

Deep Learning of GNSS Signal Detection

A Dissertation Presented

by

Parisa Borhani-Darian

to

The Department of Electrical and Computer Engineering

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

in

Computer Engineering

Northeastern University

Boston, Massachusetts

March 2023

To my family.

Contents

List of Figures	iv
List of Acronyms	vii
Acknowledgments	ix
Abstract of the Dissertation	x
1 Introduction	1
1.1 Dissertation Outline	4
1.2 Dissertation Publications	5
2 GNSS Signal Acquisition in the absence of Spoofer	8
2.1 Deep learning of GNSS signal acquisition	9
2.1.1 GNSS Signal Models	9
2.1.2 CAF Evaluation	11
2.1.3 Searching Strategies	13
2.1.4 Benchmark performance using the Receiver Operating Characteristic function	14
2.1.5 Deep Learning Approach	15
2.1.6 Simulation Setup and DNN training	18
2.1.7 Result	20
2.2 Parallelizable deep learning acquisition by CAF splitting and data fusion	25
2.2.1 Sub-Image Model structure	25
2.2.2 Non-coherent integration through fusion of classifiers	28
2.2.3 Model training	31
2.2.4 Results	32
3 GNSS Signal Acquisition in the presence of Spoofer	38
3.1 Deep learning of GNSS spoofing signals	38
3.1.1 GNSS Spoofer detection	39
3.1.2 Deep Neural Networks model and problem statement	41
3.1.3 Probabilistic learning	43
3.1.4 Simulation Environment and results	44
3.2 Parallelizable deep learning spoofer detection by CAF splitting and data fusion	51

3.2.1	GNSS signal spoofing effects on acquisition	52
3.2.2	Data-driven GNSS Spoofing detection	52
3.2.3	Probabilistic signal detection	58
3.2.4	Estimating number of signals	60
3.2.5	Results	62
4	Conclusion	66
	Bibliography	68

List of Figures

2.1	CAF evaluation at the delay/Doppler grid in the (a) absence and (b) presence of a signal with $C/N_0 = 39$ dB-Hz.	9
2.2	Detection scheme of GNSS acquisition by using the Fully Connected Structure of Deep Learning approach.	17
2.3	Detection scheme of GNSS acquisition by using the Convolutional Neural Network Structure of Deep Learning approach.	18
2.4	ROC curves for MLP and CNN using CAF generated with (a,b) 1 ms coherent integration and (c,d) 1 ms coherent integration and 10 non-coherent integrations.	20
2.5	$P_d(\gamma)$ and $P_{fa}(\gamma)$ under 1 ms coherent integration for MLP and CNN models.	21
2.6	$P_d(\gamma)$ and $P_{fa}(\gamma)$ under 1 ms coherent and 10 non-coherent integrations for MLP and CNN models.	22
2.7	Histogram of $\mathcal{T}(\mathbf{Z}_i)$ under \mathcal{H}_0 and \mathcal{H}_1 hypotheses for each Neural Network (NN) and two different C/N_0 values.	23
2.8	ROC curves for MLP and CNN using different configurations to generate CAF inputs for training and testing.	24
2.9	Classification of signal (\mathcal{H}_1) or noise (\mathcal{H}_0) in CAFs as part of the proposed GNSS signal acquisition scheme. Particularly, a set of convolutional layers followed by Fully-Connected layers provide the capabilities of deep learning from large datasets.	26
2.10	Proposed acquisition method where the CAF map \mathbf{Z}_i is split into smaller sub-images $\mathbf{Z}_i^{(m,n)}$, which are fed to a bank of parallel DNN binary classifiers to produce probability ratio maps. To increase accuracy, several ($K > 1$) probability ratio maps can be non-coherently fused, as shown on the rightmost plot.	27
2.11	Portions of the CAF fed for processing to the NN with $\Delta_m = 18$ and $\Delta_n = 5$ defining the size of the $\{m, n\}$ -th sub-image The resulting sub-image $\mathbf{Z}_i^{(m,n)}$ is shown on the reduced delay/Doppler grid in the case of (a) absence and (b) presence of a GNSS signal with $C/N_0 = 39$ dB-Hz. In the absence of signal, samples are i.i.d., while a time-freq correlation can be observed in the presence of signal.	28
2.12	Comparison of the delay/Doppler grid for (a) standard CAF map with coherent integration only, (b) probability map produced by the data-driven classifier with coherent integration only, and (c) probability map after fusing $K = 6$ non-coherently classifier outputs. The GNSS signal had a C/N_0 of 42 dB-Hz, and the red circled highlights the location of the peak generated by the GNSS signal.	30

2.13	ROC curves for a 1 ms coherently integrated snapshot and $K = 6$ non-coherently processed blocks for a variety of C/N_0 values. The performance of the proposed scheme (dashed lines) is compared to the theoretical performance of standard methods (solid lines).	33
2.14	(a) $P_d(\gamma)$ and (b) $P_{fa}(\gamma)$ probabilities for a 1 ms coherently integrated snapshot, $K = 6$ non-coherent processing, and a variety of C/N_0 values.	34
2.15	Test statistic histograms under \mathcal{H}_0 and \mathcal{H}_1 hypotheses for a 1 ms coherent integration time for a range of relevant C/N_0 values. The two histograms have overlapping areas, which suggest poor detection performance in these conditions.	35
2.16	Test statistic histograms under \mathcal{H}_0 and \mathcal{H}_1 hypotheses for a 1 ms coherent integration time and $K = 6$ non-coherent integrations for a range of relevant C/N_0 values. In this case the two histograms are clearly separated, which supports the performance results in Fig. 2.13.	36
3.1	Deep learning signal detection process in the presence of spoofer	39
3.2	CAF evaluation at the delay/Doppler grid with $C/N_0 = 45$ dB-Hz.	40
3.3	Detection scheme of GNSS acquisition by using the Fully Connected Structure of Deep Learning approach.	42
3.4	Detection scheme of GNSS acquisition by using the Convolutional Neural Network Structure of Deep Learning approach.	43
3.5	ROC curves for Simple-CNN and MLP using CAF generated with 1 ms coherent integration and 10 non-coherent integration	46
3.6	P_d and P_{fa} under 1 ms coherent and 10 non-coherent integrations for all three networks models.	47
3.7	$P_d(\gamma)$ and $P_{fa}(\gamma)$ under 1 ms coherent and 10 non-coherent integration for Complex-CNN models.	48
3.8	$P_d(\gamma)$ and $P_{fa}(\gamma)$ under 1 ms coherent and 10,20,30 and 50 non-coherent integration for Complex-CNN models.	49
3.9	Histogram of $\mathcal{T}(\mathbf{Z}_i)$ under \mathcal{H}_0 and \mathcal{H}_1 hypotheses for each NNs and two different C/N_0 values.	50
3.10	CAF evaluation at the delay/Doppler grid with $C/N_0 = 45$ dB-Hz.	51
3.11	Portions of the CAF fed for processing to the NN with $\Delta_m = 18$ and $\Delta_n = 5$ defining the size of the $\{m, n\}$ -th sub-image. The resulting sub-image $\mathbf{Z}_i^{(m,n)}$ is shown on the reduced delay/Doppler grid in the case of (a) absence and (b) presence of a GNSS spoofed signal with $C/N_0 = 45$ dB-Hz.	53
3.12	Classification of signal (\mathcal{H}_0) or signal (\mathcal{H}_0) and Spoofer (\mathcal{H}_1) in CAFs as part of the proposed GNSS signal acquisition scheme. Particularly, a set of convolutional layers followed by Fully-Connected layers provide the capabilities of deep learning from large datasets.	54
3.13	Proposed acquisition method where the CAF map \mathbf{Z}_i is split into smaller sub-images $\mathbf{Z}_i^{(m,n)}$, which are fed to a bank of parallel DNN binary classifiers to produce probability ratio maps. To increase accuracy, several ($K > 1$) probability ratio maps can be non-coherently fused, as shown on the rightmost plot.	55

3.14	Comparison of the delay/Doppler grid in a presence of spoofer for (a) standard CAF map with coherent integration only, (b) probability map produced by the data-driven classifier with coherent integration only, and (c) probability map after fusing $K = 6$ non-coherently classifier outputs. The GNSS signal had a C/N_0 of 42 dB-Hz and the red circle highlights the location of the peak generated by the GNSS signal. . .	59
3.15	Running example showing the process followed by the proposed algorithm. The experiment consists of a legitimate signal and two spoofers with $C/N_0 = 42$ dB-Hz. The various panels show the corresponding (a) CAF; (b) probability ratio map; (c) top view of the threshold probability ratio maps, after clustering is applied; and (d) three-dimensional perspective of the latter with the probability ratio maps overlaid.	61
3.16	ROC curves for a 1 ms coherently integrated snapshot and $K = 6$ non-coherently processed blocks for a variety of C/N_0 values for signal and spoofer. The performance of the proposed scheme (dashed lines) is compared to the theoretical performance of standard methods (solid lines).	63
3.17	(a) $P_d(\gamma)$ and (b) $P_{fa}(\gamma)$ probabilities for a 1 ms coherently integrated snapshot, $K = 6$ non-coherent processing, and a variety of C/N_0 values when signal and Spoofer are present.	64
3.18	Probability of correctly detecting both legitimate and spoofing signals, as well as determining their number, as a function of their relative delay separation. Also, the same Doppler, one bin, and two bin separations where considered. Bin size being 500 Hz.	65

List of Acronyms

GNSS Global Navigation Satellite Systems

DNN Deep Neural Network

CNN Convolution Neural Networks

MLP Multi Layer Perceptron

CAF Cross Ambiguity Function

PNT Position, Navigation, and Timing

NN Neural Network

GPU Graphics Processing Units

ROC Receiver Operating Characteristic

RMSE Root-Mean-Squared Error

SIS Signal-In-Space

SVM Support Vector Machines

PNN Probabilistic Neural Networks

DT Decision Tree

KNN K-Nearest Neighborhood

PSO Particle Swarm Optimization

ML Machine Learning

LSTM long short-term memory

C-SVM classification SVM

PCA Principal Component Analysis

GAN General Adverse Network

PRN Pseudo-Random Noises
ML Maximum Likelihood
MLE Maximum Likelihood Estimation
GLRT Generalized Likelihood Ratio Test
PRN Pseudo-Random Noises
SGDM Stochastic Gradient Descent with Momentum
ReLU Rectified Linear Unit
FC Fully Connected
GPS Global Positioning System
Nnci Number of non-coherent integration
RNN Recurrent Neural Network
GMM Gaussian Mixture Model
EM Expectation-Maximization
BIC Bayesian Information Criterion
CAV connected and autonomous vehicles
TEXBAT Texas Spoofing test Battery

Acknowledgments

Here I wish to thank those who have supported me during the process of the dissertation work, especially my supervisor, Dr. Pau Closas, for providing guidance and feedback throughout this project. I would like to thank my husband Miead and my family for supporting me during the compilation of this dissertation.

Abstract of the Dissertation

Deep Learning of GNSS Signal Detection

by

Parisa Borhani-Darian

Doctor of Philosophy in Computer Engineering

Northeastern University, March 2023

Dr. Pau Closas, Advisor

Global Navigation Satellite Systems (GNSS) is the *de facto* technology for Position, Navigation, and Timing (PNT) applications when it is available. GNSS relies on one or more satellite constellations that transmit ranging signals, which a receiver can use to self-localize. Signal acquisition is a crucial step in GNSS receivers, which is typically solved by maximizing the so-called Cross Ambiguity Function (CAF) resulting from a hypothesis testing problem. The CAF is a two-dimensional function that is related to the correlation between the received signal and a local code replica for every possible delay/Doppler pair, which is then maximized for signal detection and coarse synchronization. The outcome of this statistical process decides whether the signal from a particular satellite is present or absent in the received signal, as well as provides a rough estimate of its associated code delay and Doppler frequency if present. Critical infrastructures and safety-critical applications, such as in the context of connected and autonomous vehicles (CAV), heavily rely on GNSS for PNT purposes and as a consequence, these services are vulnerable to degradation and deliberate attacks on GNSS, for which this thesis proposes data-driven approaches to address those vulnerabilities.

With the increased popularity of artificial intelligence, machine learning, and deep learning begin to play an important role in adapting traditional algorithms in a variety of disciplines, especially when it comes to estimation and classification tasks where remarkable results have been demonstrated. The contributions of this work include (i) the use of data-driven models, popular in the machine learning literature, as an alternative to well-engineered signal processing blocks used in state-of-the-art GNSS receivers to perform signal acquisition. Particularly, with and without spoofing attacks. The CAFs are fed to a data-driven classifier that outputs binary class posteriors, which are used in a Bayesian hypothesis test to statistically decide the presence or absence of a legitimate GNSS signal or a spoofing signal; (ii) to increase the detection accuracy with less computational complexity by

leveraging data-driven methods. The versatility and computational affordability of the proposed methods are addressed by splitting the CAF into smaller overlapping sections, which are fed to a bank of parallel classifiers whose probabilistic results are optimally fused to provide a so-called probability ratio map from which signal detection is inferred; *(iii)* the research shows how non-coherent integration schemes are enabled through optimal data fusion, to increase the resulting classifier accuracy and *(iv)* to estimate the number and parameters of both the legitimate and spoofing signals using machine learning clustering algorithms such as Gaussian Mixture Model (GMM). This thesis provides simulation results showing that the proposed data-driven method outperforms current CAF maximization strategies, enabling enhanced acquisition at lower carrier-to-noise density ratios.

Chapter 1

Introduction

GNSS is the *de facto* technology for PNT applications, when it is available [1, 2, 3, 4, 5]. GNSS relies on one or more satellite constellations that transmit ranging signals, which a receiver can use to self-localize. Along the signal processing chain, the first step that is performed by a GNSS receiver is signal acquisition. The outcome of this statistical process decides whether the signal from a particular satellite is present or absent in the received signal, as well as provides a rough estimate of its associated code delay and Doppler frequency, if present. All GNSS receivers [6, 7, 8, 9] implement such an acquisition process by evaluating the so-called CAF and maximizing it. The CAF is a two-dimensional function that is related to the correlation between the received signal and a local code replica for every possible delay/Doppler pair, which is then maximized for signal detection and coarse synchronization.

Signal acquisition is based on solid statistical grounds, where the approach of maximizing the CAF (i.e., the correlation between the local replica and the incoming signal) can be seen to be optimal under certain model conditions (e.g. Gaussianity of the channel). However, experiments show (e.g. [10]) that reality is typically more challenging and that the assumed nominal model conditions do not always hold true necessarily. Recent attempts to modify the CAF to make it more robust to non-Gaussian behaviors (such as heavy-tailed noise distributions) showed outstanding performance, particularly in the context of GNSS operation under jamming [11, 12, 13, 14, 15, 16, 17, 18]. Regardless of its remarkable performance in outlier-rich data, the aforementioned robust approach does not accommodate for more complex situations such as multi-modal distributions or moderate-to-severe nonlinearities affecting the received signal. In [19], the authors showed preliminary results that such complex behaviors can be learned by employing efficient data-driven methods, trained over large datasets.

CHAPTER 1. INTRODUCTION

Location-based services, alongside the modern applications of Intelligent Transportation Systems, require reliable, continuous, and precise navigation, positioning and timing information for their successful operation and implantation in the market. GNSS receivers are very sensitive and vulnerable to deliberate interference, which opens the door for attackers who want to compromise a GNSS-based system or infrastructure, causing serious impacts. Due to the lack of inherent security design in GNSS systems, many applications could be potentially at risk, as reported in numerous articles. Deliberately attacks on GNSS receivers might act in two different categories: physical attacks on the receiver (non-signal attacks) or attacks at the GNSS Signal-In-Space (SIS) level (signal attacks) [20]. This research is working on the second category (signal attacks), which are intentional attacks on GNSS signals, which have three different forms: jamming, meaconing, and spoofing. Moreover, the focus of this work is on spoofing, which is the transmission of forged GNSS-like signals, with the purpose to produce a false position at the victim's receiver without disrupting GNSS operations, effectively taking control of the receiver. Notice that jamming attacks aim at denying GNSS positioning service, an opposite goal to spoofing interference.

Nowadays, the advent of deep learning as a popular tool has sped up advances in a myriad of disciplines. In short, deep learning algorithms (for instance, the variety of NN architectures currently available) are data-driven models that, instead of using complex-to-derive physics-based models, use large datasets to learn the correlations in the data. In the context of GNSS, deep learning has been recently investigated in several domains, with [21] providing an excellent summary. Some works explore the use of Deep Neural Network (DNN) as multipath mitigation strategies, one of those situations where a physics-based model is either too complex to be used or not available at all. For instance, [22] presented a deep learning-based beamforming approach to mitigate multipath. That work highlighted the limitations of conventional beamforming algorithms by developing a DNN-based model and applying it in different environments, showing a Root-Mean-Squared Error (RMSE) reduction. [23] discussed the benefits of DNN in predicting distortions in the urban area, which cause significant degradation to GNSS performance. This is improved by leveraging a DNN to extract useful features from the data to learn GNSS measurement quality for improved prediction (of satellite visibility and pseudo-range errors) in urban areas. The work [24] proposed an end-to-end deep learning method for satellite selection based on the PointNet and VoxelNet networks, as a promising alternative to standard selection procedures. The work in [25, 26] presented a methodology to substitute the CAF calculation (typically performed through local code correlation) with a DNN method that was able to learn the complexities of the multipath channel, with promising results when used in standard tracking loops. Those, and other works [27, 28, 29, 30], highlight the relevance

CHAPTER 1. INTRODUCTION

and popularity that this topic is gaining in the GNSS multipath mitigation challenge. On another set of GNSS applications, the impact of the deep learning approaches to counteract GNSS spoofing [31, 32, 33, 34, 35] and jamming [36, 37] attacks is presented in several works. In the context of GNSS for Earth sciences, deep learning was considered for earthquake prediction [38], hurricane monitoring [39], ice detection [40], and ionospheric scintillation [41, 42, 43, 44]. In [45] to obtain a higher detection probability of the Global Positioning System (GPS) spoofing, a general identification scheme with decision fusion is used. The singular values of the wavelet transform coefficients of both spoofing and genuine signal are considered as feature vectors, that are input into three classifiers, which are the Support Vector Machines (SVM), the Probabilistic Neural Networks (PNN) and the Decision Tree (DT), respectively, for GPS spoofing identification. The results of the three classifiers are fused with a K-out-of-N decision rule, and the final classification result has a higher probability of detection and lowers false alarms.

This [46] presents a method to detect a GPS spoofing based on Multi-Layer NN whose inputs are indices of features to perform spoofing detection by exploiting conventional machine learning algorithms such as K-Nearest Neighborhood (KNN) and naive Bayesian classifier. [33] also used a Multi Layer Perceptron (MLP) neural network classifier trained by Particle Swarm Optimization (PSO), with input as received signal power and correlation function distortion. Other Maximum Likelihood (ML) algorithms used to detect spoofing include Recurrent Neural Network (RNN) based on long short-term memory (LSTM), classification SVM (C-SVM) with Principal Component Analysis (PCA), and General Adverse Network (GAN) based method, as seen in [34, 35, 47, 48] respectively.

The main objective of this dissertation is to provide a complete and cohesive analysis of the acquisition process. A general methodology is developed and applied to:

- The acquisition of GNSS signals in the absence of Spoofer
- The acquisition of GNSS signals in the presence of Spoofer

This acquisition process can be regarded as a signal detection problem, where two hypotheses are available: 1) the null hypothesis \mathcal{H}_0 , and 2) the alternative hypothesis \mathcal{H}_1 . Three probabilities characterize the performance of the acquisition method: detection (the probability of correctly detecting signal/noise when there is signal/noise); false-alarm (the probability of wrongly detecting signal when the satellite is not present); and miss-detection (the probability of mistakenly deciding for the null hypothesis when the signal is present). Detection and false-alarm probabilities are used to obtain

CHAPTER 1. INTRODUCTION

an important figure of merit in hypothesis tests: the Receiver Operating Characteristic (ROC), which is a plot of the probability of detection as a function of the probability of false alarm [6, 49, 50].

In particular, the work proposed a MLP and two classes of Convolution Neural Networks (CNN) to detect the legitimate signal and the existence of spoofing signals, using the CAF as an input feature to use DNN models to carry out the detection (or classification) process involved in the signal acquisition, which are explained in chapter2 with the outcome of two papers [19, 32] respectively. The research (i) enhanced DNN models that improve flexibility and computational complexity through a dataset splitting and parallel DNN processing; (ii) allow for non-coherent integration times within the DNN framework through an optimal data fusion step. (iii) in a presence of the spoofer the ML clustering is used to detect the number of the spoofer signals, which are explained in chapter 3 respectively with the outcome of the paper [51]; and (iv) providing a more detailed discussion of results and design tradeoffs for practitioners.

1.1 Dissertation Outline

This dissertation is organized into two parts that respectively deal with the signal acquisition process in the absence of spoofer and spoofer detection.

Chapter 2 :GNSS Signal Acquisition in the absence of Spoofer

- Deep learning of GNSS signal acquisition
- Parallelizable deep learning acquisition by CAF splitting and data fusion

Chapter 3: GNSS Signal Acquisition in the presence of Spoofer

- Deep learning of GNSS spoofing signals
- Parallelizable deep learning spoofer detection by CAF splitting and data fusion

Chapter 4: Conclusion and future works

1.2 Dissertation Publications

- Journal Papers

The outcome of parallelizable deep learning signal acquisition and spoofer detection by CAF splitting and data fusion are two journal papers, which are as follows:

1. P. Borhani-Darian, H. Li, P. Wu, P. Closas, “Detecting GNSS spoofing using deep learning” *IEEE Transactions on Aerospace and Electronic Systems*. Submitted on March 2023.
2. P. Borhani-Darian, H. Li, P. Wu, and P. Closas, “Deep Learning of GNSS Acquisition,” *Sensors*, vol. 23, no. 3, p. 1566, 2023.

- Conference Papers

The outcome of deep learning of GNSS signal acquisition and spoofing signals detection are two conference papers, which are as follows:

1. P. Borhani-Darian and P. Closas, “Deep Neural Network Approach to GNSS Signal Acquisition,” in *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, 2020, pp. 1214–1223.
2. P. Borhani-Darian, H. Li, P. Wu, and P. Closas, “Deep Neural Network Approach to Detect GNSS Spoofing Attacks,” in *Proceedings of the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2020)*, 2020, pp. 3241–3252

Additionally, during the completion of this dissertation, I participated in a variety of research projects in collaboration with others. These efforts resulted in publications (two journals and one conference) for which I am a co-author.

1. H. Li, P. Borhani-Darian, P. Wu, and P. Closas, “Deep neural network correlators for GNSS multipath mitigation,” *IEEE Transactions on Aerospace and Electronic Systems*, Accepted 2022.
2. N. Williams, P. B. Darian, G. Wu, P. Closas, and M. Barth, “Impact of Positioning Uncertainty on Connected and Automated Vehicle Applications,” *SAE International Journal of Connected and Automated Vehicles*, vol. 6, no. 12-06-02-0010, 2022.

CHAPTER 1. INTRODUCTION

3. H. Li, P. Borhani-Darian, P. Wu, and P. Closas, “Deep Learning of GNSS Signal Correlation,” in Proceedings of the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2020), 2020, pp. 2836–2847.

Part I

Signal Acquisition

Chapter 2

GNSS Signal Acquisition in the absence of Spoofer

Signal acquisition is one of the first actions a receiver needs to perform, basically deciding whether the signal from a particular satellite is present or absent, as well as providing a rough estimation of the code delay and Doppler frequency of the received signal in case it is deemed present [6]. Ideally, the CAF should present a sharp peak that corresponds to the values of τ and FD matching the delay and the Doppler frequency of the SIS. However, the phase of the incoming signal, the noise, and other impairments can degrade the readability of the CAF, in which case further processing is needed. This work conjectures that such complex behaviors can be learned by employing efficient data-driven methods, trained over large datasets. In particular, we propose to use a DNN to carry out the detection (or classification) process. Prior to targeting those challenging scenarios that turn the nominal model unreliable, this work focuses on replicating the optimal results (provided by standard model-driven methods under nominal conditions) by employing data-driven models (a.k.a. NNs). This analysis is important in order to establish a set of trained, verifiable, accurate and efficient NN kernels with expected performance guarantees. Additionally, it would validate the validity of the approach, paving the way for more advance NN models that are trained in more challenging scenarios using a combination of synthetic and real training data.

