of '1' labels is usually much smaller than the number of '0' labels. That is, we aim to distinguish objects of the size of a single pixel within a column of size N, which is complicated for the DNN and gives unsatisfactory results. Therefore, by extending the column in a matrix, a better training for DNN2 is achieved. In addition, both DNNs use the generalized Dice loss function. This function controls the contribution of each class to the loss, weighting the labels by the inverse size of the expected region [19]. The generalized Dice loss between an image $\mathbf{C}$ and the ground truth $\mathbf{T}$, in case the number of classes is 2, can be expressed as

$$L_{Dice} = 1 - \frac{2\sum_{n=1}^{2} w_n \sum_{s=1}^{NR} [\mathbf{C}]_{ns} [\mathbf{T}]_{ns}}{\sum_{n=1}^{2} w_n \sum_{s=1}^{NR} [\mathbf{C}]_{ns}^2 + [\mathbf{T}]_{ns}^2}, \tag{19}$$

where w_n is the weight, for $n = 1, 2$ (two classes) assigned to each class, which corresponds to the squared inverse of the area of the expected region

$$w_n = \left(\sum_{s=1}^{NR} [\mathbf{T}]_{ns} \right)^{-2}. \tag{20}$$

The DNNs were trained with the following parameters:

- Dataset for training was obtained from simulations with parameters from Table II. A total number of 19,960 correlation matrices $\mathbf{C}^{(abs)}$ was used. For DNN1, 70% of these matrices were used for training and 30% for validation. In the case of DNN2, we used the columns of these same matrices. The DNNs are trained for SNR of 20 dB. The training and validation data include matrices with a total of 1,000 different channels, with $-N/2 \leq k_i < N/2$ and $l_i < l_{max}$ with $l_{max} = L_{ZP} = 16$. The channel complex gains g_i are randomly generated following a Rayleigh distribution. This means that the DNN is robust to changes in the channel conditions or the targets scenario, which is useful to reduce the number of times the DNN needs to be trained during the communication. The training can be performed offline. We used both $R = 16$ and $B = 16$ for the sizes of the input data.
- Adam optimizer [20], learning rate of 0.001 and mini-batch size of 64.
- The training was made for 450 iterations (DNN1) and 2700 (DNN2).

A. Conventional Choice of the Threshold

This section illustrates the conventional scheme to the DNN approach for choosing γ. Ideally, γ should be several times

TABLE II
SIMULATION PARAMETERS

Parameter	Value
Delay and Doppler bins	M=512, N=16
Carrier frequency	f_c = 26 GHz
Subcarrier spacing	Δf = 60 kHz.
Modulation	QPSK

higher than the noise and interference present in the matrix, in order to avoid them. The main sources of noise and interference are

- The effects of noise and self-interference of the data within $\mathbf{C}^{(\text{abs})}$ are proportional to $\sigma^2_{z''}$ and $\sigma^2_{x''}$.
- Furthermore, additional interference σ^2_{κ} is introduced by the correlation between the Doppler reference vectors $\boldsymbol{\kappa}_{k,p}$ and the taps in the received averaged vector $\overline{\mathbf{y}}$. Since only the first M samples of these vectors are correlated, then (13) is not strictly satisfied. The value of this interference directly depends on the complex channel gains g_i of each tap, as given in (15). This is a problem because the values of g_i are estimated after this procedure, so they are unknown at this point.