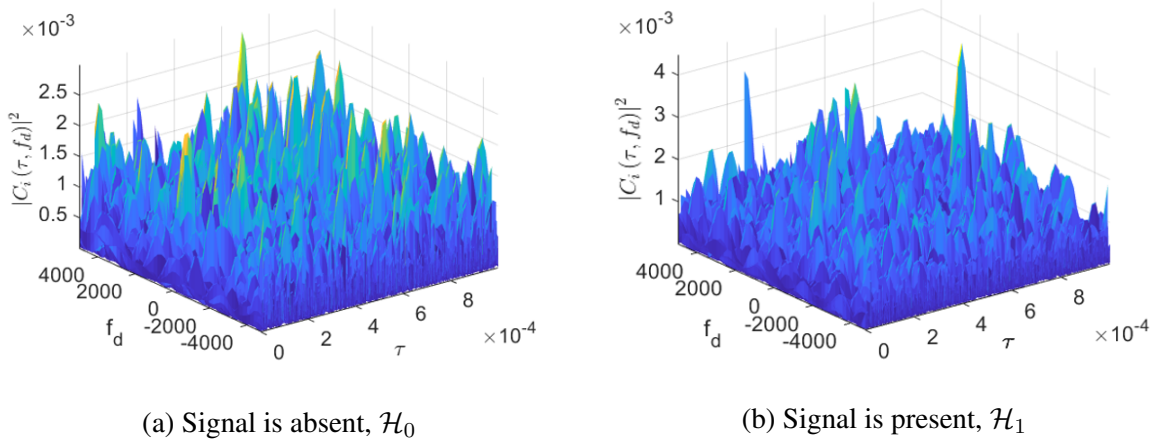


Figure 2.1: CAF evaluation at the delay/Doppler grid in the (a) absence and (b) presence of a signal with $C/N_0 = 39$ dB-Hz.

2.1 Deep learning of GNSS signal acquisition

2.1.1 GNSS Signal Models

A receiver observes signals from M satellites plus noise. After down conversion and sampling (at a rate $f_s = 1/T_s$) the samples discrete-time signal is:

$$\begin{aligned}
 y[n] &= \sum_{i=1}^M x_i[n; \boldsymbol{\theta}_i] + \eta[n] \\
 x_i[n; \boldsymbol{\theta}_i] &= \alpha_i b_i(nT_s - \tau_i) c_i(nT_s - \tau_i) e^{j2\pi f_{d,i} n T_s + j\phi_i}
 \end{aligned} \tag{2.1}$$

with α_i the amplitude of the i -th received signal; $b_i(\cdot)$ the data bits of the i -th navigation message; $c_i(\cdot)$ the spreading code of the i -th satellite; τ_i the time-evolving delay of the i -th satellite; $f_{d,i}$ the Doppler-shift; ϕ_i a carrier-phase term introduced by the channel; and $\eta[n]$ models the random noise at the receiver, typically complex, zero-mean and Gaussian distributed with variance σ^2 . For the sake of clarity, the signal parameters for the i -th satellite are gathered in a vector $\boldsymbol{\theta}_i = (\alpha_i, \phi_i, \tau_i, f_{d,i})^\top$.

Coherent and non-coherent integrations can be employed to reduce the noise impact. Therefore, when searching for the i -th satellite, this problem can be formulated as a hypothesis testing problem with two possibilities, which are shown in Fig. 2.1:

$$\begin{aligned}
 \mathcal{H}_0 &: i\text{-th satellite is not present} \\
 \mathcal{H}_1 &: i\text{-th satellite is present}
 \end{aligned}$$

Equivalently, the two competing hypothesis are:

$$\begin{aligned}\mathcal{H}_0 & : y[n] = \eta[n] \\ \mathcal{H}_1 & : y[n] = x_i[n; \boldsymbol{\theta}_i] + \eta[n]\end{aligned}\tag{2.2}$$

such that $n = 0, \dots, N - 1$ index the N samples used in acquisition (i.e., coherent integration interval). Since the parameters in $\boldsymbol{\theta}_i$ are unknown, the optimal detection framework (in the ML sense) is the Generalized Likelihood Ratio Test (GLRT), which requires Maximum Likelihood Estimation (MLE) of the vector $\boldsymbol{\theta}_i$. Given a set of N observations, $\mathbf{y} = (y[0], y[1], \dots, y[N - 1])^\top$ the MLE of $\boldsymbol{\theta}_i$ is defined as

$$\hat{\boldsymbol{\theta}}_i = \arg \max_{\boldsymbol{\theta}_i} p(\mathbf{y} | \boldsymbol{\theta}_i), \tag{2.3}$$

where it is typically assumed that the parameters in $\boldsymbol{\theta}_i$ are piece-wise constant within the N samples of \mathbf{y} and that the codes have ideal cross-correlation properties, so they can be processed independently at the receiver.

It can be seen that the GLRT results in the maximization of the correlation between the received signal and a locally generated code. This correlation operation is encoded in the so-called Cross Ambiguity Function (CAF), which is nothing but the correlation between $y[n]$ and the spreading code of the i -th satellite, at a given delay/Doppler pair (in discrete-time):

$$C_i(\tau, f_d) = \frac{1}{N} \sum_{n=0}^{N-1} y[n] \underbrace{c_i(nT_s - \tau) \exp\{-j2\pi f_{d,i} nT_s\}}_{\text{Local replica}}, \tag{2.4}$$

which can be expressed more compactly in vector notation after gathering N samples from the samples and the local code as $\mathbf{y}, \mathbf{c}_i \in \mathbb{C}^{N \times 1}$ as:

$$C_i(\tau, f_d) = \frac{\mathbf{c}_i^H \mathbf{y}}{N}. \tag{2.5}$$

The CAF is crucial in the acquisition (and tracking) of the satellites' signals. The MLE of $\boldsymbol{\theta}_i$ can be expressed in terms of it as:

$$(\hat{\tau}_i, \hat{f}_{d,i}) = \arg \max_{\tau, f_d} \left\{ |C_i(\tau, f_d)|^2 \right\} \tag{2.6}$$

$$\hat{\alpha}_i = \left| C_i(\hat{\tau}_i, \hat{f}_{d,i}) \right| \tag{2.7}$$

$$\hat{\phi}_i = \angle C_i(\hat{\tau}_i, \hat{f}_{d,i}), \tag{2.8}$$

and we decide that the i -th satellite is present by setting a detection threshold β (designed for a desired false alarm probability) on the test statistic in the optimization problem in (2.6) such as:

$$|\mathcal{C}_i(\tau, f_d)|^2 \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\geq}} \beta. \quad (2.9)$$

2.1.2 CAF Evaluation

The first stage of the acquisition block consists of the evaluation of the CAF in (2.4). More in detail, the received signal $y[n]$ is multiplied by two orthogonal sinusoids at the frequency $F_D = (f_{IF} + f_d)T_s$. In this way, two new signals are generated:

$$\begin{aligned} Y_c(n, F_D) &= y[n] \cos(2\pi F_D n) \\ Y_s(n, F_D) &= -y[n] \sin(2\pi F_D n) \end{aligned} \quad (2.10)$$

The multiplication by these two orthogonal sinusoids is aimed at translating to baseband the received signal, removing the effect of the Doppler shift. These multiplications correspond to the complex modulation in (2.4) that is implemented in GNSS receivers by splitting the incoming signal into two branches and separately multiplying them by cosine and sine. The normalized frequency:

$$F_D = (f_{IF} + f_d)T_s = \frac{f_{IF} + f_d}{f_s} \quad (2.11)$$

is given by two terms:

- the intermediate frequency, f_{IF} ,
- the local Doppler frequency f_d .

The intermediate frequency f_{IF} is known and depends on the receiver architecture[9], whereas f_d is chosen from a finite set of the type:

$$f_d = f_{d,min} + l\Delta f \quad \text{for } l = 0, 1, \dots, L - 1 \quad (2.12)$$

Different Doppler frequencies are tested to determine the Doppler shift of the incoming signal. For low dynamic applications, $-5KHz \leq f_d \leq 5KHz$. The Doppler step Δf and its normalized counterpart $\Delta F = \Delta f / f_s$ are chosen in order not to exceed a maximum loss due to Doppler residual errors. The signals $Y_c(n, F_D)$ and $Y_s(n, F_D)$ are then multiplied by a local signal

CHAPTER 2. GNSS SIGNAL ACQUISITION IN THE ABSENCE OF SPOOFER

replica that reproduces the primary Pseudo-Random Noises (PRN) code. The local signal replica is delayed by τ and signals are obtained:

$$\begin{aligned} Y_c(n, \tau, F_D) &= y[n] \cos(2\pi F_D n) c[n - \tau] \\ Y_s(n, \tau, F_D) &= -y[n] \sin(2\pi F_D n) c[n - \tau] \end{aligned} \quad (2.13)$$

The delay τ is taken from a set:

$$\tau = \tau_m i n + h \Delta \tau \quad \text{for } h = 0, 1, \dots, H - 1 \quad (2.14)$$

By testing the different delays, the acquisition block is able to estimate the delay of the received signal $y[n]$.

The signals $Y_c(n, \tau, F_D)$ and $Y_s(n, \tau, F_D)$ are then integrated, leading to the in-phase and quadrature components $Y_I(\tau, F_D)$ and $Y_Q(\tau, F_D)$:

$$\begin{aligned} Y_I(\tau, F_D) &= \frac{1}{N} \sum_{n=0}^{N-1} Y_c(n, \tau, F_D) \\ Y_Q(\tau, F_D) &= \frac{1}{N} \sum_{n=0}^{N-1} Y_s(n, \tau, F_D) \end{aligned} \quad (2.15)$$

In(2.15), N represents the number of samples used for evaluating the in-phase and quadrature components and is used to define the coherent integration time:

$$T_c = N T_s \quad (2.16)$$

The two component of (2.15) represent the real and the imaginary parts of the CAF that is finally given by:

$$Y(\tau, F_D) = Y_I(\tau, F_D) + j Y_Q(\tau, F_D) \quad (2.17)$$

and that correspond to Eq. (2.6).

The CAF is a function that depends on the delay τ and the Doppler frequency f_d of the local replica. The optimization in (2.6) is performed over a grid of possible τ and f_d values, typically evaluating the CAF on a set of discrete values. Such a bi-dimensional grid is referred to as the search space. The search space consists of a set of cells that include the different values of delay and

Doppler, which we gather in vectors $\boldsymbol{\tau} \in \mathbb{R}^{n_\tau}$ and $\mathbf{f}_d \in \mathbb{R}^{n_f}$, respectively. Typically, we have that $n_\tau > n_f$. The evaluation of this grid can be performed following several strategies that trade-off search speed and performance. The next step after calculating the CAF with acquisition algorithms is a time to use one of the searching strategies to explore the search space more quickly. Three searching strategies are typically considered: maximum search, serial search, and hybrid search strategies [6].

2.1.3 Searching Strategies

In this section, three searching strategies are explained to choose the best one for searching space more quickly and efficiently.

1. Maximum: this strategy evaluates the CAF all over the search space $\mathbb{R}^{n_\tau} \times \mathbb{R}^{n_f}$, such that each cell corresponds to a CAF value at the corresponding delay/Doppler pair. The overall maximum value of the ambiguity function is then selected and compared to the threshold β , if the maximum value is greater than β the satellite is considered acquired, with estimated code delay and Doppler frequency corresponding to those of the maximum's cell.
2. Serial: in this strategy, the ambiguity function is evaluating serially cell by cell. In each cell, when the ambiguity function (2.9) is computed, it is immediately compared to the threshold. If the value exceeds the threshold, the acquisition process stops and the value of the estimated code delay and Doppler frequency is matched to those from the cell under the test. This strategy has the benefit of reducing the number of CAF evaluations, at the expense of some performance degradation.
3. Hybrid: this strategy evaluates the ambiguity function row-by-row (or column-by-column), and at the end of each row (column) the values of the computed ambiguity functions are compared to the threshold. As soon as the maximum value in the current row (column) exceeds the threshold, the acquisition process stops and the estimated code delay and Doppler frequency are set to the corresponding cell. This strategy brings a balance between the two approaches above.

In this work, the maximum strategy is used as a searching strategy. This allows us to have a full representation of the CAF in (2.9), which is then fed to NNs for classification between \mathcal{H}_0 and \mathcal{H}_1 hypotheses, as detailed in Chapter 2.1.5.

2.1.4 Benchmark performance using the Receiver Operating Characteristic function

The so-called Receiver Operating Characteristic ROC is a popular metric to assess the performance of any detector/classifier. A ROC is a plot of the detection probability (P_d) as a function of the false alarm probability (P_{fa}). More precisely, $P_d = \mathbb{P}\{|C_i(\tau, f_d)|^2 > \beta | \mathcal{H}_1\}$ is the probability of correctly detecting a GNSS signal given that it was present, while $P_{fa} = \mathbb{P}\{|C_i(\tau, f_d)|^2 > \beta | \mathcal{H}_0\}$ is the probability of detecting the signal given that it should have not being detected. Ideally, the aim is to have the classifier operate such that $P_d \rightarrow 1$ and $P_{fa} \rightarrow 0$.

Theoretical ROC curves are well known for GNSS signal acquisition [6] and are used to benchmark different algorithmic solutions. The remainder of this section provides a quick summary of the theoretical ROC used in coherent/non-coherent integration schemes. In this article, we use this theoretical ROC to assess the performance of our DNN-based solution against the best achievable performance under the standard (i.e., non-data driven) method.

In order to calculate the ROC curves, first one needs to calculate the P_{fa} and P_d probabilities. The value of the detection threshold β is typically computed for a given false alarm probability, given by

$$P_{fa,K}(\beta) = \exp\left(-\frac{\beta}{2\sigma_n^2}\right) \sum_{k=0}^{K-1} \frac{1}{k!} \left(\frac{\beta}{2\sigma_n^2}\right)^k \quad (2.18)$$

where K indicates the number of non-coherent integrations (i.e. averages of K coherent integrations as in (2.9)) considered (such that $K = 1$ in absence of non-coherent integration) and $\sigma_n^2 = \frac{\sigma^2}{2N}$ is the variance of the in-phase and quadrature outputs.

Then, the P_d can be calculated as a function of β as

$$P_{d,K}(\beta) = Q_K\left(\sqrt{K\frac{\lambda}{\sigma_n^2}}, \sqrt{\frac{\beta}{\sigma_n^2}}\right) \quad (2.19)$$

where $\lambda = \alpha_i^2/4$ is the non-centrality parameter, and the generalized Marcum Q -function is defined as

$$Q_K(a, b) = \frac{1}{a^{K-1}} \int_b^{+\infty} x^K \exp\left(-\frac{a^2 + x^2}{2}\right) I_{K-1}(ax) dx, \quad (2.20)$$

which allows for computation of the ROC curves.

2.1.5 Deep Learning Approach

In this work, the goal is to create a neural network model that is capable of recognizing the presence/absence of satellite signal. To that aim, we use as inputs the CAF evaluated at the delay/Doppler grid, which can be considered as an image. Such images (refer to Fig.2.1 for an exemplary situation) has certain characteristics that can be used to determine whether the signal is present or not: *i*) in the absence of signal from a specific satellite, the image should be composed of random values (theory telling that the CAF would be exponentially distributed in that case); and *ii*) in the presence of a satellite a peak should emerge from the random noise floor. This knowledge can be used to train a data-driven model (e.g. a neural network of some sort) such that a classifier can be used which learns to discriminate between \mathcal{H}_0 and \mathcal{H}_1 , the hypotheses described earlier in Section 2.1.1.

In this work, two different structures of artificial neural networks are considered and compared: *i*) a MLP, which is a neural network architecture with moderate complexity that has been widely used in the machine learning literature; and *ii*) a Convolution Neural Network (CNN), very popular within the computer vision community thanks to its ability to capture complex non-linear phenomena, at the expenses of a much larger complexity compared to MLPs. These models are discussed in this section after a brief overview on how the classifier is built following a probabilistic approach, in which the different NNs are in charge of delivering Bayesian estimates of the probabilities of each hypothesis given the observed data. Recall that the data fed to the NNs is the CAF's delay/Doppler map for the i -th satellite, which we denote with \mathbf{Z}_i in the sequel. The proposed methodology works on a per-satellite basis. That is, the $\{m, n\}$ element of the input matrix is defined as

$$[\mathbf{Z}_i]_{m,n} = |C_i([\boldsymbol{\tau}]_m, [\mathbf{f}_d]_n)|^2, \quad (2.21)$$

where $\boldsymbol{\tau}$ and \mathbf{f}_d are vectors containing the tested delay and Doppler-shifts, respectively. We use the convention that $[\mathbf{a}]_m$ represents the m -th element in the vector, \mathbf{a} and that $[\mathbf{A}]_{m,n}$ provides a shortcut for the element of \mathbf{A} in the m -th row and n -th column.

In the Bayesian sense, the information of the models is gathered in their *a posteriori* distribution after observing the data. An optimal (Bayesian) test between \mathcal{H}_0 and \mathcal{H}_1 is given by the ratio,

$$\frac{p(\mathcal{H}_1|\mathbf{Z}_i)}{p(\mathcal{H}_0|\mathbf{Z}_i)} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\geq}} 1, \quad (2.22)$$

in which case we basically favor the model with largest a posteriori probability. This can be further

expanded in terms of the likelihood and a priori distributions, as

$$\frac{p(\mathbf{Z}_i|\mathcal{H}_1) \mathbb{P}(\mathcal{H}_1)}{p(\mathbf{Z}_i|\mathcal{H}_0) \mathbb{P}(\mathcal{H}_0)} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} 1, \quad (2.23)$$

where we readily identify that $\mathbb{P}(\mathcal{H}_i)$ denotes the a priori probability of the i -th hypothesis. In the absence of better priors, we may assume equally likely hypotheses $\mathbb{P}(\mathcal{H}_0) = \mathbb{P}(\mathcal{H}_1) = 1/2$. Otherwise, we might incorporate that information in the hypothesis test, resulting in the adjustment of a threshold γ . The resulting test statistic is such that

$$\mathcal{T}(\mathbf{Z}_i) \triangleq \frac{p(\mathbf{Z}_i|\mathcal{H}_1)}{p(\mathbf{Z}_i|\mathcal{H}_0)} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \gamma, \quad (2.24)$$

which would substitute the standard acquisition test defined in (2.9). Since the test statistic is a ratio of probabilities, we have that $0 < \mathcal{T}(\mathbf{Z}_i) < \infty$.

The trained NNs (explained below) are then providing the probabilities of each of the hypotheses in (2.24). Therefore, the input data would be \mathbf{Z}_i and the output of the NN would be the estimated probability for the i -th satellite to be absent or present in the dataset \mathbf{y} used to build \mathbf{Z}_i .

If the test in (2.24) is in favor of \mathcal{H}_1 , then an estimate of the delay/Doppler for the i -th satellite is given by the arguments of the largest element in \mathbf{Z}_i . That is,

$$\{\hat{m}, \hat{n}\} = \arg \max_{m,n} [\mathbf{Z}_i]_{m,n} \quad (2.25)$$

such that $\hat{\tau}_i = [\boldsymbol{\tau}]_{\hat{m}}$ and $\hat{f}_{d,i} = [\mathbf{f}_d]_{\hat{n}}$.

NNs are models composed of neurons, which are information processing units, for complex data processing. A NN typically contains an input layer, one or more hidden layers, and an output layer, as well as pre-defined activation functions that connect adjacent layers. Each layer has a specific weight, which is usually determined with backpropagation during a training process that involves large amounts of data with known labels [52, 53].

Remarkably, there are implementations of DNNs that are extremely efficient, allowing for fast, real-time execution of the DNN classifier once the network is trained. The main challenge for DNN being to have enough training data to characterize those effects in a data-driven manner, an issue discussed in Section 2.1.6.

2.1.5.1 MLP

The first neural network structure that is used in this research is the so-called Multilayer Perceptron (MLP), which is referred to as a traditional neural network. This type of network is

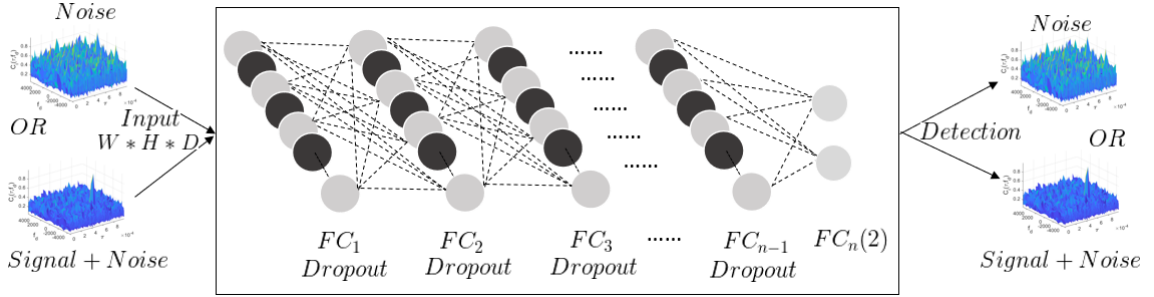


Figure 2.2: Detection scheme of GNSS acquisition by using the Fully Connected Structure of Deep Learning approach.

composed of one or more layers of neurons (which consist of a row of neurons). Fig. 2.2 shows an exemplary representation of this type of network. The first layer is called the input layer, which is fed with the training dataset for learning the parameters of the (potentially several) hidden layers that are not directly exposed to the input. During training, a number of nodes in the hidden layer are randomly ignored or “dropped out”, which are shown in black in Fig. 2.2. This process will temporarily remove them from the network with a given probability for all incoming and outgoing connections. The last layer of the MLP is called output layer, follow up by a *softmax* layer. The number of neurons in this last layer depends on the number of the classes that one wants to classify, since those provide their probabilities. The softmax layer calculates probabilities for each possible class, given a dataset. Those probabilities must add up to 1, and are used as in (2.24) in order to determine which class is the most probable given an input set Z_i .

2.1.5.2 CNN

The second artificial neural network structure considered here is the so-called Convolutional Neural Network (CNN), which is one of the most popular models for deep learning in the context of learning class labels from image datasets.

A CNN can have tens or hundreds of layers, where each of these layers learn to identify different features of an image [54, 55]. At each layer, filters are applied to each training image and the output of each convolution image is used as an input to the next layer.

Fig. 2.3 illustrates a CNN structure. In contrast to other neural networks such as MLP, CNN is composed of an input layer, many hidden layers, and an output layer. During training, the input size of the CNN is fixed, the input is going through a stack of convolutional layers with the same or different filter sizes. In each convolution layer, the filter sweeps the input image from left to

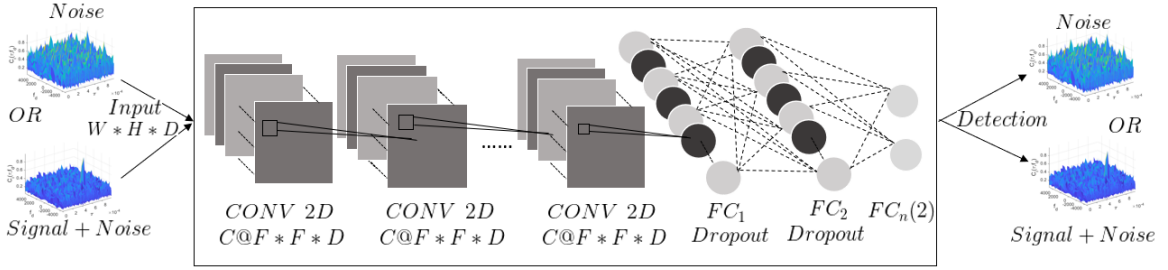


Figure 2.3: Detection scheme of GNSS acquisition by using the Convolutional Neural Network Structure of Deep Learning approach.

right and up to down by using a stride with 2 pixels size, which is the number of pixels that each time the filter shifts. In the end, the convolution layers are followed by Fully Connected (FC) layers and a final *softmax* layer, which is used for classification purposes [55].

The CNN structure is shown in the box of Fig. 2.3 which has several convolution and fully connected layers. Each convolution layer consists of a number of filters (C), with filter size (F) and channel size (D). It transforms the input images with dimensions of $W_1H_1D_1$ through a set of convolution filters, each of these filters activates certain features from the images and creates an output with dimensions of $W_2H_2D_2$ as an input to the next layer.

After each layer, the batch normalization will be used to speed up learning and using the activation function to make input as a non-linearity output. The number of convolution layers are depended on the structure that is using. After the last convolution layer, the CNN architecture highlights fully connected layers in charge of a classification task, and the output of these layers is a vector with dimensions of the number of classes (2 in this case) that will be predicted. The output would be the predicted probabilities for each class, as needed to compute (2.24).

2.1.6 Simulation Setup and DNN training

The goal of this work is framed within the described satellite signal acquisition process on a GNSS receiver. In particular, we propose to use a DNN to carry out the detection (or classification) process. The results of this work are shown in the next section, which are established from simulating a synthetics dataset with the two types of NNs described earlier, namely MLPs and CNNs.

The dataset that is used consists of 10^4 snapshots of, GPS L1 C/A, I&Q samples with different Carrie-to-Noise-density ratio (C/N_0) varying between 30 to 45 dB-Hz, as well as randomly generated delays between 0 to 1 ms and Doppler shifts between -4000 to 4000 Hz. These samples are then processed to compute the CAF over a Doppler-delay grid. An analogy to images can be

CHAPTER 2. GNSS SIGNAL ACQUISITION IN THE ABSENCE OF SPOOFER

made for these, CAFs where each Doppler/delay cell is a pixel whose value is that of the CAF, \mathbf{Z}_i . For instance, if 20 Doppler bins are considered to acquire a GPS L1 C/A signal, those images would be 20×1023 dimensional. These images are fed to the input layer of the DNN, whose output would be the classification between presence/absence of the i -th satellite signal, as well as its Doppler and delay values if present. In a supervised training scheme, these input/output pairs are provided by labeled data by using the aforementioned synthetic data generation.

The proposed method is based on NNs, which is using a snapshot containing either signal-plus-noise or noise-only as ground truth and use the classifier network with softmax layer and dropout to classify the snapshot and predict the probability of belonging to each class.

The model implemented with MLP structures considered different number of fully connected layers. Each fully connected layer followed up with Rectified Linear Unit (ReLU) activation function to allows for faster and more effective training by mapping negative value to zero [56]. After each layer, a $1/2$ dropout probability was considered. The last fully connected layer contained two neurons, used in predicting the class for which the input image \mathbf{Z}_i belongs to.

After defining the network structure, the training options were specified. The network trained with Stochastic Gradient Descent with Momentum (SGDM) optimizer with an initial learning rate of 0.001. The maximum number of epochs, which is a full training cycle on the entire training dataset, was set to 30 and at every epoch the data was shuffled. After 20 epochs, the learning rate dropped by a factor of 0.1. the training progress shows the mini-batch loss and accuracy and the validation loss and accuracy. The loss is the cross-entropy loss, and the accuracy was defined as the percentage of inputs that the network classified correctly.

The CNN structure that was used is very similar to the VGG 16 structure [54]. The network has 13 convolution layers and 3 fully connected layers. Each convolution layer was followed by a batch normalization layer and a ReLU activation function. The batch normalization layers, normalizing the activation and gradients propagation through a network and using it between the convolution layer and ReLU layers to speed up network training [57]. Each fully connected layer follows up with a ReLU activation function and a dropout layer with the probability of 0.5. The last fully connected layer contains two neurons to predict each image belongs to which class, since two type of classes are existing in this work. After defining the network structure, the training options were specified, which were the same as for the MLP training options.

CHAPTER 2. GNSS SIGNAL ACQUISITION IN THE ABSENCE OF SPOOFER

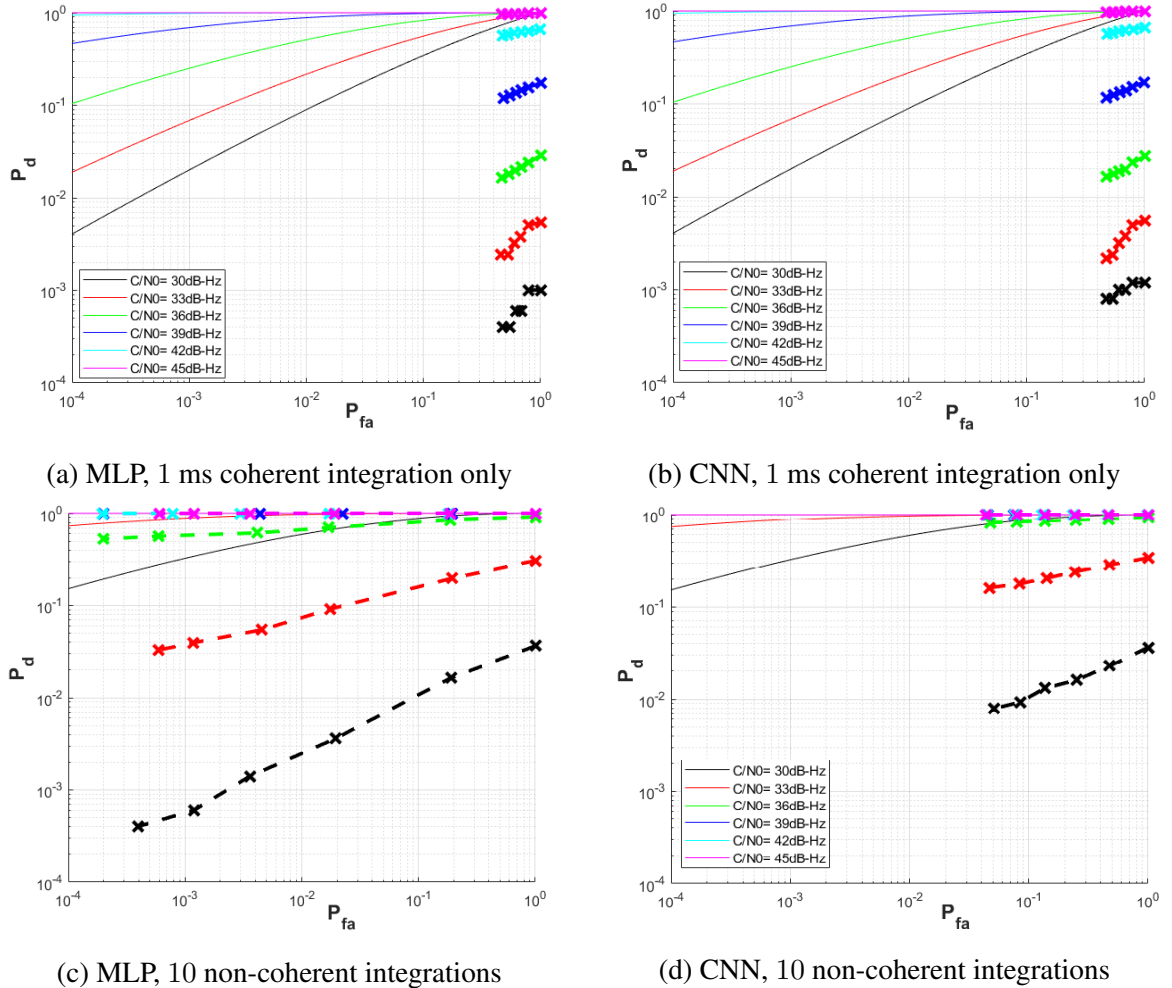


Figure 2.4: ROC curves for MLP and CNN using CAF generated with (a,b) 1 ms coherent integration and (c,d) 1 ms coherent integration and 10 non-coherent integrations.

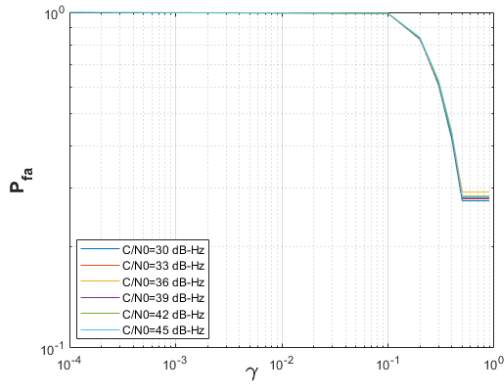
2.1.7 Result

The machine learning method was implemented using the two described types of NNs. In training, both were trained using either coherent integration data (i.e., 1 ms coherent) or non-coherent integrations (i.e., 1 ms coherent and 10 ms non-coherent).

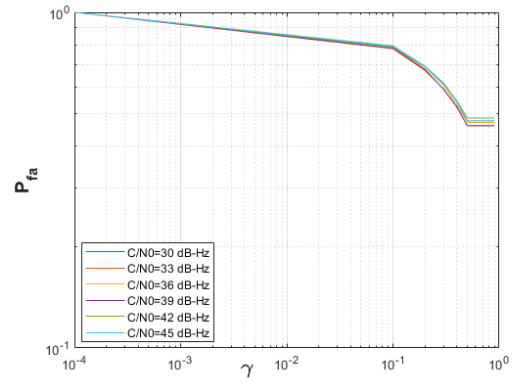
Fig. 2.4 to 2.6 shows the obtained result for both MLP and CNNs under the two integration configurations. Notice that in this case, they are trained and tested under the same integration configurations.

From Fig. 2.4, it can be observed that for low C/N_0 values, the performance of NNs is not attaining the theoretical ROC curves. However, as C/N_0 increases, such an approach is able to

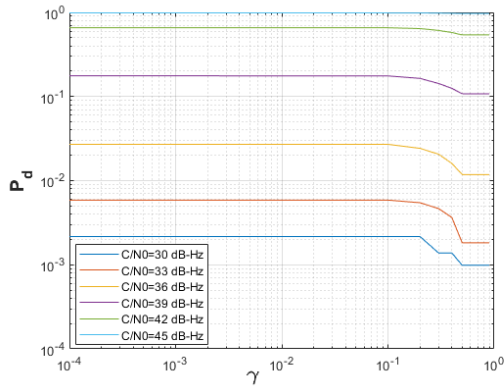
CHAPTER 2. GNSS SIGNAL ACQUISITION IN THE ABSENCE OF SPOOFER



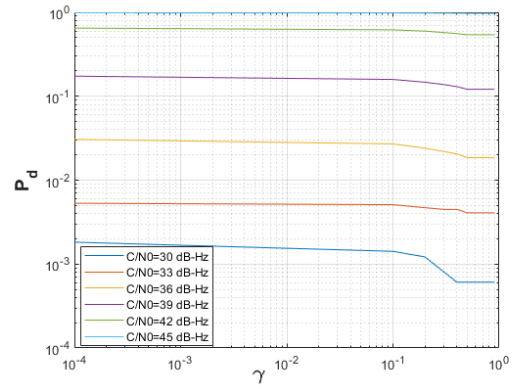
(a) MLP



(b) CNN



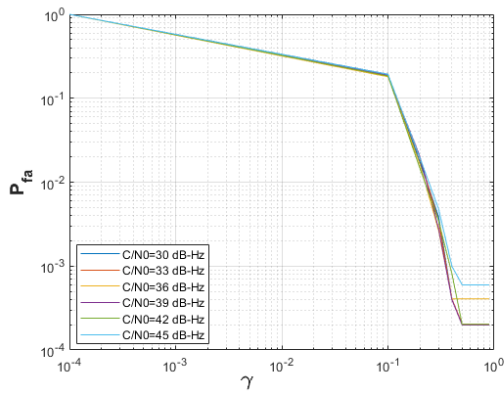
(c) MLP



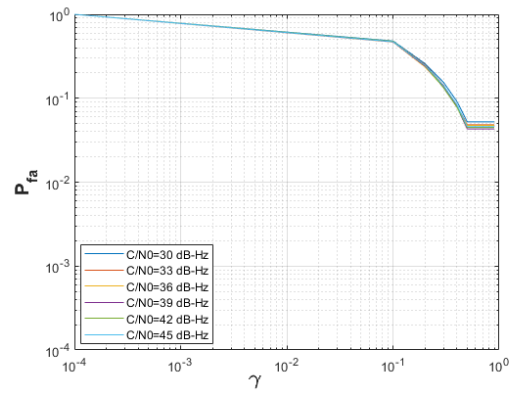
(d) CNN

Figure 2.5: $P_d(\gamma)$ and $P_{fa}(\gamma)$ under 1 ms coherent integration for MLP and CNN models.

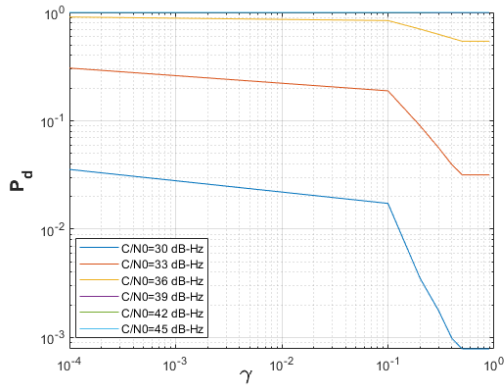
CHAPTER 2. GNSS SIGNAL ACQUISITION IN THE ABSENCE OF SPOOFER



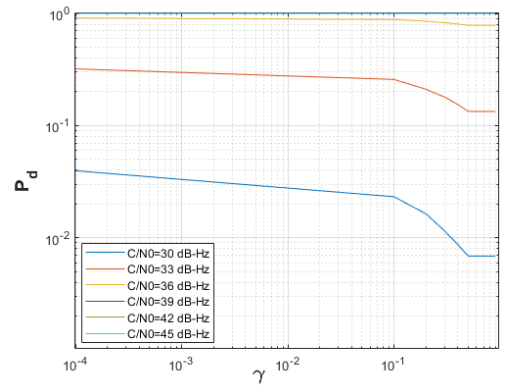
(a) MLP



(b) CNN



(c) MLP



(d) CNN

Figure 2.6: $P_d(\gamma)$ and $P_{fa}(\gamma)$ under 1 ms coherent and 10 non-coherent integrations for MLP and CNN models.

CHAPTER 2. GNSS SIGNAL ACQUISITION IN THE ABSENCE OF SPOOFER

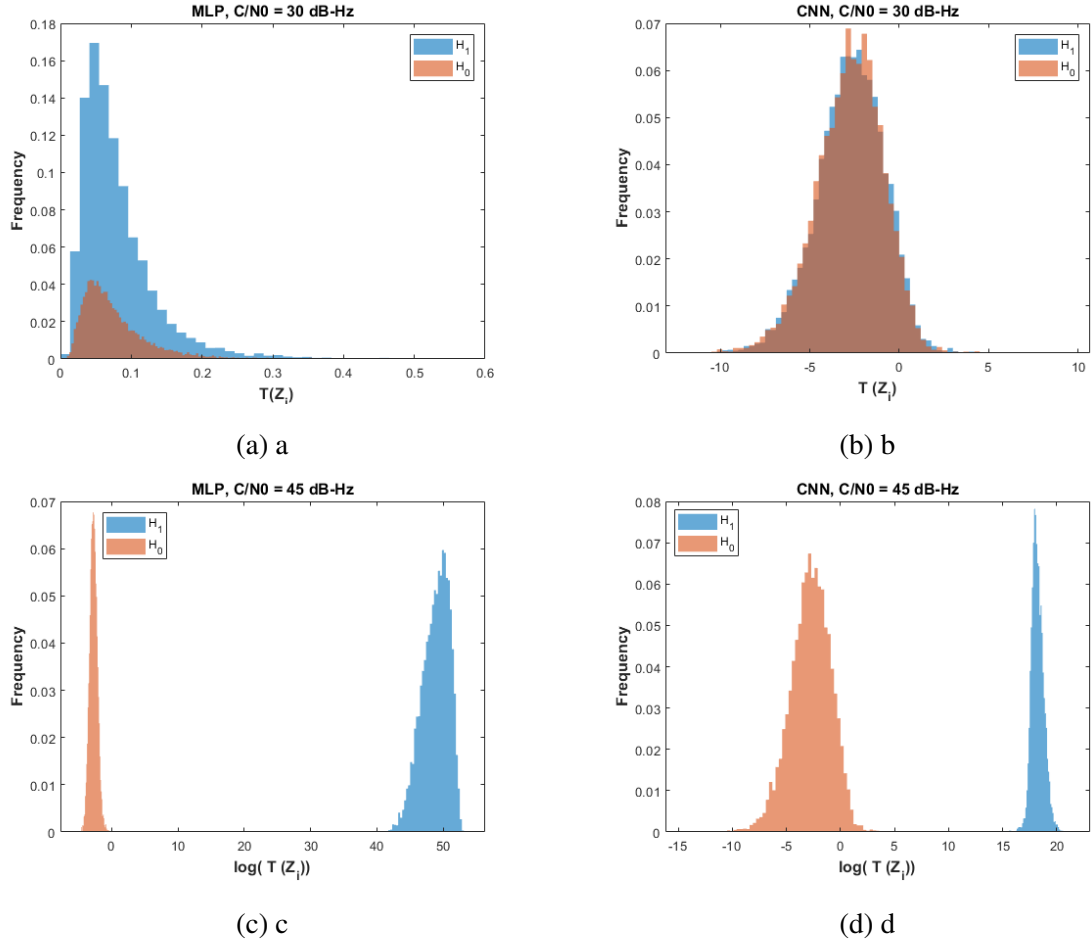


Figure 2.7: Histogram of $\mathcal{T}(\mathbf{Z}_i)$ under \mathcal{H}_0 and \mathcal{H}_1 hypotheses for each NN and two different C/N_0 values.

reach theoretical limits. An explanation could be that for low, C/N_0 the various NNs cannot extract the relevant features from the *image*, \mathbf{Z}_i . When C/N_0 , either because of an actual power increase or longer integration times, the NN is able to perform classification on whether the signal from the i -th satellite is present or not.

As for a comparison between MLP and CNN architectures, although their performance is very similar as per Fig. 2.4, the results show that MLP slightly outperforms CNN classification results. These considerations are supported by the results in Fig. 2.5, where detection and false alarm probabilities are plotted against the classification threshold (i.e. $P_d(\gamma)$ and $P_{fa}(\gamma)$), for different value of C/N_0 under 1 ms coherent integration. Likewise in Fig. 2.6, where non-coherent values are shown. In essence, those probability plots show that CNN achieves larger P_{fa} values than MLP,

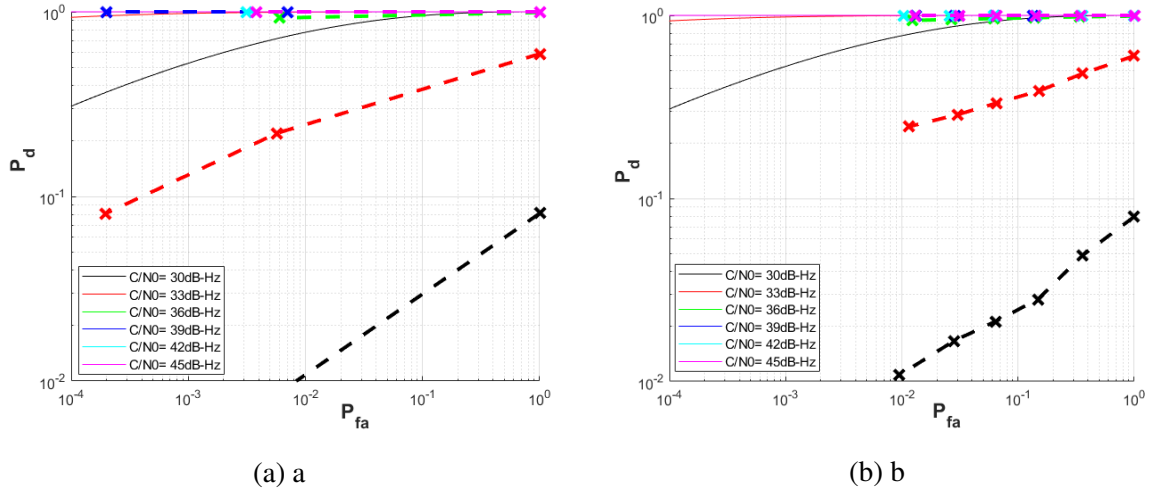


Figure 2.8: ROC curves for MLP and CNN using different configurations to generate CAF inputs for training and testing.

which causes a degradation in their ROC curves. The impact of low C/N_0 on ROC performance is further explained by the histograms of the test in (2.24) under both hypotheses, shown in Fig 2.7. Recall that one would like to have the histograms under \mathcal{H}_0 and \mathcal{H}_1 as *separate* as possible, which happens for large C/N_0 but clearly does not for low C/N_0 values. Whereas in the former the empirical distributions can be clearly distinguished (notice these where plot for $\log \mathcal{T}(\mathbf{Z}_i)$ to be visible), the former are overlapped such that a sample from $\mathcal{T}(\mathbf{Z}_i)$ cannot be meaningfully discerned among both distributions.

In the last set of results, we aim at understanding the generalization properties of these models under different receiver configurations, namely the integration intervals. The ROC curves for both MLP and CNN models for such experiment are shown in Fig. 2.8. For both models, during training the network is using CAF generated with 10 ms non-coherent integrations and then for the testing part it is using the CAF generated with 15 non-coherent integration time. These results should be compared to those reported in Fig. 2.4. At the light of the plots, it seems that MLP has less generalization properties than CNN. It has shown that in both MLP and CNN models the neural network’s performance increase by increasing the number of the integration time even if the network is trained with less integration interval when compared to Fig. 2.4(c), 2.4(d). Further research is needed in understanding how learning from a configuration (e.g., coherent integration) can be transferred to other receiver configurations (e.g., non-coherent integrations).

2.2 Parallelizable deep learning acquisition by CAF splitting and data fusion

The versatility and computational affordability of the proposed method are addressed by splitting the CAF into smaller overlapping sections, which are fed to a bank of parallel classifiers whose probabilistic results are optimally fused to provide a so-called probability ratio map from which acquisition is decided. This method is very helpful to increase the accuracy when the size of the dataset is too large. Additionally, this chapter shows how non-coherent integration schemes are enabled through optimal data fusion, with the goal of increasing the resulting classifier accuracy. At the end of this chapter, it provides simulation results showing that the proposed data-driven method outperforms current CAF maximization strategies, enabling enhanced acquisition at lower carrier-to-noise density ratios.

2.2.1 Sub-Image Model structure

CNNs are one of the most popular models for deep learning, with demonstrated performance in label classification in the context of image datasets. A CNN can have tens or hundreds of layers, where each of these layers learn to identify different features of an image [54, 55]. At each layer, a cascade of filters is applied to input images, whose parameters were previously learned from pairs of known input/output images. The output of each layer is used as an input to the next layer sequentially. Fig. 2.9 illustrates a CNN structure, as employed in this work. In contrast to other neural networks such as MLPs, CNNs is composed of an input convolutional layer (whereby the image is filtered through convolution with filters learned from the data), several fully-connected hidden layers, and an output layer. During training, the input size of the CNN is fixed, the input is going through a stack of convolutional layers with the same or different filter sizes. In each convolution layer, the filter sweeps the input image from left to right and up to down by using a stride with 2 pixels size, which is the number of pixels that each time the filter shifts. In the end, the convolution layers are followed by FC layers and a final *softmax* layer, which is used for classification purposes and produces the desired class probabilities [55].

The CNN structure is shown in the central box of Fig. 2.9 which features several convolution and fully connected layers. Each convolution layer consists of a number of filters (C), with filter size (F) and channel size (D). The ℓ -th convolutional layer transforms its input images from the previous layer with dimensions of $W_{\ell-1} \times H_{\ell-1} \times D_{\ell-1}$ through a set of convolution filters, each

CHAPTER 2. GNSS SIGNAL ACQUISITION IN THE ABSENCE OF SPOOFER

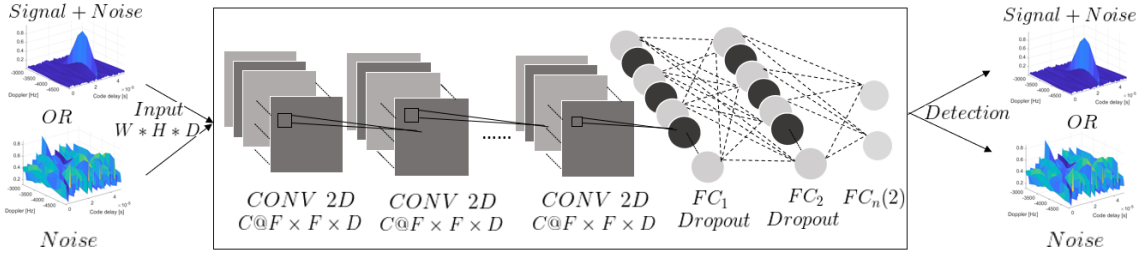


Figure 2.9: Classification of signal (\mathcal{H}_1) or noise (\mathcal{H}_0) in CAFs as part of the proposed GNSS signal acquisition scheme. Particularly, a set of convolutional layers followed by Fully-Connected layers provide the capabilities of deep learning from large datasets.

of these filters activates certain features from the images and creates an output with dimensions of $W_\ell \times H_\ell \times D_\ell$ as an input to the next layer. Notice that initial dimensions are such that $W_0 = n_\tau$ and $H_0 = n_f$, whereas $D_\ell = 1, \forall \ell$ since the data are matrices.