Hence, we can give a lower-bound to γ, so that

$$\gamma > \sqrt{\sigma^2_{x''} + \sigma^2_{z''} + \sigma^2_{\kappa}} = \sigma_T, \tag{21}$$

where we define $\sigma^2_T = \sigma^2_{x''} + \sigma^2_{z''} + \sigma^2_{\kappa}$ as the total interference power at the correlation matrix $\mathbf{C}^{(\text{abs})}$. In practice, however, the variances are not straightforward to compute. A realistic application implies a prior estimation of this interference. Whenever the matrix $\mathbf{C}^{(\text{abs})}$ is sufficiently sparse (i.e. the number of propagation taps or targets $L_p << MN$) or, equivalently, $MN \to \infty$, applying fundamental statistical analysis, it can be demonstrated that the total interference power of the matrix can be estimated with the average power of the matrix as

$$\sigma^2_T \approx \frac{1}{MN} \sum_{k=1}^{N} \sum_{l=1}^{M} [\mathbf{C}^{(\text{abs})}]^2_{k,l}. \tag{22}$$

In practice, of course, this approximation may lead to a poor computation of γ if the matrix is not sparse (there are a large number of targets) or if M or N are not high enough. Setting the threshold above the interference level, i.e. how many levels above it to set, is also a difficult decision in practice. Given this, and its direct dependence on the channel gains, it will be difficult to update this calculation in real time during communication.

In fact, the threshold decision expression (22) can be viewed as a simple form of constant false alarm rate (CFAR) detection. This problem is well-known in the radar literature. To solve this threshold decision problem for radar target detection, several variants of CFAR have been proposed [21]. However, many CFAR techniques also require specific conditions to work properly. For example, in cell-averaging CFAR (CA-CFAR), the threshold in each cell (for each point in the matrix) is calculated based on the average noise power of adjacent cells. This provides near-optimal performance for homogeneous scenarios. However, when additional sources of interference or clutter are present, the interference estimation is biased and can lead to an excessive number of false alarms or target masking. Other more sophisticated CFAR techniques can be used as an alternative, but the authors of [22] provide an interesting analysis of CFAR detectors, which concludes that no technique will be suitable for every combination of homogeneous and non-homogeneous background noise, and suggests that other adaptive techniques may provide the best solution for changing operating conditions.

Therefore, DNN-based approaches can be easily adapted to the problem and can be a viable solution. DL-based radar target detection techniques have been proposed for other applications [1]. In this paper, the DNN designed for tap detection in our ST-OTFS technique for ISAC is successfully adapted to this scenario and shows that it achieves good results in terms of detection probabilities and false alarms.

V. SIMULATION RESULTS

In this section, we present some results to evaluate the performance of the proposed DNN for tap detection. The simulation parameters are the ones in Table II.

The system is evaluated in terms of detection and false alarm probabilities. To calculate the probability of detection, the detected $\tilde{L}_p$ targets $(\tilde{k}_i, \tilde{l}_i)$ are compared with the database of true targets. If the target is found, it is considered detected. If a target is detected but not found in the database, it is considered a false alarm. A tap is considered detected if it is within a minimum distance of the true target. The correct Doppler shift estimate is based on a distance of $d_k = \pm 1$, while for the delay shift it is $d_l = 0$, where $d_k = k_i - \tilde{k}_i$ and $d_l = l_i - \tilde{l}_i$. The expressions for the probability of detection and false alarm are

$$P_d = \frac{L_{det}}{L_p}, \;\; P_{fa} = \frac{L_{fa}}{ML_{ZP} - L_p} \tag{23}$$

where L_{det} is the number of detected targets and L_{fa} is the number of false alarms, so that $\tilde{L}_p = L_{det} + L_{fa}$.

Fig. 2 shows a comparison between the proposed DNN and the conventional threshold decision techniques based on the probability of detection P_d. In this figure, the ideal γ calculation is considered as the upper-bound defined in (21), where the interference value σ^2_T is assumed to be known. To avoid the interference levels, the threshold is chosen to be slightly larger than σ^2_T, in particular $\gamma = 4\sigma_T$. This gives a constant false alarm rate of approximately $P_{fa} = 4 \cdot 10^{-4}$, which is shown in Fig. 3. We also show γ_r, which is the realistic or practical computation of the threshold based on the estimated interference power, i.e., $\tilde{\sigma}^2_T$ in (22). We choose $\gamma_r = 3\tilde{\sigma}_T$, which gives a false alarm rate of $P_{fa} = 10^{-4}$.