After each layer, a batch normalization is used to speed up learning, and an activation function employed before generating the layer output. The number of convolution layers depends on the structure that is used, where the tradeoff is between complexity (reduced number of convolutional layers) and performance (large number of layers). After the last convolution layer, the CNN architecture has a set of fully connected layers in charge of the classification task, and the output of these last layer has the dimensions of the number of classes (two in the case of this article where a binary test is solved in (2.2)) that will be predicted. The output would be the predicted probabilities for each class, as required to compute the test in (2.24).

The main objectives of this work are to classify the absence/presence of satellite signals in CAF maps, as well as to accurately estimate its delay/Doppler parameters in case of its positive detection. To achieve the latter, a CAF map is computed in a dense delay/Doppler grid – as it is common for standard acquisition schemes – which is then fed to the NN model in charge of producing the posterior class probabilities. As a consequence, the input matrix size can be potentially large (i.e., $n_\tau n_f$) which not only might pose a computational complexity limit, but also requires all input matrices to be of the same dimension. In order to alleviate this issue, a sliding scheme is proposed in this work, whereby the large input CAF matrix is scanned using lower dimensional images as the input to the NN classifier.

More precisely, the input dataset image is split into several sub-images, each corresponding to a test delay/Doppler value. The objective being to reduce the initial dimensions W_0 and H_0 , such that the processing is computationally affordable and parallelizable. These sub-images are separately

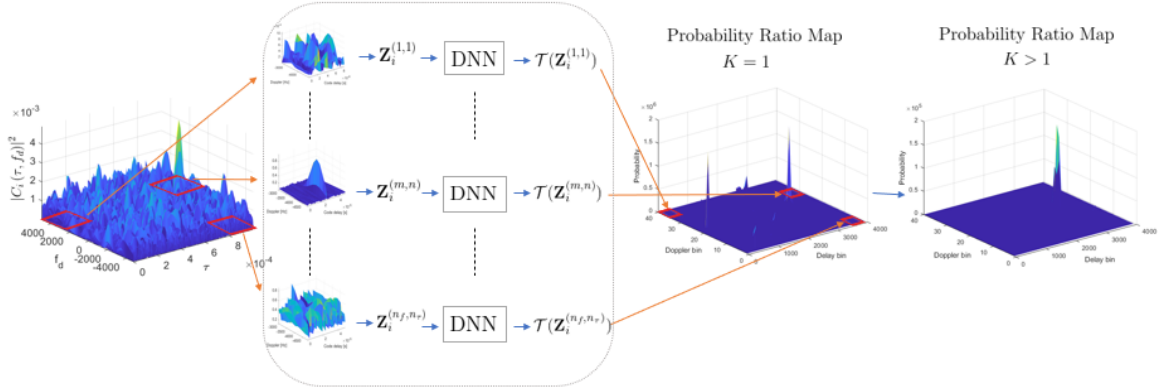


Figure 2.10: Proposed acquisition method where the CAF map \mathbf{Z}_i is split into smaller sub-images $\mathbf{Z}_i^{(m,n)}$, which are fed to a bank of parallel DNN binary classifiers to produce probability ratio maps. To increase accuracy, several ($K > 1$) probability ratio maps can be non-coherently fused, as shown on the rightmost plot.

fed to multiple (parallelizable) NNs that provide the corresponding class probability conditional on a specific delay/Doppler hypothesis or bin. The concept is sketched in Fig. 2.10, where the $\{m, n\}$ -th sub-image corresponds to the correct location of the delay/Doppler. The output of the DNN structure, labeled as $K = 1$ in the plot, is the probability ratio map derived by the Bayesian hypothesis test.

The sub-images are, possibly overlapping, regions of the full CAF map which are centered at a specific delay/Doppler bin hypothesis. Recall that indices m and n map to the corresponding delay $[\tau]_m$ and Doppler $[f_d]_n$ values, respectively. Therefore, the $\{m, n\}$ -th sub-image for the i -th satellite will be defined as

$$\mathbf{Z}_i^{(m,n)} = [\mathbf{Z}_i]_{m+\delta_m, n+\delta_n} \quad (2.26)$$

where $\delta_m = [-\Delta_m, \dots, 0, \dots, \Delta_m]$ and $\delta_n = [-\Delta_n, \dots, 0, \dots, \Delta_n]$ for some positive integers $\Delta_m, \Delta_n \in \mathbb{Z}^+$. Thus, resulting in a sub-image dimension of $(2\Delta_m + 1) \times (2\Delta_n + 1)$ which is much smaller than the original CAF dimension of $n_\tau \times n_f$.

Fig. 2.11 provides an example of an arbitrary sub-image $\mathbf{Z}_i^{(m,n)}$. As a consequence of the splitting image approach, the statistical test in (2.24) is in reality implemented for each sub-image such that

$$\mathcal{T}(\mathbf{Z}_i^{(m,n)}) \triangleq \frac{p(\mathcal{H}_1 | \mathbf{Z}_i^{(m,n)})}{p(\mathcal{H}_0 | \mathbf{Z}_i^{(m,n)})} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\geq}} 1 \quad (2.27)$$

is computed for each $\{m, n\}$ pair, resulting in a *probability ratio map* (in contrast to the CAF map) for every test delay and Doppler value in τ and f_d . Recall that $m = \{1, \dots, n_\tau\}$ and $n = \{1, \dots, n_f\}$.

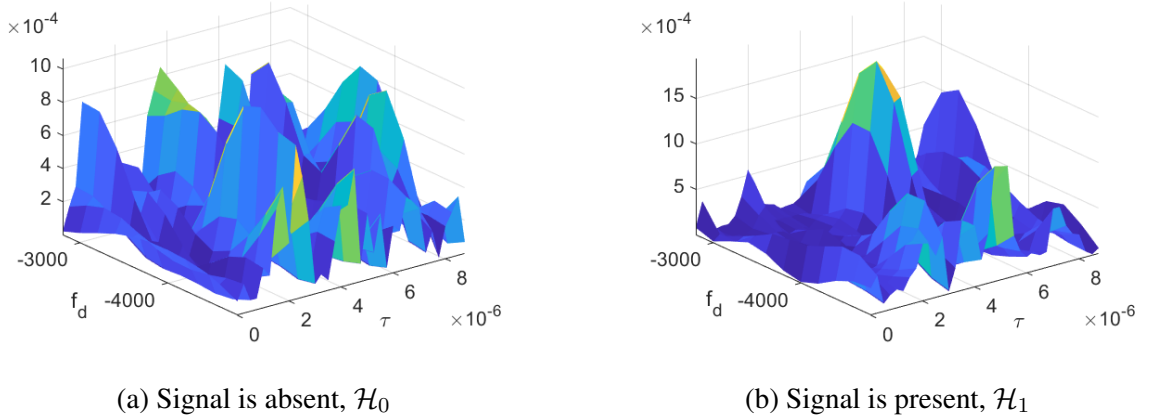


Figure 2.11: Portions of the CAF fed for processing to the NN with $\Delta_m = 18$ and $\Delta_n = 5$ defining the size of the $\{m, n\}$ -th sub-image. The resulting sub-image $\mathbf{Z}_i^{(m,n)}$ is shown on the reduced delay/Doppler grid in the case of (a) absence and (b) presence of a GNSS signal with $C/N_0 = 39$ dB-Hz. In the absence of signal, samples are i.i.d., while a time-freq correlation can be observed in the presence of signal.

It is worth noting that the probability ratio map may contain false peaks, as shown in Fig. 2.10 under $K = 1$. To mitigate those potential false detections, section 2.2.2 describes a methodology to fuse non-coherent integrations of K DNN outputs. The effect of those integrations is depicted in Fig. 2.10 in the rightmost panel for $K = 6$ non-coherent integrations, where the signal probability is accentuated in the correct delay/Doppler bin while false peaks arose from noise are attenuated in the fused probability ratio map.

2.2.2 Non-coherent integration through fusion of classifiers

Coherent integration of long code sequences can be implemented in computing the CAF map, $\mathcal{C}_i(\cdot, \cdot)$, in the usual manner. In implementing non-coherent integrations, an alternative is to fuse the multiple probability ratio maps resulting from processing CAF images through the NN architecture described earlier in Section 2.2.1. We denote by $K \in \mathbb{Z}^+$ the total number of non-coherent integrations. This section discusses the data fusion of such multiple classifiers. It is known that increasing integration time (both coherently and non-coherently) improves the overall detection performance of the acquisition process, this same rational holds in the case of the data-driven classifier proposed here, whereby non-coherent integrations (i.e. fusion of multiple classifier solutions) improves the reliability of the so-called probability maps (i.e. by attenuating falsely

detected peaks or enhancing locations where actual signals reside).

When processing non-coherent snapshots of data, a set of K CAF maps are computed. In the standard approach, that would correspond to full CAF maps $\mathbf{Z}_{i,k}$ with $k = 1, \dots, K$. In the sub-image approach, the result is a different sub-image for every integration period, $\mathbf{Z}_{i,k}^{(m,n)}$. In order to combine the class probabilities of the K classifiers (which are assumed conditionally independent given their own data), we use Bayes' rule to derive an optimal fusion rule. For an arbitrary $\{m, n\}$ pair, the optimal Bayes detector based on the K non-coherent integrations is

$$\mathcal{T}(\mathbf{Z}_{i,1}^{(m,n)}, \dots, \mathbf{Z}_{i,K}^{(m,n)}) = \frac{p(\mathcal{H}_1 | \mathbf{Z}_{i,1}^{(m,n)}, \dots, \mathbf{Z}_{i,K}^{(m,n)})}{p(\mathcal{H}_0 | \mathbf{Z}_{i,1}^{(m,n)}, \dots, \mathbf{Z}_{i,K}^{(m,n)})} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\geq}} 1, \quad (2.28)$$

where using the conditional independence assumption of the K snapshots we obtain

$$\begin{aligned} p(\mathcal{H}_1 | \mathbf{Z}_{i,1}^{(m,n)}, \dots, \mathbf{Z}_{i,K}^{(m,n)}) &= \frac{\mathbb{P}(\mathcal{H}_1) \prod_{k=1}^K p(\mathbf{Z}_{i,k}^{(m,n)} | \mathcal{H}_1)}{p(\mathbf{Z}_{i,1}^{(m,n)}, \dots, \mathbf{Z}_{i,K}^{(m,n)})} \\ &\propto \left[\prod_{k'=1}^{K-1} \mathbb{P}(\mathcal{H}_1) \right]^{-1} \prod_{k=1}^K p(\mathcal{H}_1 | \mathbf{Z}_{i,k}^{(m,n)}) \end{aligned} \quad (2.29)$$

and

$$\begin{aligned} p(\mathcal{H}_0 | \mathbf{Z}_{i,1}^{(m,n)}, \dots, \mathbf{Z}_{i,K}^{(m,n)}) &= \frac{\mathbb{P}(\mathcal{H}_0) \prod_{k=1}^K p(\mathbf{Z}_{i,k}^{(m,n)} | \mathcal{H}_0)}{p(\mathbf{Z}_{i,1}^{(m,n)}, \dots, \mathbf{Z}_{i,K}^{(m,n)})} \\ &\propto \left[\prod_{k'=1}^{K-1} \mathbb{P}(\mathcal{H}_0) \right]^{-1} \prod_{k=1}^K p(\mathcal{H}_0 | \mathbf{Z}_{i,k}^{(m,n)}) \end{aligned} \quad (2.30)$$

which explicitly contain the binary class probabilities of the K classifiers: $p(\mathcal{H}_0 | \mathbf{Z}_{i,k}^{(m,n)})$ and $p(\mathcal{H}_1 | \mathbf{Z}_{i,k}^{(m,n)})$. The statistical test can then be formulated as

$$\mathcal{T}(\mathbf{Z}_{i,1}^{(m,n)}, \dots, \mathbf{Z}_{i,K}^{(m,n)}) = \prod_{k=1}^K \frac{p(\mathcal{H}_1 | \mathbf{Z}_{i,k}^{(m,n)})}{p(\mathcal{H}_0 | \mathbf{Z}_{i,k}^{(m,n)})} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\geq}} \left(\frac{\mathbb{P}(\mathcal{H}_1)}{\mathbb{P}(\mathcal{H}_0)} \right)^{K-1} \triangleq \gamma \quad (2.31)$$

such that the decision threshold becomes $\gamma = 1$ when $\mathbb{P}(\mathcal{H}_0) = \mathbb{P}(\mathcal{H}_1)$. It can be observed that the optimal fusion rule is to multiply the K binary class probabilities (similar to what was shown in [58]) resulting from the K non-coherent integrations processed by the NN classifier. The role of the decision threshold is relevant, as will be discussed later, in establishing P_d and P_{fa} of the overall classifier. A reasonable choice is to assume that both hypotheses are equally probable, such that $\gamma = 1$.

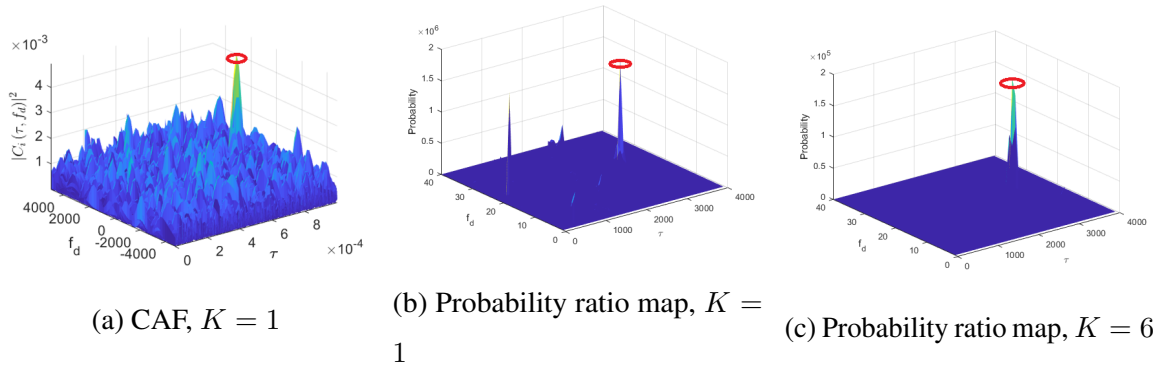


Figure 2.12: Comparison of the delay/Doppler grid for (a) standard CAF map with coherent integration only, (b) probability map produced by the data-driven classifier with coherent integration only, and (c) probability map after fusing $K = 6$ non-coherently classifier outputs. The GNSS signal had a C/N_0 of 42 dB-Hz, and the red circled highlights the location of the peak generated by the GNSS signal.

A qualitative example of how the fusion rule impacts the performance of the classifier is provided in Fig. 2.12. On the one hand, Fig. 2.12a shows the CAF delay/Doppler map used in standard signal acquisition without any non-coherent integration and just 1 ms coherent integration. It can be seen, as it is known from GNSS literature, that outside the true peak (denoted with a red circle) the noise floor is relatively spiky and can cause substantial false alarms, particularly at low C/N_0 values. On the other hand, the proposed data-driven method takes the CAF values and processes them to produce the so-called probability ratio maps, as defined on the right-hand side of (2.31). The probability ratio map resulting from processing the CAF in Fig. 2.12a can be observed in Fig. 2.12b where it is clear that the variability in the noise floor has been reduced, although residual spikes can still be detected at delay/Doppler bins where no signal was present. This effect is smoothed further with the fusion method, as shown in Fig. 2.12c where $K = 6$ non-coherent integrations were considered. Notice that the NN uses sub-images as inputs to produce a class probability pair, as depicted in Fig. 2.11. As a consequence, the posterior probabilities are taking into consideration the delay/Doppler correlations of the CAF around the signal peak, in contrast to the standard method which only considers the maximum value of the CAF thus neglecting the waveform arising from the noise form (i.e. the autocorrelation function of the corresponding spreading code).

2.2.3 Model training

This section provides details on how the model was trained. Particularly, we used a realistic GNSS signal simulator to generate I&Q samples from GPS L1 C/A satellites with various parameters according to the training plan described here. In order to increase the detection and localization accuracy, a larger sampling frequency might be desirable, since that accentuates the correlates samples around the CAF peak. However, this has an impact in the number of samples to be processed, and a trade-off needs to be considered. Therefore, here we increased the sampling frequency to 4 MHz, compared to the 2 MHz that were considered in our preliminary work [19]. As discussed earlier in Section 2.2.1, in order to reduce the complexity regardless of increasing f_s , the full CAF image is split and a sliding DNN scheme is considered in this work.

More precisely, a dataset of consisting of three thousand snapshots of GPS L1 C/A, I&Q samples were generated for model training purposes. The dataset consisted of a range of representative carrier-to-Noise-density ratios (C/N_0) varying between 33 to 45 dB-Hz. The length of those snapshots was 1 ms, the duration of a code, such that this constitutes the coherent integration time of the approach. Additionally, the dataset was generated with random delays between 0 to 1 ms and Doppler shifts between -4000 to 4000 Hz. These I&Q samples were then processed to compute the CAF maps over the Doppler-delay grid, which is then split and processed through the DNN model considered in this article. An analogy to images can be made for these CAFs where each Doppler/delay cell is a pixel whose value is that of the CAF, \mathbf{Z}_i . As discussed earlier in Section 2.2.1, this can be computationally expensive if a single NN has to process \mathbf{Z}_i entirely. For instance, if 20 Doppler bins are considered (i.e., 200 Hz bins) in generating the CAF for a GPS L1 C/A signal, those images would be 4000×50 dimensional for the f_s considered in this work. Alternatively, if \mathbf{Z}_i is split into smaller images of size 11×36 (read as: Doppler \times delay) there are a total of 158600 low-dimensional sub-images to be efficiently processed by the NN, potentially in parallel. An interpretation of the sliding concept is similar to the convolutional layers in a CNN, with a stride of one sample, whereby the entire CAF is scanned in smaller windows that can contain the main signal peak of interest. This peak, in contrast to peaks generated by random noise, show a correlation in the delay and (more noticeable) in the Doppler domains that can be exploited by the NN classifier. In order to train the NN-based classifier, the generated dataset contained either signal-plus-noise (\mathcal{H}_1) or noise only (\mathcal{H}_0) snapshots, which were then split into sub-images as shown in Fig. 2.11. The classifier learnt its parameters by observing a set of 3000 input/output pairs in a supervised manner. The output of the NN was a *softmax* layer with dropout, such that the resulting outcome of the NN

CHAPTER 2. GNSS SIGNAL ACQUISITION IN THE ABSENCE OF SPOOFER

are the binary class probabilities required to compute the test in (2.27) or its non-coherent version in (2.31).

The particular convolution neural networks structure that was based on the architecture in [54], containing 7 convolution layers and 3 fully connected layers. Each convolutional layer was followed by a batch normalization layer and a ReLU activation function. The batch normalization layers are used to normalize the activation and gradients propagation through the network between the convolution and ReLU layers, which is known to speed up network training tasks [57]. Each fully connected layer follows up with ReLU activation function and a dropout layer with 1/2 probability rate. Since the task is a binary classification, The last fully connected layer contains two neurons, predicting the posterior probabilities of each hypothesis. Other relevant training options were specified, such that the use of a SGDM optimizer with an initial learning rate of 0.001. The maximum number of epochs, which is a full training cycle on the entire training dataset, was set to 30 and at every epoch, the data was shuffled. After 20 epochs, the learning rate dropped by a factor of 0.1. The loss, which SGDM optimizes, was the cross-entropy loss and the accuracy was defined as the percentage of inputs that the network classified correctly. Particularly, the validation accuracy after training the model reached 93%, which is deemed a high enough rate to consider the NN ready for deployment. Section 3.2.5 provides testing results of the trained and validated model, showing ROC performances and other relevant metrics.

2.2.4 Results

The proposed data-driven signal detection scheme was tested, and its performance assessed through simulated data. The details on the model can be consulted in Section 2.2.1, while the training process is discussed in Section 2.2.3. While training of the model was conducted using CAF images produced by 1 ms coherent integration times, the overall method was tested with and without non-coherent integration schemes. Particularly, when considering non-coherent integration times, $K = 6$ were considered. To assess the performance of the detection scheme, its ROC curves were empirically obtained through simulations and compared to the theoretical performance of standard methods (as reviewed in Section 2.1.4 or more in depth in [6]). Fig. 2.13 provides results for $K = 6$ non-coherent integration periods (dashed lines), as compared to the theoretical performance (solid lines) of standard methods (aiming at maximizing the CAF) with those coherent/non-coherent values. Results show that whereas at low C/N_0 values the proposed method can barely achieve the state-of-the-art performance, it does remarkably well at larger C/N_0 values. Worth noting that the

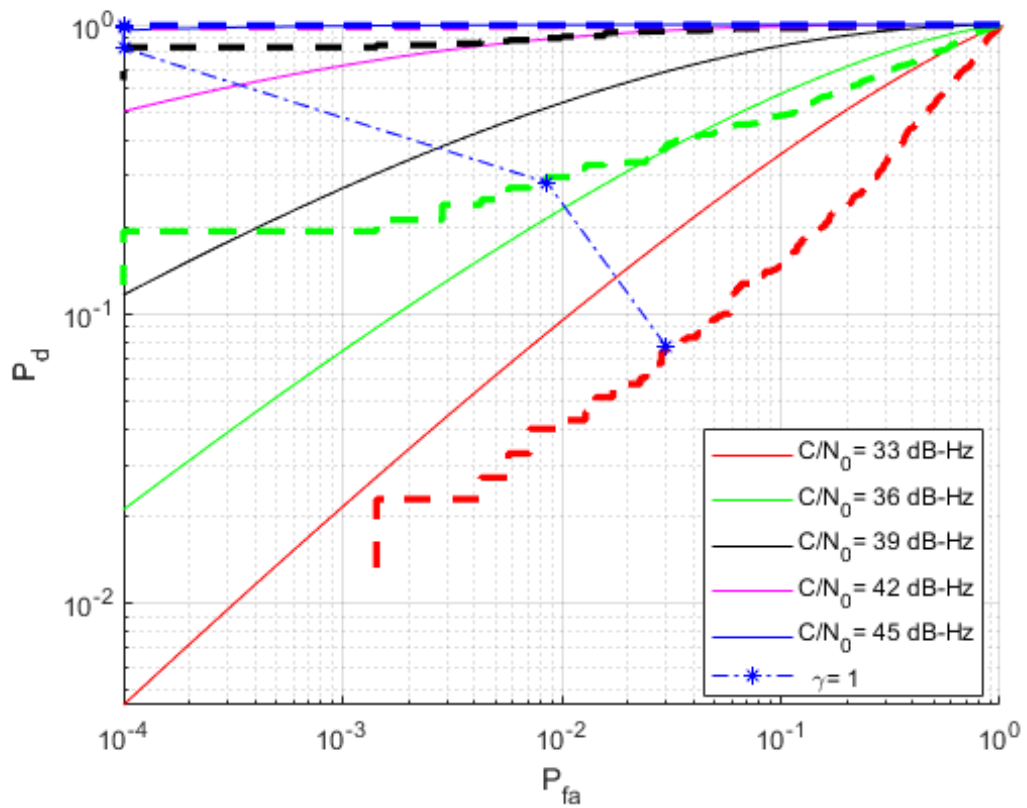


Figure 2.13: ROC curves for a 1 ms coherently integrated snapshot and $K = 6$ non-coherently processed blocks for a variety of C/N_0 values. The performance of the proposed scheme (dashed lines) is compared to the theoretical performance of standard methods (solid lines).

CHAPTER 2. GNSS SIGNAL ACQUISITION IN THE ABSENCE OF SPOOFER

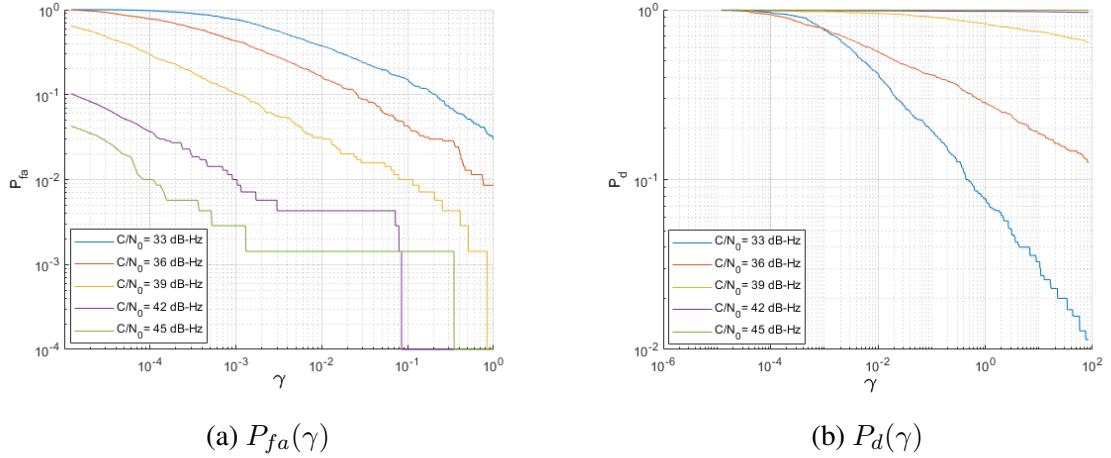


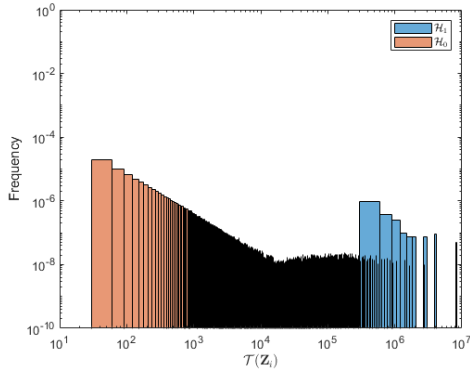
Figure 2.14: (a) $P_d(\gamma)$ and (b) $P_{fa}(\gamma)$ probabilities for a 1 ms coherently integrated snapshot, $K = 6$ non-coherent processing, and a variety of C/N_0 values.

improved performance starts at C/N_0 as low as 36 dB-Hz which could be considered to be on the limit of the moderate-low range. An explanation is that for low C/N_0 the DNN cannot extract the relevant features from the corresponding *sub-image*, $\mathbf{Z}_i^{(m,n)}$, but at higher C/N_0 values the relevant features can be extracted, and the classification task successfully performed with desirable P_d and P_{fa} rates. Surprisingly, the results in Fig. 2.13 also show that the proposed data-driven scheme outperforms current performance bounds, suggesting it is leveraging additional information. This additional information comes from the prior that is embedded in the classifier through the seen training dataset. More precisely, whereas standard methods are based on the maximization of the CAF and identifying the associated bin, the proposed data-driven method exploits the correlation across neighboring bins to compute the class probability. That is, the classifier uses a sub-image that contain a detail of the CAF that under \mathcal{H}_1 contains the relevant waveform of the CAF and its delay/Doppler correlated values).

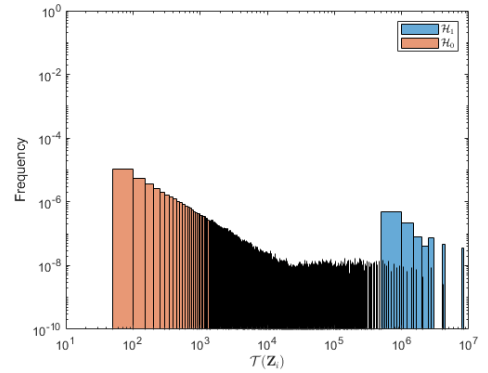
On the other side, it is worth mentioning that the performance of the scheme for $K = 1$ is substantially degraded compared to standard model-based schemes. This is explained by the low signal-to-noise ratio in this situation, as argued similarly earlier.

A benefit of the Bayes test approach considered in this work is that the adjustment of the detection threshold γ in (2.27) (or the one in (2.31) when $K > 1$) has a probabilistic interpretation: how much larger the posterior probability of \mathcal{H}_1 has to be from \mathcal{H}_0 to be accepted. A reasonable choice would be $\gamma = 1$, such that one picks the class with largest posterior probability. Fig. 2.13 shows the ROC results when such a choice is made for the detection threshold.

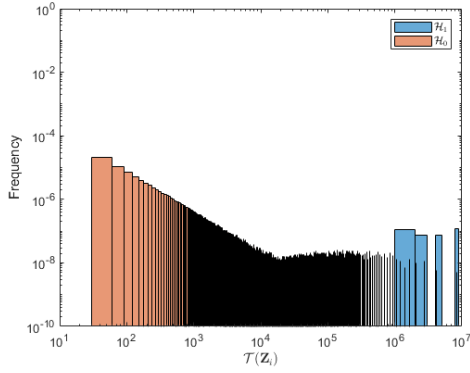
CHAPTER 2. GNSS SIGNAL ACQUISITION IN THE ABSENCE OF SPOOFER



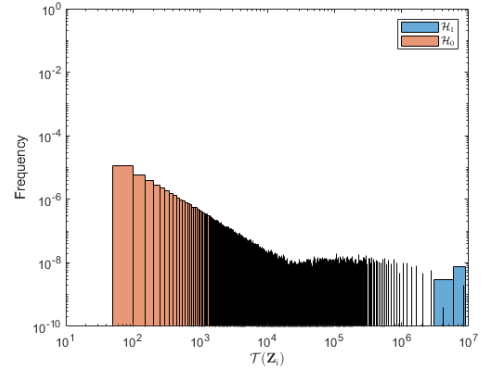
(a) $C/N_0 = 36$ dB-Hz, $K = 1$



(b) $C/N_0 = 39$ dB-Hz, $K = 1$



(c) $C/N_0 = 42$ dB-Hz, $K = 1$



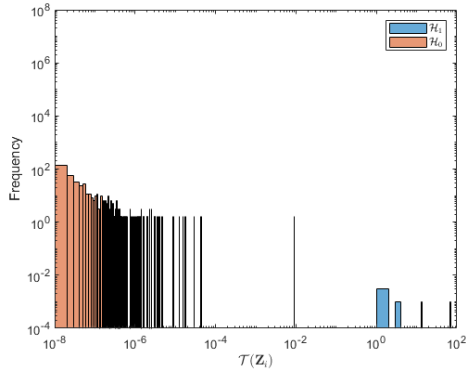
(d) $C/N_0 = 45$ dB-Hz, $K = 1$

Figure 2.15: Test statistic histograms under \mathcal{H}_0 and \mathcal{H}_1 hypotheses for a 1 ms coherent integration time for a range of relevant C/N_0 values. The two histograms have overlapping areas, which suggest poor detection performance in these conditions.

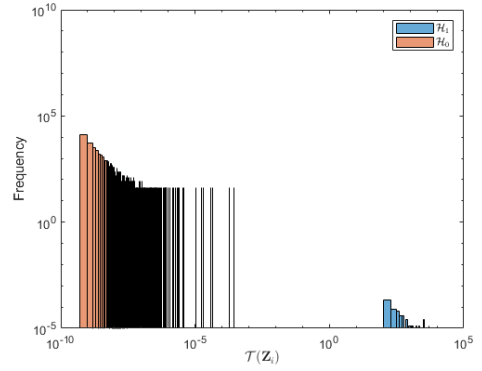
According to the results, $\gamma = 1$ provides good results for $C/N_0 > 36$ dB-Hz, with low false alarm and outstanding detection probabilities. For the sake of completeness, Fig. 2.14 shows the false alarm and the detection probabilities corresponding to the ROC in Fig. 2.13. The impact of low C/N_0 values on ROC curves is further explained by the histograms of the test statistic $\mathcal{T}(\mathbf{Z}_i^{(m,n)})$, which are shown in Fig. 2.15 and 2.16 for the cases of $K = 1$ and $K = 6$, respectively. Recall that one would like to have the histograms under \mathcal{H}_0 and \mathcal{H}_1 as *distant* as possible, which happens for large C/N_0 but clearly does not for low C/N_0 values. In Fig. 2.16 the empirical distributions can be clearly distinguished in compare with Fig.2.15 that a sample from $\mathcal{T}(\mathbf{Z}_i)$ cannot be statistically discerned between both distributions.

More precisely, without the non-coherent fusion rule it is hard for the DNN to distinguish

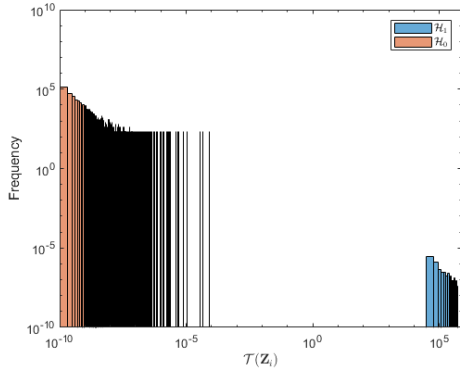
CHAPTER 2. GNSS SIGNAL ACQUISITION IN THE ABSENCE OF SPOOFER



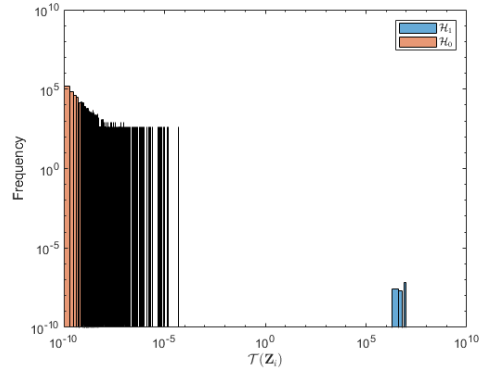
(a) $C/N_0 = 36$ dB-Hz, $K = 6$



(b) $C/N_0 = 39$ dB-Hz, $K = 6$



(c) $C/N_0 = 42$ dB-Hz, $K = 6$



(d) $C/N_0 = 45$ dB-Hz, $K = 6$

Figure 2.16: Test statistic histograms under \mathcal{H}_0 and \mathcal{H}_1 hypotheses for a 1 ms coherent integration time and $K = 6$ non-coherent integrations for a range of relevant C/N_0 values. In this case the two histograms are clearly separated, which supports the performance results in Fig. 2.13.

the difference between noise and signal sub-images, particularly when the C/N_0 is low, which is shown in Fig. 2.15. Although for large C/N_0 the separation increases, it is still far from desirable ROC regions. On the other hand, when non-coherent integration is considered, Fig. 2.16 shows that there is no overlap between the two histograms, which causes and increased detection accuracy.

In summary, when the signal power is high enough (or when non-coherent integration is used to increase that power) the DNN classifier performs remarkably well, even exceeding current state-of-the-art well-known performance results.

Part II

Spoofers Detection

Chapter 3

GNSS Signal Acquisition in the presence of Spoofers

This chapter focuses on the signal detection process in presence of a spoofer and will explore the capabilities of deep learning models in complementing the standard design of GNSS receivers, which to the best of our knowledge has not been explored in the past, which is shown in Fig.3.1, in which when the Spoofer is present, the receivers can receive two types of data, Spoofed-free GNSS and Spoofed GNSS signal. These data will be fed to the NNs which expected can be detected by NNs.

3.1 Deep learning of GNSS spoofing signals

The goal of this work, is to detect the GNSS signal spoofing attack. In particular, to use a DNN to carry out the detection (or classification) task. Two different hypotheses considered, the null hypothesis (\mathcal{H}_0), that the legitimate signal and noise are present, but there is no spoofing signal and the alternative hypothesis (\mathcal{H}_1), that both the legitimate signal, spoofed signal, and noise are present in the dataset.

To this end, three different neural networks designed and evaluated the two hypotheses detection process such as Complex-CNN, Simple-CNN, and MLP. The networks are trained with two different optimizers for various C/N_0 between 33 and 45 dB-Hz at a different delay and Doppler frequency with 1 ms coherent and 10 non-coherent integration configuration.

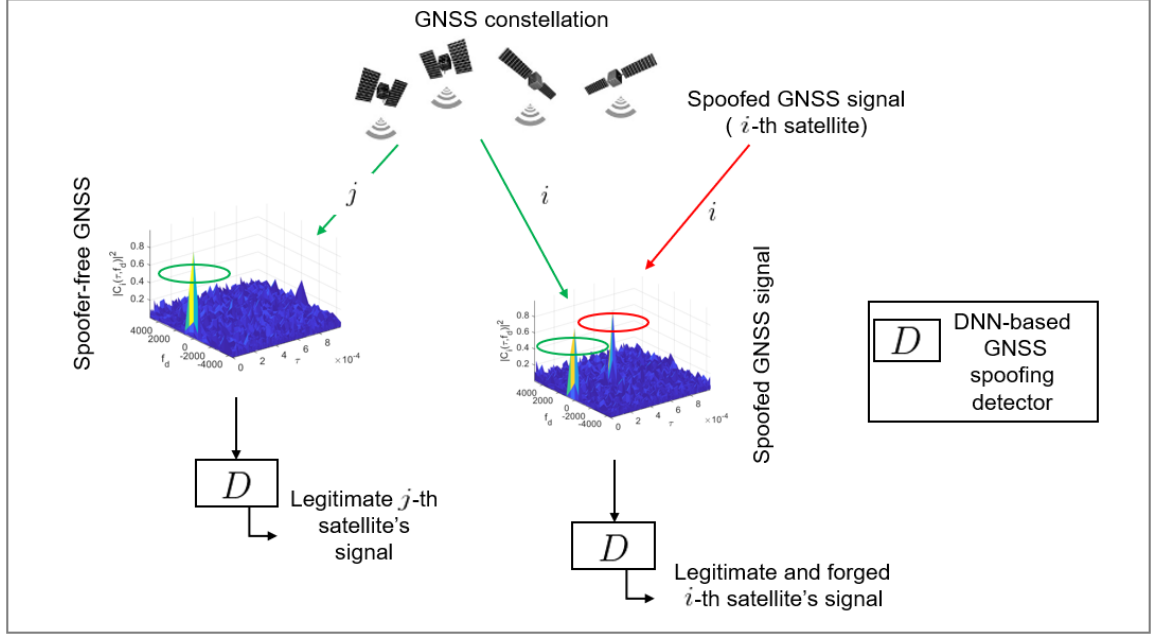


Figure 3.1: Deep learning signal detection process in the presence of spoofer

3.1.1 GNSS Spoofer detection

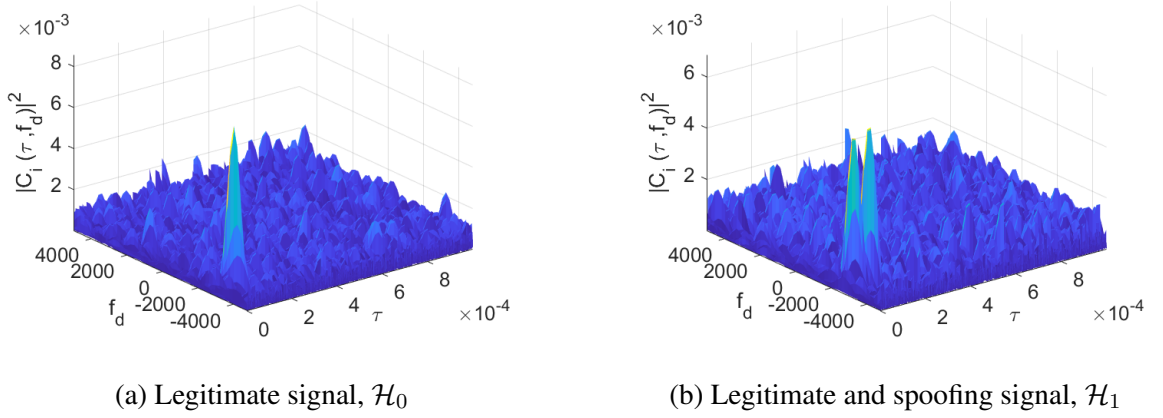
To understand the impact of a spoofing attack - and its potential countermeasures -, we first recall that the first stage in the operation of a GNSS receiver is acquisition. This process results in deciding whether the satellite signal is present or absent in the received signal, as well as provides rough estimates of the parameters, such as code delay and Doppler shift, of the signal transmitted by the satellite. All GNSS receivers [6, 59, 60, 61] implement such an acquisition process by evaluating the so-called CAF, usually in discrete-time.

In order to detect the presence of visible satellite signals, a receiver performs a two-dimensional search. The acquisition process involves calculating the CAF value for a given satellite by correlating the received signal and local pseudo-random code for every possible delay/Doppler pair in the search space. When the signal is present, the CAF exhibits a large peak for a specific delay/Doppler cell, which is then passed on to the tracking loops for fine estimation and tracking. Fig. 3.2 shows the exemplary representation of this process.

The received GNSS signal is considered as:

$$X(t) = \sum_{i=1}^N \alpha_i b_i(t - \tau_i(t)) c_i(t - \tau_i(t)) \exp(j(\omega_{IF}t - \omega_c \tau_i(t) - \phi_i(t)))$$

where N is the number of spreading code, $\alpha_i(t)$ is the carrier amplitude of the i -th signal, b_i is the


 Figure 3.2: CAF evaluation at the delay/Doppler grid with $C/N_0 = 45$ dB-Hz.

i -th signal's data bit stream, $c_i(t)$ is its spreading code, $\tau_i(t)$ is the i -th signal's code phase, ω_c is the carrier frequency, ω_{IF} the intermediate frequency, and $\phi_i(t)$ is the i -th carrier phase. The spoofer sends a set of false signals that are similar to the true signal, except for those parameters that would eventually cause a different position estimate at the receiver unless properly detected.

A received GNSS spoofed signal is:

$$X_s(t) = \sum_{i=1}^{N_s} \alpha_{s,i} b_i(t - \tau_{s,i}(t)) c_i(t - \tau_{s,i}(t)) \exp(j(\omega_{IF}t - \omega_c \tau_{s,i}(t) - \phi_{s,i}(t)))$$

where N_s , denotes the number of spoofed signals. In order to deceive the receiver, each spoofed signal must have the same spreading code $c_i(t)$ as the corresponding true signal and broadcast its best estimate of the same data bit stream b_i , the spoofed amplitude $\alpha_{s,i}(t)$, code phases $\tau_{s,i}(t)$, and carrier phases $\phi_{s,i}(t)$ for $i = 1, \dots, N$ are different from those of the true signal [62].

The total signal at the victim receiver antenna during a spoofing attack is:

$$Y(t) = X(t) + X_s(t) + \eta(t) \quad (3.1)$$

where $\eta(t)$ is the received noise, typically modeled as zero-mean, additive, white, and Gaussian.

Therefore, two hypotheses are tested:

1. The null hypothesis (\mathcal{H}_0), that the legitimate signal and noise are present, but there is no spoofing signal.

$$\mathcal{H}_0 : Y(t) = X(t) + \eta(t)$$

2. The alternative hypothesis (\mathcal{H}_1), that both the legitimate signal, spoofed signal, and noise are present in the dataset;

$$\mathcal{H}_1 : Y(t) = X(t) + X_s(t) + \eta(t)$$

In this context, we propose to treat the spoofing detection problem as a hypothesis testing problem on the CAF's delay/Doppler map for each satellite independently, for instance, the i -th satellite. After downconversion and sampling (at a rate $f_s = 1/T_s$), the CAF can be computed as the correlation of the digital signal and the known local code for the i -th satellite. At a given delay/Doppler pair:

$$C_i(\tau, f_d) = \frac{1}{N} \sum_{n=0}^{N-1} y[n] \underbrace{c_i(nT_s - \tau) \exp\{-j2\pi f_d nT_s\}}_{\text{Local replica}}, \quad (3.2)$$

which can be expressed more compactly in vector notation after gathering N samples from the samples and the local code as $\mathbf{y}, \mathbf{c}_i \in \mathbb{C}^{1 \times N}$ as

$$C_i(\tau, f_d) = \frac{\mathbf{y} \mathbf{c}_i^H}{N}. \quad (3.3)$$

The effect of a spoofing signal on the CAF is well-known and shown in Fig. 3.2 for clarity, showing an arbitrary CAF under both hypotheses. This work proposes to train a DNN, data-driven model to learn to classify between spoofed or clean signal receptions.

Three types of error probabilities characterize the performance of the detector: detection (the probability of correctly detecting signal+spoofed when there is signal+spoofed), false-alarm (the probability of wrongly detecting satellite is spoofed when there is an only signal), and miss-detection (the probability of mistakenly deciding for the null hypothesis when the satellite is spoofed). The two first probabilities are used in order to obtain an important figure of merit in acquisition performance: the ROC, a plot of the probability of detection as a function of the probability of false alarm [6, 63, 64]. We will use ROC curves in evaluating the performance of the proposed detection method.

3.1.2 Deep Neural Networks model and problem statement

NNs are models composed of neurons, which are information processing units, for complex data processing. A NN typically contains an input layer, one or more hidden layers, and an output layer, as well as pre-defined activation functions that connect adjacent layers. Each layer has a specific weight, which is usually determined with backpropagation during a training process that

CHAPTER 3. GNSS SIGNAL ACQUISITION IN THE PRESENCE OF SPOOFER

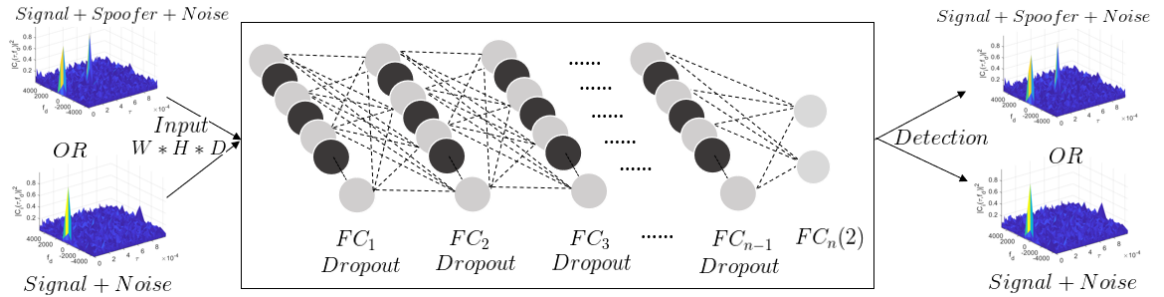


Figure 3.3: Detection scheme of GNSS acquisition by using the Fully Connected Structure of Deep Learning approach.

involves large amounts of data with known labels [65, 66]. One of the most important part in NN is how to design and choose a network to get the high accuracy with less network complexity. some of effective aspects are: number of layers, number of neurons, and type of optimizer. In the simulation and result section, the accuracy of different types of networks is discussed more. In this work, in order to detect the spoofed signal, the performance of two different structure of neural networks are evaluated and compared together: a MLP and a CNN.

3.1.2.1 MLP Network

The first neural network structure that will be used in this research is the MLP, which is referred to as a traditional neural network. This type of network is composed of one or more layers of neurons (which consist of a row of neurons). Fig.3.3 shows an exemplary representation of this type of network. The first layer is called the input layer, which is fed with the training dataset for learning the parameters of the (potentially several) hidden layers that are not directly exposed to the input. During training, the number of nodes in the hidden layer are randomly ignored or “dropped out”, which are shown in black in Fig.3.3. The number of neurons in this last layer depends on the number of the classes that one wants to classify, since those provide their probabilities.