In Fig. 2 and Fig. 3, the conventional practical method of threshold estimation results in a γ_r with the lowest P_{fa}, but at the expense of a low detection probability P_d, which indicates that this threshold may not be the most appropriate. Conversely, the DNN approach exhibits a detection capability that closely resembles the ideal case. It achieves this at the

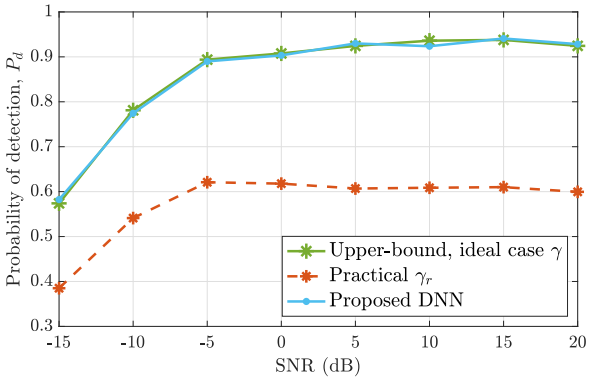

Fig. 2. Probability of detection P_d, for the upper-bound case γ (21), practical γ_r (22) and proposed DNN.

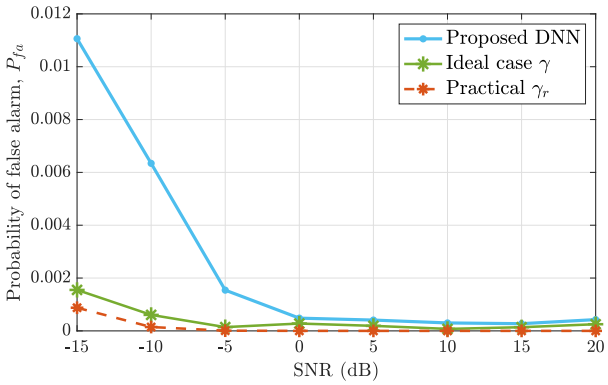

Fig. 3. Probability of false alarm for the proposed DNN.

expense of a slightly higher false alarm probability for low SNR regime (SNR $<$ 0 dB), while for SNR $\geq$ 0 dB it achieves an average false alarm probability of about $P_{fa} = 4 \cdot 10^{-4}$, which is almost identical to that of the ideal scenario. And, given that for communications to be successful, the scenario will rarely exhibit such low SNR, it can be concluded that the DNN surpasses the conventional γ_r method in both detection accuracy and false alarm rates, approaching an ideal performance.

VI. Conclusions

In this work, we enhance our previously proposed channel estimation technique with ST-OTFS for ISAC [10] by incorporating a DNN for tap detection. Traditionally, target detection relies on selecting a threshold, which is challenging without prior knowledge of the channel conditions and interference. Choosing an incorrect threshold compromises the system performance, leading to higher probabilities of miss-detection or false alarm. In this study, we introduce a DNN-based approach that eliminates the need for threshold decisions. The proposed DNN architecture, which requires only a moderate number of layers, can be trained offline. Moreover, it demonstrates robustness to changes in channel conditions, reducing the frequency of model training during the communication.

Acknowledgments

This work has been funded by the Spanish National project IRENE-EARTH (PID2020-115323RB-C33/AEI /10.13039/501100011033) and the work of K. Chen-Hu was also, in part, supported by the Villum Investigator Grant "WATER" from the Velux Foundation, Denmark.

References

[1] A. Masaracchia, V. Sharma, B. Canberk, O. A. Dobre, and T. Q. Duong, "Digital twin for 6G: Taxonomy, research challenges, and the road ahead," *IEEE Open J. Commun. Soc.*, vol. 3, pp. 2137–2150, Nov, 2022.

[2] J. Lee, A. A. Badrudeen, and S. Kim, "6G integrated sensing and communication: Recent results and future directions," in *ICTC 2O22*, 2022, pp. 1219–1221.