3.1.2.2 CNN Network

The second artificial neural network structure considered here is the so-called CNN, which is one of the most popular models for deep learning in the context of learning class labels from image datasets. A CNN can have tens or hundreds of layers, where each of these layers learns to identify different features of an image [55]. At each layer, filters are applied to each training image and the output of each convolution image is used as an input to the next layer. Fig.3.4 illustrates a CNN

CHAPTER 3. GNSS SIGNAL ACQUISITION IN THE PRESENCE OF SPOOFER

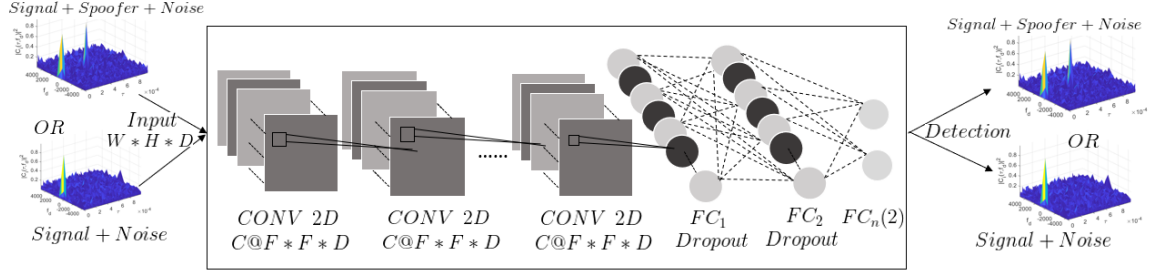


Figure 3.4: Detection scheme of GNSS acquisition by using the Convolutional Neural Network Structure of Deep Learning approach.

structure. During training, the input size of the CNN is fixed, the input is going through a stack of convolutional layers with the same or different filter sizes. In each convolution layer, the filter sweeps the input image from left to right and up to down by using a stride with 2 pixels size, which is the number of pixels that each time the filter shifts. In the end, the convolution layers are followed by FC layers and a final softmax layer, which is used for classification purposes [55].

3.1.3 Probabilistic learning

The aforementioned DNN models have the task of classifying CAF maps into spoofed/cleaned cases. We aim at doing that in a probabilistic sense, thus providing probabilities for both hypotheses. The inputs to the NNs is the CAF evaluated at the delay/Doppler grid, which can be considered as an *image*. Such images (refer to Fig. 3.2 for an exemplary situation) has certain characteristics that can be used to determine whether the signal is present or not: (i) in the absence of spoofing signal, the image should exhibit a single peak corresponding to the legitimate satellite signal (if received with enough power); and (ii) in the presence of a spoofing signal, the CAF *image* should be composed of two peaks and exponentially distributed noise in the remaining cells. This is used to train a NN model to classify between \mathcal{H}_0 and \mathcal{H}_1 , the hypotheses described earlier. The proposed methodology works on a per-satellite basis. Recall that the input data fed to the NNs is the corresponding CAF image for the satellite, which we denote with \mathbf{Z}_i in the sequel. That is, the $\{m, n\}$ element of the input matrix is defined as:

$$[\mathbf{Z}_i]_{m,n} = |\mathcal{C}_i([\boldsymbol{\tau}]_m, [\mathbf{f}_d]_n)|^2, \quad (3.4)$$

where $\boldsymbol{\tau}$ and \mathbf{f}_d are vectors containing the tested delay and Doppler-shifts, respectively. We use the convention that $[\mathbf{a}]_m$ represents the m -th element in the vector, \mathbf{a} and that $[\mathbf{A}]_{m,n}$ provides a shortcut

for the element of \mathbf{A} in the m -th row and n -th column.

From a Bayesian perspective, all the information resides in the *a posteriori* distribution of each hypothesis once data is observed. An optimal (Bayesian) test between \mathcal{H}_0 and \mathcal{H}_1 is given by the ratio,

$$\frac{p(\mathcal{H}_1|\mathbf{Z}_i)}{p(\mathcal{H}_0|\mathbf{Z}_i)} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} 1, \quad (3.5)$$

in which case we basically favor the model with largest a posteriori probability. This can be further expanded in terms of the likelihood and a priori distributions, as

$$\frac{p(\mathbf{Z}_i|\mathcal{H}_1) \mathbb{P}(\mathcal{H}_1)}{p(\mathbf{Z}_i|\mathcal{H}_0) \mathbb{P}(\mathcal{H}_0)} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} 1, \quad (3.6)$$

where we readily identify that $\mathbb{P}(\mathcal{H}_i)$ denotes the a priori probability of the i -th hypothesis. In the absence of better priors, we may assume equally likely hypotheses $\mathbb{P}(\mathcal{H}_0) = \mathbb{P}(\mathcal{H}_1) = 1/2$. Otherwise, we might incorporate that information in the hypothesis test, resulting in the adjustment of a threshold γ . The resulting test statistic is such that

$$\mathcal{T}(\mathbf{Z}_i) \triangleq \frac{p(\mathbf{Z}_i|\mathcal{H}_1)}{p(\mathbf{Z}_i|\mathcal{H}_0)} \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \gamma, \quad (3.7)$$

which would act as our spoofer detection algorithm, where threshold γ is a tuning parameter. Since the test statistic is a ratio of probabilities, we have that $0 < \mathcal{T}(\mathbf{Z}_i) < \infty$. Similarly, as in [19], the DNNs provide estimated probabilities for each of the hypotheses in (3.7). Therefore, the input data would be \mathbf{Z}_i and the output of the DNN would be the estimated probabilities in the dataset \mathbf{y} used to build \mathbf{Z}_i .

3.1.4 Simulation Environment and results

To accomplish the objective, we train DNNs using synthetically generated data. Particularly, we use a dataset consisting of 10000 snapshots of I&Q samples with different Carrier-to-Noise-density ratio (C/N_0) varying between 33 and 45 dB-Hz, as well as randomly generated delays between 0 and 1 ms and Doppler shifts between -4000 to 4000 Hz. These samples are then processed to compute the CAF over a Doppler-delay grid. An analogy to images can be made for these CAFs where each Doppler/delay cell is a pixel whose value is that of the CAF. For instance, if 20 Doppler bins are considered to acquire a GPS L1 C/A signal, those images would be 20×1023 dimensional. These images are fed to the input layer of the DNN, whose output would be the classification into a present/absent of the spoofed satellite signal. In a supervised training scheme, these input/output pairs are provided by labeling and using the aforementioned synthetic data generation. In the experiments,

CHAPTER 3. GNSS SIGNAL ACQUISITION IN THE PRESENCE OF SPOOFER

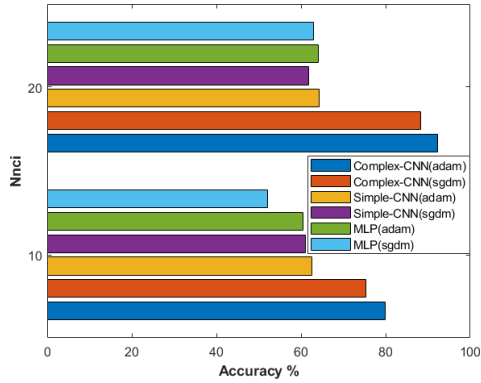
the ReLU activation function and two different optimize are used to train the network SGDM and Adam optimizer. In this work, two different NN structures will be considered: MLP, and CNN.

The model implemented with MLP structures considered different number of fully connected layers. Each fully connected layer followed up with rectified linear unit ReLU activation function to allows for faster and more effective training by mapping negative value to zero [67]. After each layer, a 1/2 dropout probability was considered. The last fully connected layer contained two neurons, used in predicting the class for which the input image belongs to. After defining the network structure, the training options were specified. The network trained with two different optimizers SGDM and Adam optimizer with an initial learning rate of 0.001. The maximum number of epochs, which is a full training cycle on the entire training dataset, was set to 30 and at every epoch the data was shuffled. After 20 epochs, the learning rate dropped by a factor of 0.1. the training progress shows the mini-batch loss and accuracy and the validation loss and accuracy. The loss is the cross-entropy loss and the accuracy was defined as the percentage of inputs that the network classified correctly.

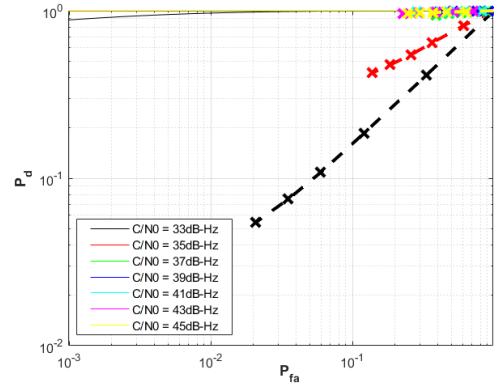
In this work, two different CNN structures are used. First, the simple CNN structure is used and evaluated to detect the Spoofed signal, which contains 3 convolution layers and 3 fully connected layer. Each convolution layer was followed by a batch normalization layer and a ReLU activation function. The batch normalization layers, normalizing the activation and gradients propagation through a network and using it between the convolution layer and ReLU layers to speed up network training [57]. Each fully connected layer follows up with the ReLU activation function and a dropout layer with a probability of 0.5. The last fully-connected layer contains two neurons to predict each image belongs to which class, since two types of classes are existing in this work. The result of this network was not promising since as it was said in [19] it is expected that CNN outperforms the MLP to detect the Spoofed signal. Therefore, the more complex CNN structure is decided to use and evaluated to detect the Spoofed signal.

The complex CNN structure which is used in this research is very similar to the VGG 16 structure training [54]. The network has 13 convolution layers that each layer was followed by a batch normalization layer and a ReLU activation function. Also, it contains 3 fully connected layers that each layer follows up with the ReLU activation function and a dropout layer with a probability of 0.5. The last fully-connected layer contains two neurons to predict each image belongs to which class, since two type of classes are existing in this work. In the end, after defining the network structure, the training options were specified, which were the same as for the MLP training options. Fig. 3.5a shows the training accuracy of these three networks with Adam and SGDM optimizer for C/N_0

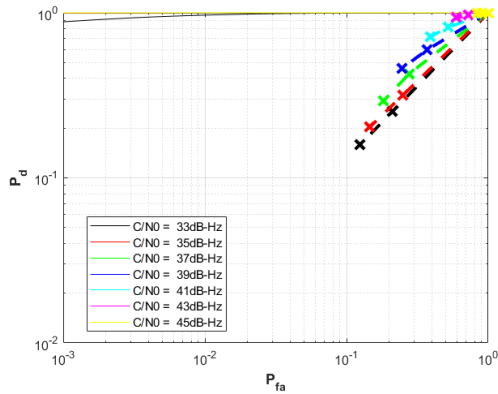
CHAPTER 3. GNSS SIGNAL ACQUISITION IN THE PRESENCE OF SPOOFER



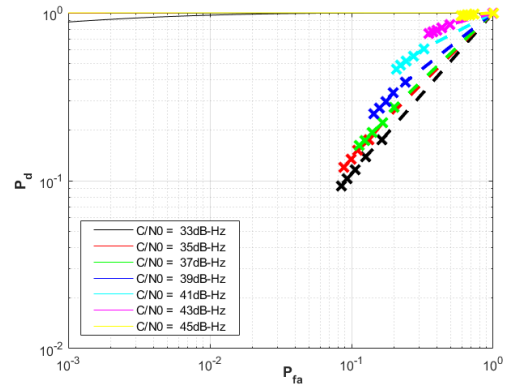
(a) Neural networks training accuracy for $C/N_0 = [33 - 45]$ dB-Hz



(b) ROC Curve Complex-CNN (VGG16) for $C/N_0 = [33 - 45]$ dB-Hz



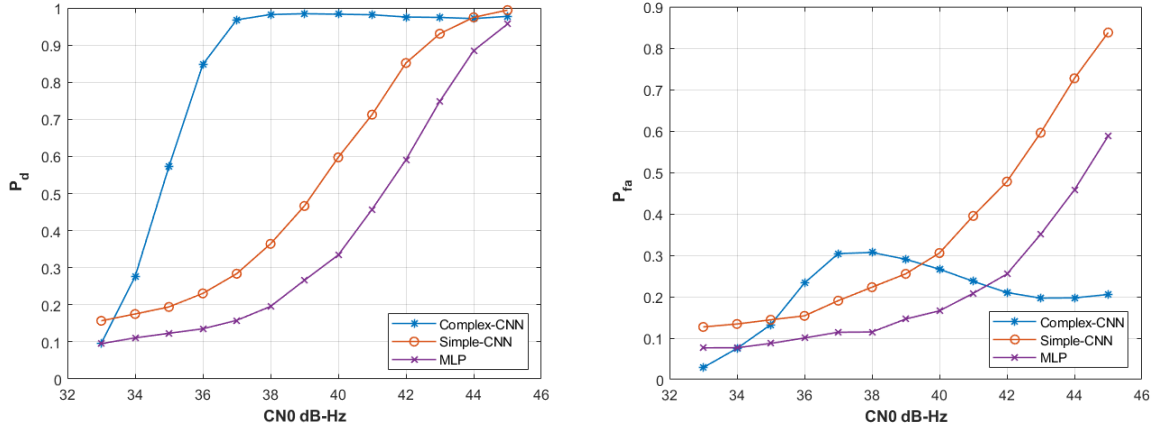
(c) Simple-CNN for $C/N_0 = [33 - 45]$ dB-Hz



(d) MLP for $C/N_0 = [33 - 45]$ dB-Hz

Figure 3.5: ROC curves for Simple-CNN and MLP using CAF generated with 1 ms coherent integration and 10 non-coherent integration

CHAPTER 3. GNSS SIGNAL ACQUISITION IN THE PRESENCE OF SPOOFER



(a) Probability of detection for different C/N_0 (b) Probability of false alarm for different C/N_0
 Figure 3.6: P_d and P_{fa} under 1 ms coherent and 10 non-coherent integrations for all three networks models.

between 33 and 45 dB-Hz with 2 different number of Number of non-coherent integration (N_{nci}) values. The result shows that the Complex-CNN (VGG16) network outperforms Simple-CNN and MLP. Moreover, in all networks, the Adam optimizer has better accuracy than the SGDM optimizer.

The detection process determines the presence or absence of the Spoofers and the output is the random variable, which is called a decision variable. If the Spoofers is present, The probability that the decision variable passes a threshold β is called the detection probability and if the Spoofers is absent it called the false alarm probability. Then the plot of detection probability (P_d) versus the false alarm probability (P_{fa}) is called the ROC. Fig. 3.5b and 3.5 show the ROC curve results that they are trained and tested under the same integration configuration for Complex-CNN, Simple-CNN and MLP respectively.

From Fig. 3.5b it can be observed that for low C/N_0 values, the performance of NNs is not attaining the theoretical ROC curves. However, as C/N_0 increases, such an approach can reach theoretical limits. An explanation could be that for low, C/N_0 the various NNs cannot extract the relevant features from the input images. When C/N_0 increased, either because of an actual power increase or longer integration times, the NN is able to perform classification on whether the Spoofers present or not. In comparing the ROC curve result of Fig. 3.5b with Fig. 3.5 it is shown that Complex-CNN outperforms the Simple-CNN and MLP network structures. The Simple-CNN and MLP are not able to reach the theoretical, even in higher C/N_0 . Indeed, these results were predictable from Fig. 3.5a since the neural network can be trained much better with Complex-CNN,

CHAPTER 3. GNSS SIGNAL ACQUISITION IN THE PRESENCE OF SPOOFER

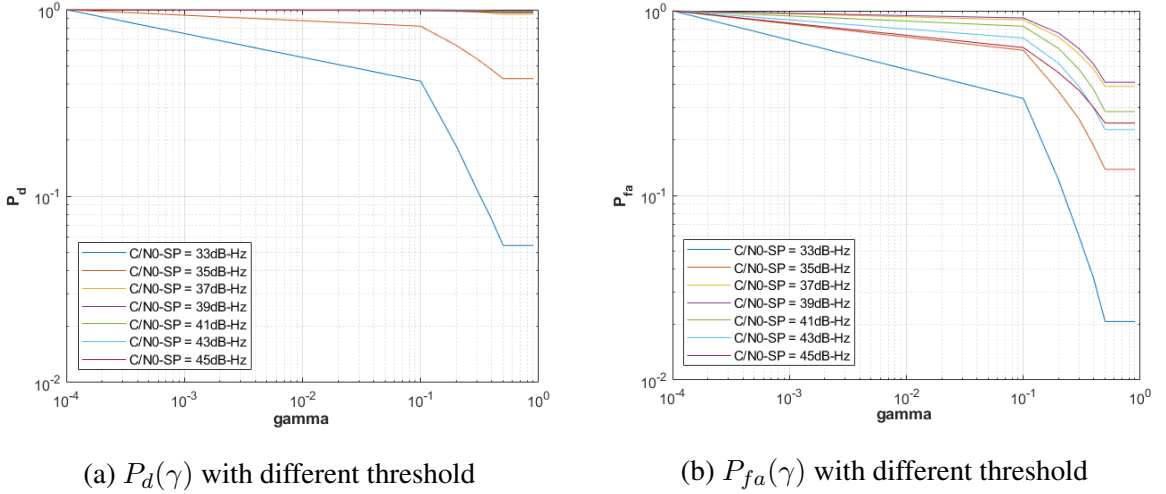


Figure 3.7: $P_d(\gamma)$ and $P_{fa}(\gamma)$ under 1 ms coherent and 10 non-coherent integration for Complex-CNN models.

which is 80%, in comparison with Simple-CNN and MLP, which are around 60%.

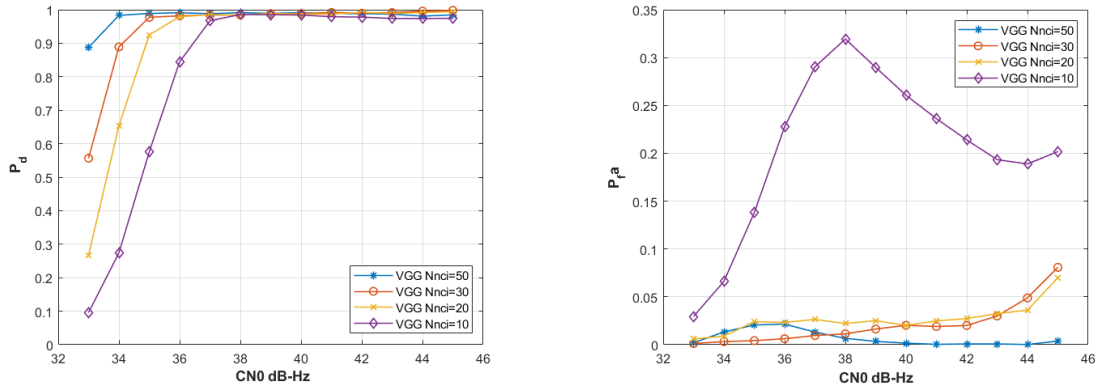
As it was discussed in [19] for simple scenario and dataset, the MLP and Simple-CNN are working similarly. However, for a more complex scenario, CNN outperforms the MLP because inCNN the input pass through a set of convolution filters, each of these filters activates certain features from the images and creates an output that causes to improve the detection. In order to, increase this feature designing more Complex-CNN would be helpful which is discussed in this work.

Fig. 3.6 shows the probability of detection and probability of false alarm of different C/N_0 for all 3 networks. As was discussed, the Complex-CNN has the best probability of the detection for all different C/N_0 , which is close to 90% for $C/N_0 = 36$ dB-Hz and close to 100% for $C/N_0 = 37$ dB-Hz and greater. Moreover, Simple-CNN and MLP have a worse probability of detection than Complex-CNN, even at higher C/N_0 . However, in this situation, Simple-CNN outperforms MLP.

In Fig. 3.7, detection and false alarm probabilities are plotted against the classification threshold (i.e., $P_d(\gamma)$ and $P_{fa}(\gamma)$), for different values of C/N_0 under 10 non-coherent integration for complex-CNN network models. It can be taken from these figures the probability of detection and probability of false alarm for all different C/N_0 converged when the (γ) is equal to 0.5.

The result of probability of detection and probability of false alarm at the convergence point ($\gamma = 0.5$) for different C/N_0 under 1 ms coherent and 10, 20, 30 and 50 non-coherent integration for Complex-CNN models is depicted in Fig. 3.8(a), 3.8(b) respectively. It can be taken from the

CHAPTER 3. GNSS SIGNAL ACQUISITION IN THE PRESENCE OF SPOOFER



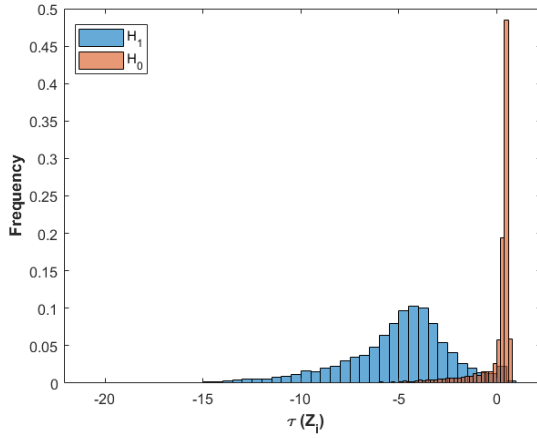
(a) Probability of detection for different C/N_0 (b) Probability of false alarm for different C/N_0

Figure 3.8: $P_d(\gamma)$ and $P_{fa}(\gamma)$ under 1 ms coherent and 10,20,30 and 50 non-coherent integration for Complex-CNN models.

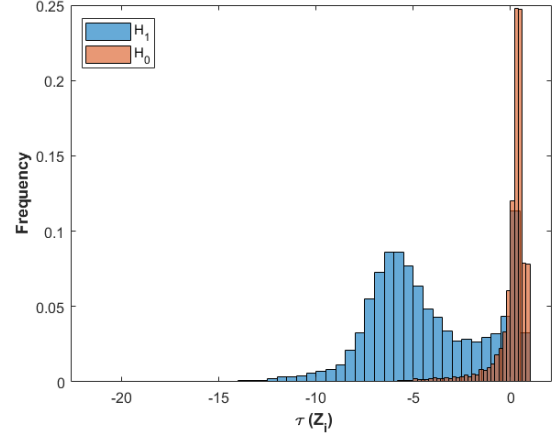
figures by increasing the number of non-coherent integration the probability of detection increased and the probability of false alarm decreased significantly. As it shows in fig. 3.8(a), the detection probability for C/N_0 38 and above for all number of non-coherent integration are the same and close to 100%, but increasing the number of non-coherent integration affected the result of lower C/N_0 significantly. For example, for C/N_0 33 when the number of non-coherent integration is equal to 10 the probability of detection is almost 10%, however, by increasing it to 50 the probability of detection to reach the 90%. A similar improvement happened for the probability of false alarm in Fig. 3.8(b) that show when the number of non-coherent integration is equal to 10 the probability of false alarm is almost 20% even for highest C/N_0 , however, by increasing it to 50 the probability of false alarm decrease to the 0 for almost all C/N_0 .

The impact of low C/N_0 on ROC performance is further explained. The histograms under \mathcal{H}_0 and \mathcal{H}_1 are shown in Fig. 3.9 for all 3 networks under two different C/N_0 . Ideally, one would like to have as few histograms overlapping as possible, such that they can be easily distinguished. Fig. 3.9(d) & 3.9(f) shows the Simple-CNN and MLP networks for $C/N_0=36$ respectively, as the result got from Fig. 3.6(a) for lower C/N_0 of these two networks P_d is too low, that the reason is obvious in histogram plot since two hypotheses completely overlapped, which make the detection almost impossible.

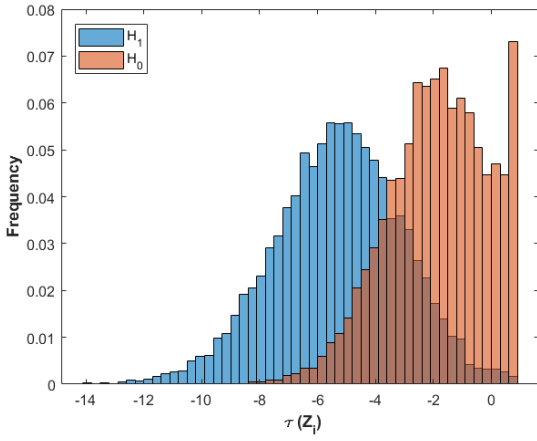
CHAPTER 3. GNSS SIGNAL ACQUISITION IN THE PRESENCE OF SPOOFER



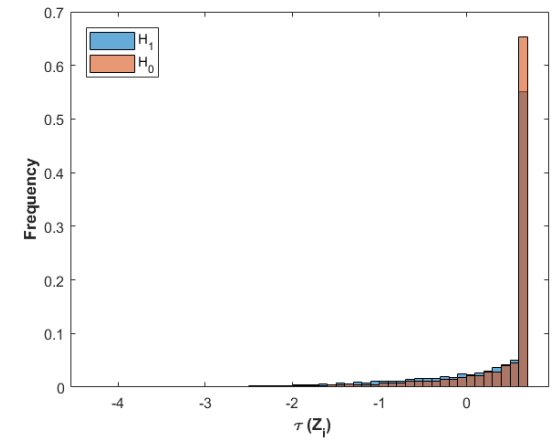
(a) Complex-CNN with $C/N_0 = 45$ dB-Hz



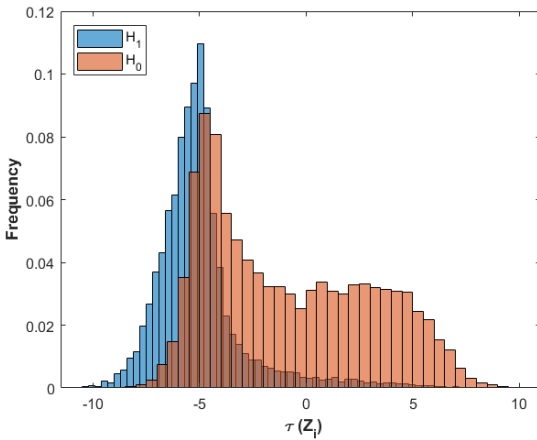
(b) Complex-CNN with $C/N_0 = 36$ dB-Hz



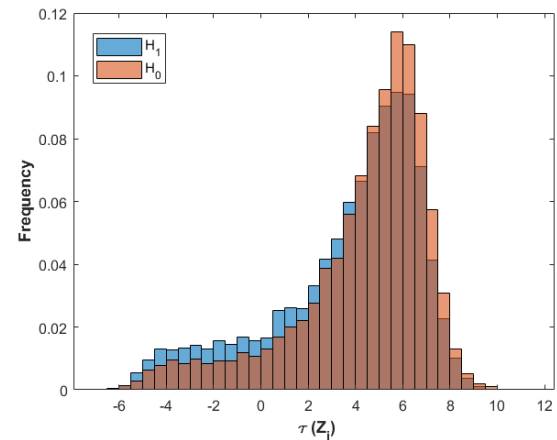
(c) Simple-CNN with $C/N_0 = 45$ dB-Hz



(d) Simple-CNN with $C/N_0 = 36$ dB-Hz



(e) MLP with $C/N_0 = 45$ dB-Hz



(f) MLP with $C/N_0 = 36$ dB-Hz

Figure 3.9: Histogram of $\mathcal{T}(\mathbf{Z}_i)$ under \mathcal{H}_0 and \mathcal{H}_1 hypotheses for each NNs and two different C/N_0 values.

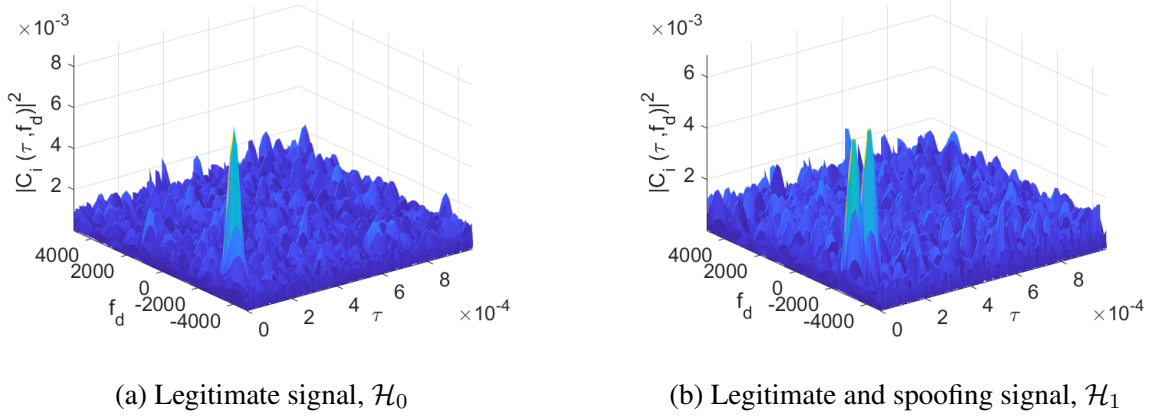


Figure 3.10: CAF evaluation at the delay/Doppler grid with $C/N_0 = 45$ dB-Hz.

3.2 Parallelizable deep learning spoofer detection by CAF splitting and data fusion

This chapter will be explaining the solution when the dataset is too large and the spoofer is present that is not doable with standard method or simple ML/DNN method because it causes high complexity and may not result in a good detection. In order to solve this type of problem, the idea of the split original image into many sub-images will be going to use, but since the spoofer is present it is more complicated than the problem in that the spoofer is absent. In these types of scenarios, we have to look for two or more peaks, and it made it difficult to NN to train and to do the detection since the dataset is large. For this purpose, we split the original CAF image into a sub-image and then apply the fusion dataset which type of non-coherent integration through the fusion classifier, resulting to have probability ratio map with multiple peaks. The next step after getting the probability ratio map is to recognize the number of peaks that appears in the data. To this end, the machine learning algorithm is going to be used to help us to do a clustering method to cluster the data according to their distances and their values. Several machine learning algorithms are available for clustering in this work, the GMM clustering algorithm is going to be used to help to do clustering. The input of the algorithm is the probability ratio map dataset and the output will be the number of the cluster according to their locations and values. Finally, in the end, the corresponding delay/Doppler value of each cluster, which represents a peak, will be calculated.

3.2.1 GNSS signal spoofing effects on acquisition

The spoofer is an interference transmission of a forged GNSS-like signal generated with the purpose of manipulating a victim's receiver's estimated position and time. The spoofer sends a set of false signals that mimic the legitimate satellite signal, except for those parameters that would eventually cause a different position estimate at the receiver unless properly detected. The received GNSS signal, with a spoofing attack, is therefore:

$$y[n] = \sum_{i=1}^M x_i[n; \boldsymbol{\theta}_i] + \sum_{j=1}^{M_s} x_j[n; \boldsymbol{\theta}_{s,j}] + \eta[n] \quad (3.8)$$

where M_s , denotes the number of spoofed signals. In order to deceive the receiver [62], each spoofed signal must have the same spreading code c_i of the satellite it is trying to supersede and broadcast a valid navigation message b_i . The spoofed amplitude, code phases, and carrier phases are gathered in (3.8) by $\boldsymbol{\theta}_{s,j}$ for the j -th spoofer. When building a spoofer detector, two hypotheses are tested:

1. The null hypothesis (\mathcal{H}_0), that the legitimate signal and noise are present, but there is no spoofing signal,

$$\mathcal{H}_0 : y[n] = \sum_{i=1}^M x_i[n; \boldsymbol{\theta}_i] + \eta[n] .$$

2. The alternative hypothesis (\mathcal{H}_1), that the legitimate signal, spoofed signal, and noise are present in the dataset;

$$\mathcal{H}_1 : y[n] = \sum_{i=1}^M x_i[n; \boldsymbol{\theta}_i] + \sum_{i=1}^{M_s} x_{s,i}[n; \boldsymbol{\theta}_{s,i}] + \eta[n]$$

The effect of a spoofing signal on the CAF is well-known and shown in Fig. 3.10 for clarity. Fig. (3.10a) shows an arbitrary CAF under \mathcal{H}_0 Fig. (3.10b) shows the situation when a spoofing signal is present as well, causing the appearance of a secondary peak on the CAF. This work proposes to train a DNN, data-driven model to learn to classify between spoofed or clean signal receptions.

3.2.2 Data-driven GNSS Spoofing detection

The purpose of this work is to design and use a NN in order to recognize the spoofed signal from CAF *images*. Neural networks are models composed of neurons, which are information

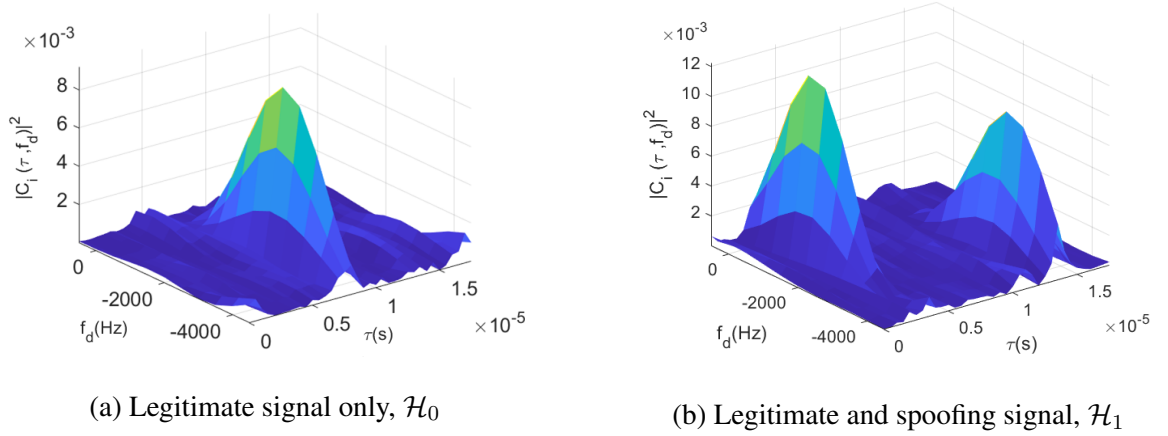


Figure 3.11: Portions of the CAF fed for processing to the NN with $\Delta_m = 18$ and $\Delta_n = 5$ defining the size of the $\{m, n\}$ -th sub-image. The resulting sub-image $\mathbf{Z}_i^{(m,n)}$ is shown on the reduced delay/Doppler grid in the case of (a) absence and (b) presence of a GNSS spoofed signal with $C/N_0 = 45$ dB-Hz.

processing units, for complex data processing. A NN typically contains an input layer, one or more hidden layers, and an output layer, as well as pre-defined activation functions that connect adjacent layers. Each layer has a specific weight, which is usually determined with backpropagation during a training process that involves large amounts of data with known labels [65, 66]. The aforementioned DNN models have the task of classifying CAF maps into spoofed/cleaned cases. We aim at doing that in a probabilistic sense, thus providing probabilities for both hypotheses. The inputs to the NNs are the sub-image that is driven by sliding on CAF evaluated at the delay/Doppler grid, which can be considered as an *image*, which is explained in more detail in 3.2.2.2.

Such images (refer to Fig. 3.11 for an exemplary situation) have certain characteristics that can be used to determine whether the signal is spoofed or not: *i*) in the absence of a spoofing signal, the image should exhibit a single peak corresponding to the legitimate satellite signal (if received with enough power); and *ii*) in the presence of a spoofing signal, the CAF *image* should be composed of at least two peaks in each sub-image, or we have more than one sub-image with a single peak in the CAF *image*. This is used to train a NN model to classify between \mathcal{H}_0 and \mathcal{H}_1 , the hypotheses described earlier. The details of trained NN are explained in 3.2.2.2.

The proposed methodology works on a per-satellite basis. Recall that the input data fed to the NNs is the corresponding CAF image for the satellite, which we denote with $\mathbf{Z}_i \in \mathbb{R}^{n_\tau \times n_f}$ in the sequel. The proposed methodology works on a per-satellite basis. The detail was explained in 2.1.5

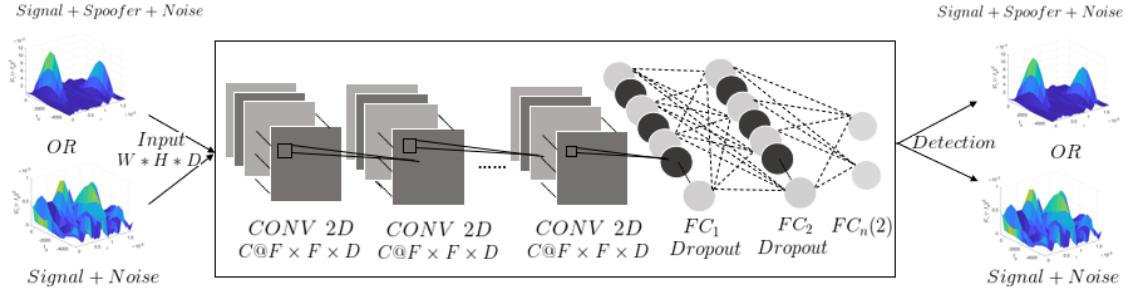


Figure 3.12: Classification of signal (\mathcal{H}_0) or signal (\mathcal{H}_0) and Spoofers (\mathcal{H}_1) in CAFs as part of the proposed GNSS signal acquisition scheme. Particularly, a set of convolutional layers followed by Fully-Connected layers provide the capabilities of deep learning from large datasets.

3.2.2.1 Deep Neural Networks structure

CNNs are one of the most popular models for deep learning, with demonstrated performance in label classification in the context of image datasets. A CNN can have tens or hundreds of layers, where each of these layers learns to identify different features of an image [54, 55]. At each layer, a cascade of filters is applied to input images, whose parameters were previously learned from pairs of known input/output images.

The output of each layer is used as an input to the next layer sequentially. Fig. 3.12 illustrates a CNN structure, as employed in this work. In contrast to other neural networks such as MLPs, CNNs are composed of input, convolutional layers (whereby the image is filtered through convolution with filters learned from the data), several fully-connected hidden layers, and an output layer. During training, the input size of the CNN is fixed, and the input is going through a stack of convolutional layers with the same or different filter sizes. In each convolution layer, the filter sweeps the input image from left to right and up to down by using a stride with 2 pixels size, which is the number of pixels each time the filter shifts. In the end, the convolution layers are followed by FC layers and a final *softmax* layer, which is used for classification purposes and produces the desired class probabilities [55].

The CNN structure is shown in the central box of figure 3.12 which features several convolutions and fully connected layers. Each convolution layer consists of a number of filters (C), with filter size (F) and channel size (D). The ℓ -th convolutional layer transforms its input images from the previous layer with dimensions of $W_{\ell-1} \times H_{\ell-1} \times D_{\ell-1}$ through a set of convolution filters, each of these filters activates certain features from the images and creates an output with dimensions

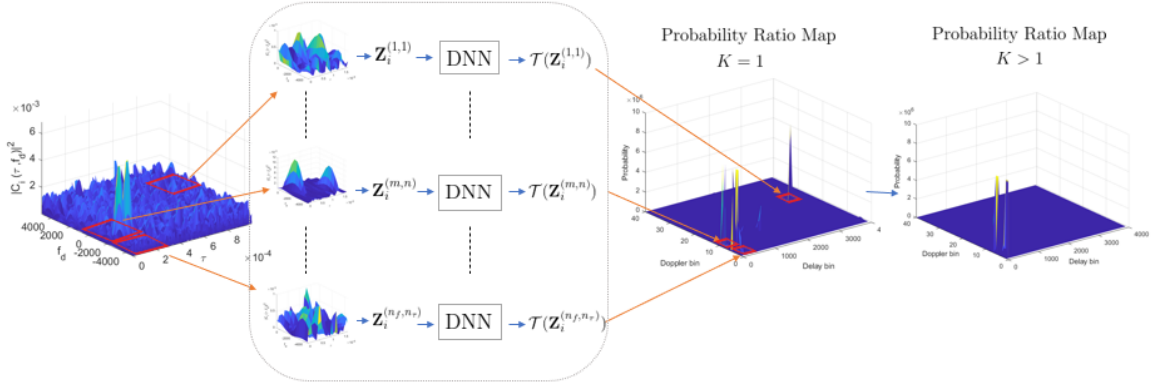


Figure 3.13: Proposed acquisition method where the CAF map Z_i is split into smaller sub-images $Z_i^{(m,n)}$, which are fed to a bank of parallel DNN binary classifiers to produce probability ratio maps. To increase accuracy, several ($K > 1$) probability ratio maps can be non-coherently fused, as shown on the rightmost plot.

of $W_\ell \times H_\ell \times D_\ell$ as an input to the next layer. Notice that initial dimensions are such that $W_0 = n_\tau$ and $H_0 = n_f$, whereas $D_\ell = 1, \forall \ell$ since the data are matrices. After each layer, batch normalization is used to speed up learning, and an activation function is employed before generating the layer output. The number of convolution layers depends on the structure that is used, where the tradeoff is between complexity (reduced number of convolutional layers) and performance (high accuracy). After the last convolution layer, the CNN architecture has a set of fully connected layers in charge of the classification task, and the output of these last layer has the dimensions of the number of classes (two in the case of this article where a binary test is solved in (2.2)) that will be predicted. The output would be the predicted probabilities for each class, as required to compute the test in (2.24).

In this research, the main purpose is to classify the presence/absence of the signal and spoofed signal in CAF maps, as well as estimate accurately the corresponding delay/Doppler parameters in case of the positive detection of the signal and spoofed signal. To achieve the purpose, a CAF map is computed in a dense delay/Doppler grid – as it is common for standard acquisition schemes – which is then fed to the NN model in charge of producing the posterior class probabilities. As a consequence, the input matrix size can be potentially large (i.e., $n_\tau n_f$) which not only might pose a computational complexity limit, but also increase the expense, since the processing device needs a Graphics Processing Units (GPU) with larger memory. In order to alleviate this issue, a sliding scheme is proposed in [51], whereby the large input CAF matrix is scanned using lower dimensional images as the input to the NN classifier.

More precisely, the input dataset image is split into several sub-images, each corresponding to a test delay/Doppler value. The objective is to reduce the initial dimensions W_0 and H_0 , such that the processing is computationally affordable and parallelized. These sub-images are separately fed to multiple (parallelize) NNs that provide the corresponding class probability conditional on a specific delay/Doppler hypothesis or bin. The concept is sketched in Fig. 3.13, where the $\{m, n\}$ -th sub-image corresponds to the correct location of the delay/Doppler. The output of the DNN structure, labeled as $K = 1$ in the plot, is the probability ratio map derived by the Bayesian hypothesis test.

The sub-images are, possibly overlapping, regions of the full CAF map which are centered at a specific delay/Doppler bin hypothesis. While this might increase the overall computational complexity, it enables direct parallelization of the task, which would counteract such complexity. Recall that indices m and n map to the corresponding delay $[\tau]_m$ and Doppler $[f_d]_n$ values, respectively. Therefore, the $\{m, n\}$ -th sub-image for the i -th satellite defined with detail in 2.2.1

It is worth noting that the probability ratio map may contain false peaks, as shown in Fig. 3.13 under $K = 1$. To mitigate those potential false detections, section 3.2.3 describes a methodology to fuse non-coherent integrations of K DNN outputs. The effect of those integrations is depicted in Fig. 3.13 in the rightmost panel for $K = 6$ non-coherent integrations, where the signal probability is accentuated in the correct delay/Doppler bin while false peaks arose from noise are attenuated in the fused probability ratio map.

3.2.2.2 Deep Neural Networks training

This section provides details on how the model was trained. Particularly, we used a realistic GNSS signal simulator to generate I&Q samples from GPS L1 C/A satellites with various parameters according to the training plan described here. In order to increase the detection and localization accuracy, a larger sampling frequency might be desirable, since that accentuates the correlated samples around the CAF peak and helps in increasing the accuracy. However, this has an impact on the number of samples to be processed, and a trade-off needs to be considered. Therefore, here we increased the sampling frequency to 4 MHz, compared to the 2 MHz that were considered in our preliminary work [32]. Increasing the sampling frequency can make the CAF image become high-dimensional if applied directly to a DNN model for classification. That would make the use of DNN more complex and expensive, in that case, the processing device might be required to have a GPU with larger memory to process the GNSS acquisition. In order to reduce the complexity and the expense regardless of increasing f_s , the full CAF image is split, and a sliding DNN scheme is

considered, which is used in [51] to detect the GNSS signal and applied in this work to detect the spoofed attack.

More precisely, a dataset consisting of hundreds of snapshots of GPS L1 C/A, I&Q samples were generated for model training purposes. The dataset consisted of a range of representative carrier-to-Noise-density ratios (C/N_0) varying between 36 to 45 dB-Hz. The length of those snapshots was 1 ms, the duration of a code, such that this constitutes the coherent integration time of the approach. Additionally, the dataset was generated with random delays between 0 to 1 ms and Doppler frequency between -4000 to 4000 Hz. These I&Q samples were then processed to compute the CAF maps over the Doppler-delay grid, which is then split and processed through the DNN model considered in this article. An analogy to images can be made for these CAFs where each Doppler/delay cell is a pixel whose value is that of the CAF, \mathbf{Z}_i . As discussed earlier in Section 3.2.2.1, this can be computationally expensive if a single NN has to process \mathbf{Z}_i entirely. For instance, if 50 Doppler bins are considered (i.e., 200 Hz bins, such that the DNN has more resolution to identify the CAF peak) in generating the CAF for a GPS L1 C/A signal, those images would be 4000×50 dimensional for the f_s considered in this work. Alternatively, if \mathbf{Z}_i is split into smaller images of size 11×36 (read as Doppler \times delay) there are a total of 158600 low-dimensional sub-images to be efficiently processed by the NN, potentially in parallel. The sub-image with the size 11×36 was considered to provide a reasonable trade-off between sub-image size and model complexity, since in this method we consider those sub-images with the complete CAF peak exactly in the middle of the sub-image. Considering the size of the sub-images smaller than the current size might cause issues where the CAF peak might not be included in any of the sub-images. Larger sub-image sizes would cause multiple peaks to appear and higher computational complexity.

An interpretation of the sliding concept is similar to the convolutional layers in a CNN, with a stride of one sample, whereby the entire CAF is scanned in smaller windows that can contain the main signal peak and the spoofed signals. These peaks, in contrast to peaks generated by random noise, show a correlation in the delay and (more noticeable) in the Doppler domains that can be exploited by the NN classifier.

In order to train the NN-based classifier, the generated dataset contained either spoofed signal and signal-plus-noise (\mathcal{H}_1) or signal-plus-noise (\mathcal{H}_0) snapshots, which were then split into sub-images as shown in Fig. 3.11. Then, these types of snapshots are fed to NN for training. The classifier learned its parameters by observing a set of 3000 input/output pairs in a supervised manner. The output of the NN was a *softmax* layer with dropout, such that the resulting outcome of the NN is the binary class probabilities required to compute the test in (2.27) or its non-coherent version,

which is explained in [51] and section 3.2.3.

3.2.3 Probabilistic signal detection

Coherent integration of long code sequences can be implemented in computing the CAF map, $\mathcal{C}_i(\cdot, \cdot)$, in the usual manner. In implementing non-coherent integration, an alternative is to fuse the multiple probability ratio maps resulting from processing CAF images through the NN architecture described earlier in Section 3.2.2.1. We denote by $K \in \mathbb{Z}^+$ the total number of non-coherent integrations. This section discusses the data fusion of such multiple classifiers, which was first introduced in [51] and quickly reviewed here. It is known that increasing integration time (both coherently and non-coherently) improves the overall detection performance of the acquisition process, this same rationale holds in the case of the data-driven classifier proposed here, whereby non-coherent integrations (i.e. fusion of multiple classifiers solutions) improves the reliability of the so-called probability maps (i.e. by attenuating falsely detected peaks or enhancing locations where actual signals reside).

A qualitative example of how the fusion rule impacts the performance of the classifier is provided in Fig. 3.14. On the one hand, Fig. 3.14a shows the CAF delay/Doppler map used in standard signal acquisition in a presence of spoofer without any non-coherent integration and just 1 ms coherent integration. It can be seen, as it is known from GNSS literature, that outside the true peak (denoted with a red circle) the noise floor is relatively spiky and can cause substantial false alarms, particularly at low C/N_0 values. On the other hand, the proposed data-driven method takes the CAF values and processes them to produce the so-called probability ratio maps, as defined on the right-hand side of (2.31). The probability ratio map resulting from processing the CAF in Fig. 3.14a can be observed in Fig. 3.14b where it is clear that the variability in the noise floor has been reduced, although residual spikes can still be detected at delay/Doppler bins where no signal was present. This effect is smoothed further with the fusion method, as shown in Fig. 3.14c where $K = 6$ non-coherent integrations were considered. Notice that the NN uses sub-images as inputs to produce a class probability pair, as depicted in Fig. 3.11. As a consequence, the posterior probabilities are taking into consideration the delay/Doppler correlations of the CAF around the signal peak, in contrast to the standard method which only considers the maximum value of the CAF thus neglecting the waveform arising from the noise form (i.e. the autocorrelation function of the corresponding spreading code).