[3] R. Hadani and A. Monk, "OTFS: A new generation of modulation addressing the challenges of 5G," 2018.

[4] Y. Hong, T. Thaj, and E. Viterbo, *Delay-Doppler Communications: Principles and applications*. Elsevier, 2022.

[5] O. Zacharia and M. Vani Devi, "Fractional delay and doppler estimation for OTFS based ISAC systems," in *IEEE WCNC 2023*, 2023, pp. 1–6.

[6] S. Li, W. Yuan, J. Yuan, and G. Caire, "On the potential of spatially-spread orthogonal time frequency space modulation for isac transmissions," in *IEEE ICASSP 2022*, 2022, pp. 8722–8726.

[7] W. Shen, L. Dai, S. Han, I. Chih-Lin, and R. W. Heath, "Channel estimation for orthogonal time frequency space (OTFS) massive MIMO," in *ICC 2019*, May 2019, pp. 1–6.

[8] H. B. Mishra, P. Singh, A. K. Prasad, and R. Budhiraja, "OTFS channel estimation and data detection designs with superimposed pilots," *IEEE Trans. Wirel. Commun.*, vol. 21, no. 4, pp. 2258–2274, Apr. 2022.

[9] Y. Wu, C. Han, and Z. Chen, "DFT-spread orthogonal time frequency space system with superimposed pilots for terahertz integrated sensing and communication," *IEEE Trans. Wirel. Commun.*, pp. 1–1, Early Access 2023.

[10] L. Méndez-Monsanto Suárez, K. Chen-Hu, M. J. Fernández-Getino García, and A. García Armada, "Robust integrated sensing and communications in delay-Doppler domain using superimposed training," *submitted to IEEE GLOBECOM 2023*, 2023.

[11] C. Yang, J. Wang, Z. Pan, and S. Shimamoto, "Delay-doppler frequency domain-aided superimposing pilot OTFS channel estimation based on deep learning," in *IEEE VTC 2022-Fall*, 2022, pp. 1–6.

[12] X. Zhang, W. Yuan, C. Liu, F. Liu, and M. Wen, "Deep learning with a self-adaptive threshold for otfs channel estimation," in *ISWCS 2022*, 2022, pp. 1–5.

[13] Q. Li, Y. Gong, F. Meng, L. Han, and Z. Xu, "A novel channel estimation method based on deep neural network for otfs system," in *CISP-BMEI 2022*, 2022, pp. 1–6.

[14] Q. Cheng, Z. Shi, J. Yuan, and M. Zhou, "Environment-robust signal detection for otfs systems using deep learning," in *IEEE GLOBECOM 2022*, 2022, pp. 5219–5224.

[15] O. A. Aghda, M. J. Omidi, and H. Saeedi-Sourck, "Superimposed channel estimation in OTFS modulation using compressive sensing," 2022.

[16] P. Raviteja, K. T. Phan, and Y. Hong, "Embedded pilot-aided channel estimation for OTFS in delay–doppler channels," *IEEE Trans. on Veh. Technol.*, vol. 68, no. 5, pp. 4906–4917, 2019.

[17] H. B. Mishra, P. Singh, A. K. Prasad, and R. Budhiraja, "Iterative channel estimation and data detection in OTFS using superimposed pilots," in *ICC 2021*, 2021, pp. 1–6.

[18] B. Li and W. Yuan, "OTFS communications-assisted environment sensing," in *IEEE JC&S 2023*, 2023, pp. 1–6.

[19] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," in *3DV 2016*, 2016, pp. 565–571.

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[21] D. Gusland, S. Rolfsjord, and B. Torvik, "Deep temporal detection - a machine learning approach to multiple-dwell target detection," in *2020 IEEE International Radar Conference (RADAR)*, 2020, pp. 203–207.

[22] P. Gandhi and S. Kassam, "Analysis of CFAR processors in nonhomogeneous background," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 24, no. 4, pp. 427–445, 1988.