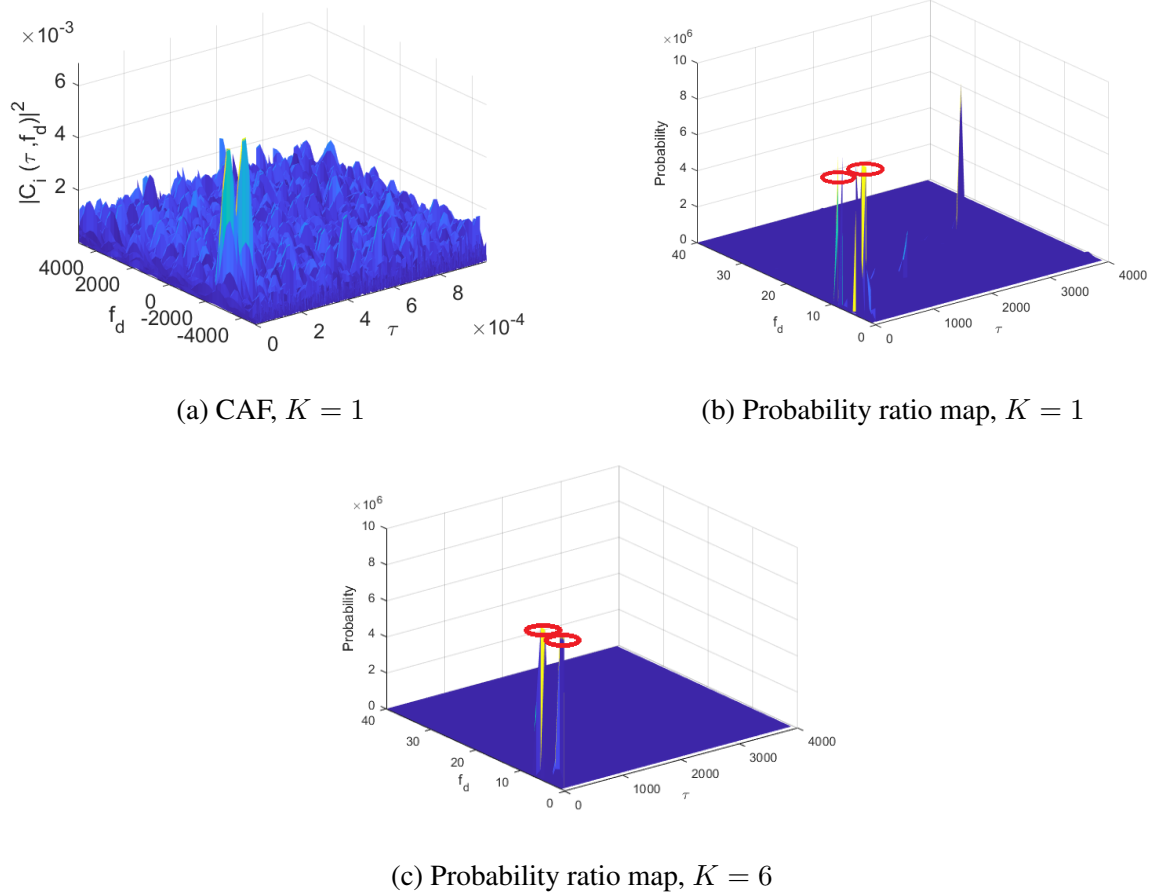


Figure 3.14: Comparison of the delay/Doppler grid in a presence of spoofer for (a) standard CAF map with coherent integration only, (b) probability map produced by the data-driven classifier with coherent integration only, and (c) probability map after fusing $K = 6$ non-coherently classifier outputs. The GNSS signal had a C/N_0 of 42 dB-Hz and the red circle highlights the location of the peak generated by the GNSS signal.

3.2.4 Estimating number of signals

As opposed to the case in [51], this paper deals with the possible situation where multiple signals might be present in the data. To estimate that number of signals (that is, the legitimate signal and an arbitrary number of spoofers), the proposed DNN scheme is connected to a GMM based clustering method. In particular, the probability ratio map produced by the DNN is fed into the GMM, which is in charge of determining the number of peaks (i.e. signals) and their location in order to approximate that probability map as a linear combination of Gaussian functions.

GMM consists of a linear superposition of Gaussian components, providing a richer class of density models than single Gaussian models [68], particularly relevant in the case at hand where the probability map is known to be multimodal in the presence of spoofers. The Gaussian mixture distribution for the problem at hand can then be written as:

$$p(\boldsymbol{\psi}) = \sum_{\ell=1}^L \mathcal{N}(\boldsymbol{\psi} | \boldsymbol{\mu}_\ell, \boldsymbol{\Sigma}_\ell) \quad (3.9)$$

where $\boldsymbol{\psi} = (\tau, f_d)^\top$ is a two-dimensional vector with the delay/Doppler values at which the probability map $\mathcal{T}(\mathbf{Z}_i)$ is evaluated. L determines the number of components in the mixture, which in this case represents an estimate of the number of spoofing signals plus legitimate signal. The parameters $\{\boldsymbol{\mu}_\ell, \boldsymbol{\Sigma}_\ell\}_{\ell=1}^L$ denote, respectively, the mean and covariance of each of the L Gaussian in the mixture.

The assumption is that the satellite signal is always observed, that is, when the spoofing signal is absent, $L = 1$ and $\boldsymbol{\psi}$ is Gaussian distributed; and when the spoofing signal occurs, $L > 1$ and $\boldsymbol{\psi}$ would be a Gaussian mixture. Given the observed data, an Expectation-Maximization (EM) algorithm is used in order to learn the parameters L and $\{\boldsymbol{\mu}_\ell, \boldsymbol{\Sigma}_\ell\}_{\ell=1}^L$. In order to compare different mixture complexities (i.e. values of L), the Bayesian Information Criterion (BIC) is employed. BIC is a popular criterion for model selection among a finite set of models, a model with lower BIC is generally preferred. It is defined as:

$$\text{BIC} = m \ln n - 2 \ln(\hat{\mathcal{L}}) \quad (3.10)$$

where \mathcal{L} is the marginal likelihood of the model;

n is the sample size, which in this work it is a number of the point that passed the threshold; m is the number of parameters estimated by the model, being $m = 6K$ in our case. By fitting different GMMs with varying L values using the EM algorithm, BIC can be used to measure the

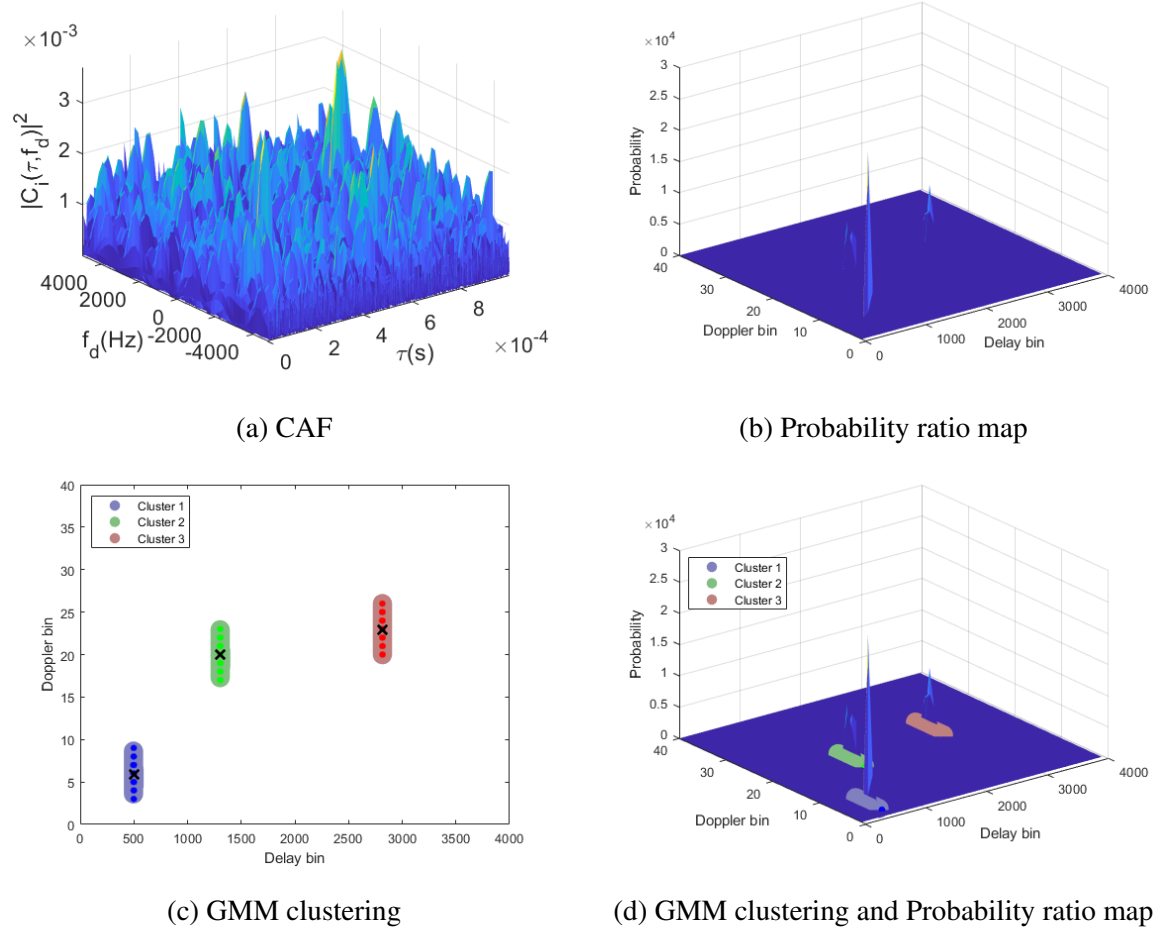


Figure 3.15: Running example showing the process followed by the proposed algorithm. The experiment consists of a legitimate signal and two spoofers with $C/N_0 = 42$ dB-Hz. The various panels show the corresponding (a) CAF; (b) probability ratio map; (c) top view of the threshold probability ratio maps, after clustering is applied; and (d) three-dimensional perspective of the latter with the probability ratio maps overlaid.

performance of each and assess their modeling capabilities, ultimately useful in estimating the number of signals given the observed data.

Fig. 3.15 explains the process that is followed by the GMM clustering method in order to detect and localize the signal and spoofing signals. In particular, a running example where three signals (i.e. one legitimate and two spoofers) are present in the dataset. Initially, the proposed method uses the computed CAF delay/Doppler map, shown in Fig. 3.15a, to feed a DNN classifier. The DNN model is in charge of producing probability ratio maps, used to determine the presence or absence of signals in the map. This model can operate with coherent processing of codes (e.g. 1 ms long codes for GPS L1 C/A signals) or non-coherently over K coherently computed CAFs. After the DNN process, the probability ratio map is obtained, as shown in Fig. 3.15b. The data in Fig. 3.15b is thresholded and then fed to the GMM clustering algorithm, which uses BIC to decide the number of clusters L . As shown in Fig. 3.15c and Fig. 3.15d, it can be seen how the clustering algorithm is capable to associate the probability values corresponding to each of the signals together. In addition, notice that the resulting estimate of the mean of each mode in the mixture (that is, μ_ℓ) is also an estimate of the delay/Doppler for that signal.

3.2.5 Results

The proposed data-driven spoofer detection scheme was validated using synthetic data. Particularly, a first set of experiments were performed to evaluate the ROC plots of the proposed classifier for the case of one spoofing signal. In these experiments, different C/N_0 values were considered and the DNN model training considered CAF images produced by 1 ms coherent correlation process without non-coherent integration times, as done in [51]. In training, the use of $K = 6$ DNN outputs was fused according to the methodology described earlier.

Fig. 3.16 shows the ROC results for the proposed method (dashed lines). These are compared to the theoretical performance of the standard method (solid lines), although in that case, the theoretical results are for the case of detecting a single signal in noise for which this result is available [6], while it is not the case for multiple signals detection. The result shows that for low C/N_0 values, the proposed scheme performs poorly, mostly caused by the CAF peaks being too weak for the DNN to discern from noise. However, when the C/N_0 values are increased, the CAF peaks become higher in the sub-images and the proposed method eventually outperforms the standard methods since it can be distinguished the difference between signal/spoofers and noise.

The probabilities of false alarm and detection for all C/N_0 corresponding to ROC in Fig.

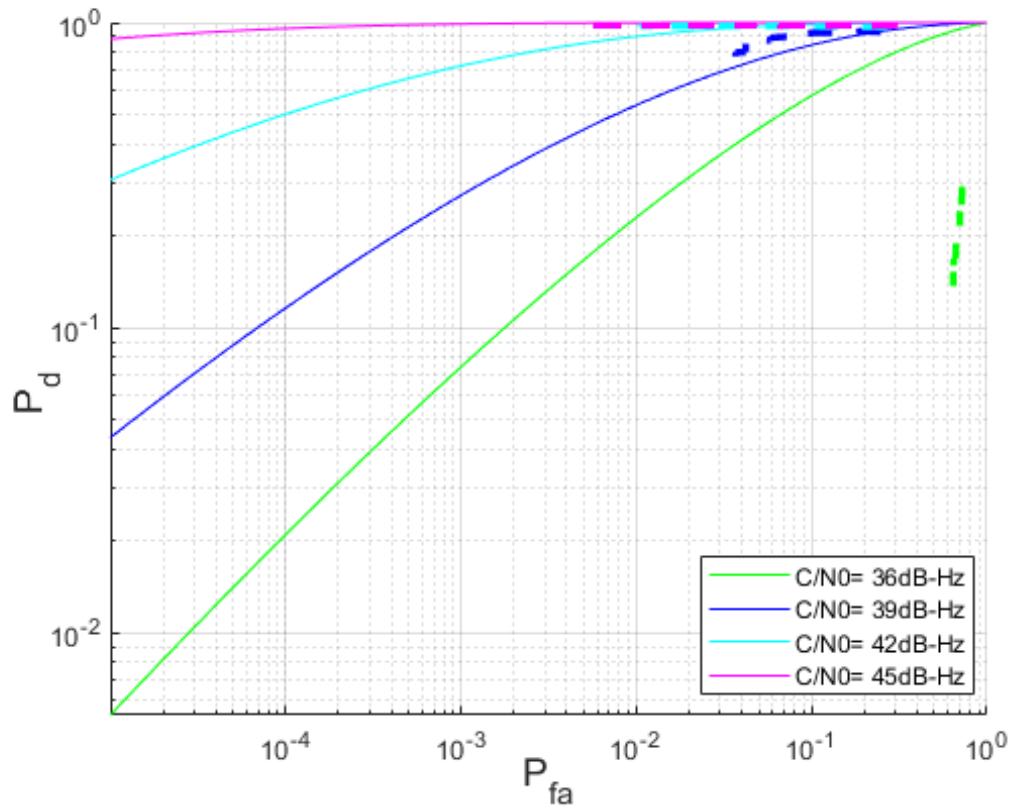


Figure 3.16: ROC curves for a 1 ms coherently integrated snapshot and $K = 6$ non-coherently processed blocks for a variety of C/N_0 values for signal and spoofer. The performance of the proposed scheme (dashed lines) is compared to the theoretical performance of standard methods (solid lines).

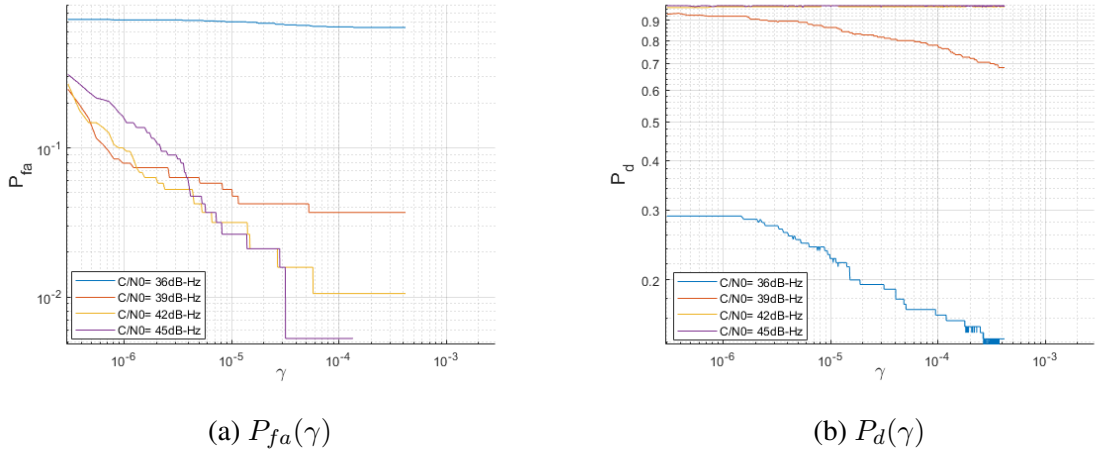


Figure 3.17: (a) $P_d(\gamma)$ and (b) $P_{fa}(\gamma)$ probabilities for a 1 ms coherently integrated snapshot, $K = 6$ non-coherent processing, and a variety of C/N_0 values when signal and Spoofers are present.

3.16 are shown in Fig. 3.17. As it is shown in Fig. 3.17a for the lowest C/N_0 the probability of a false alarm is high because the NN detects the noise as a signal/spoofers, while for other C/N_0 values increasing the threshold (γ) decreases the probability of false alarm. Additionally, the probability of detection is depicted in Fig. 3.17b where a similar behavior can be observed.

Another relevant experiment that was performed with the DNN (plus GMM) scheme proposed in this paper was to evaluate its performance for different separations of the signal and spoofing signals. The objective was to assess the impact on both the DNN probability ratio maps and the GMM-based clustering method. It is indeed relevant to understand, how close signals can be before the detector starts degrading. Fig. 3.18 shows the probability of the detection according to the relative delay ($\Delta\tau$) for three different Doppler bins. The red solid line with ellipses represented the case where the two signals have the same Doppler frequency; the green solid line with circles reports the results when they have 1 Doppler bin separation; and the blue solid line with stars represents the case when they are two Doppler bins separated. As it is expected, by decreasing the delta delays the probability of detection decreases as well. Similarly, when they are in the same Doppler bin, they have the worst probability of detection as they cannot be distinguished as they get closer.

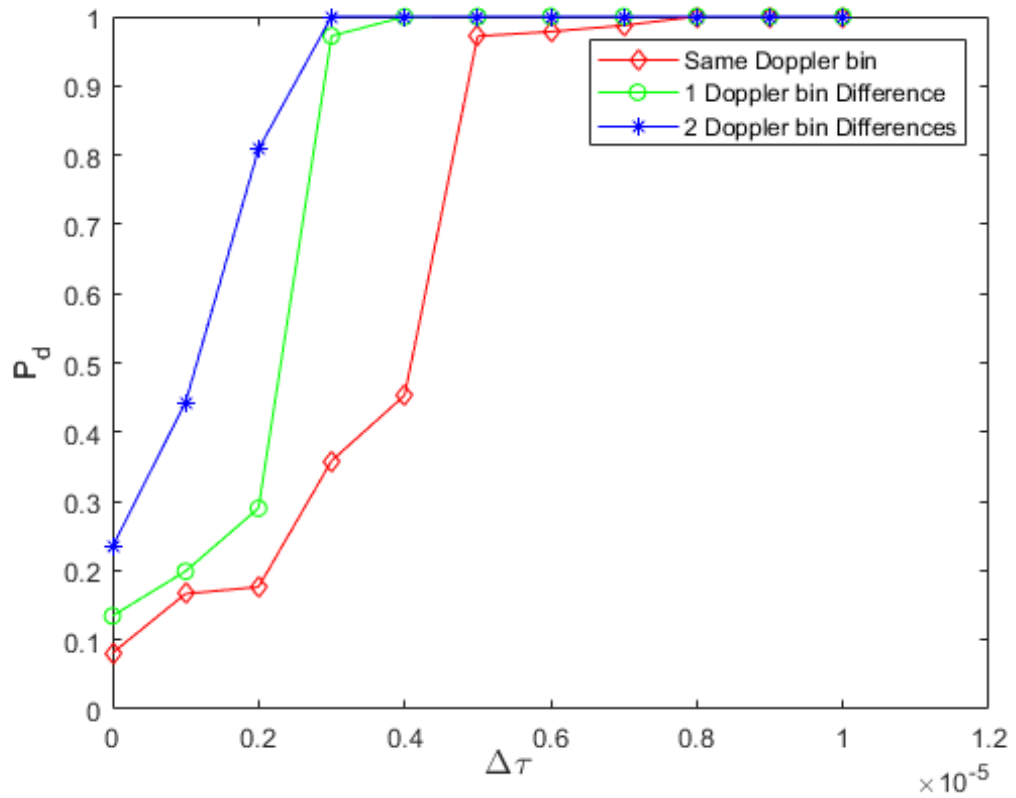


Figure 3.18: Probability of correctly detecting both legitimate and spoofing signals, as well as determining their number, as a function of their relative delay separation. Also, the same Doppler, one bin, and two bin separations were considered. Bin size being 500 Hz.

Chapter 4

Conclusion

In this work, the use of machine learning models (also known as neural networks, NNs) to learn the classification task of detecting signals from GNSS satellites and the use of data-driven models in the signal acquisition stage of the receiver investigated by addressing a classification problem from Cross Ambiguity Function delay/Doppler maps. It proposed to use deep learning models to perform such acquisition, whereby the CAF is fed to a data-driven classifier that outputs binary class posteriors. The class posteriors are used to compute a Bayesian hypothesis test to statistically decide the presence or absence of a GNSS signal. Traditional schemes based on correlation are optimal, and thus NNs are not expected to outperform those in nominal conditions. The use of NNs to substitute traditional approaches could bring benefits when nominal physics-based models do not hold. This work presented a proof-of-concept validating the use of NNs for detection purposes in GNSS. The detection results in nominal situations are compared to the theoretical bound in the receiver operating characteristic (ROC) plots. Two NNs architectures, MLP and CNN were discussed, out of which MLP seemed to provide slightly better results, presumably due to the simplicity of the classification task.

Then it represented a DNN approach to detect spoofing attacks. Spoofing attacks are, in general, difficult to model and counteract [69]. In those situations, data-driven schemes become useful if enough training data is available. This work explores this approach using the CAF delay/Doppler map as an input to a DNN for classification purposes. Particularly, several neural network models are trained, and their performance is compared in terms of detection and false alarm probabilities. Results show promising performances, particularly with more complex NNs, able to capture the nature of spoofing attacks and their impact on CAF maps. The reason is that complex CNN has more convolution layers, each layer has a filter that extracts a specific feature of the data, which results to

CHAPTER 4. CONCLUSION

extract more features and increase the detection accuracy.

This work addressed the problem of using a large dataset with DNN, based on our knowledge, which is not doable with previous ideas since it increases the complexity with a high computational process. The versatility and computational affordability of the proposed method are addressed by splitting the CAF, which enables the flexible use of the method on CAFs of different dimensions (depending on the delay/Doppler bin sizes) into smaller overlapping sections, which are fed to a bank of parallel classifiers whose probabilistic results are optimally fused to provide a so-called probability ratio map from which acquisition is decided. Additionally, the research shows how non-coherent integration schemes are enabled through optimal data fusion, with the goal of increasing the resulting classifier accuracy.

The simulation results show that the proposed data-driven method outperforms current CAF maximization strategies, enabling enhanced acquisition at lower carrier-to-noise density ratios. It is shown that the deep learning method can outperform standard approaches, even exceeding their fundamental limits. This result can be explained by the fact that standard methods are based on the bin maximization of the CAF, whereas the proposed data-driven method exploits the correlation across neighboring bins. Additionally, an optimal fusion rule is provided in order to extend the methodology to non-coherent integration, which is also seen to improve the overall classification performance.

The use of deep neural networks as a method to detect GNSS spoofing attacks for a large dataset is investigated. In the case of spoofing, the situation is slightly more challenging, which makes the use of deep learning models more relevant. In addition to efficiently implementing the data-driven classifier through an image-splitting process (enabling parallelization), the research considers a Gaussian mixture model approach to determine the number of spoofing signals. Results show that the proposed deep learning method can outperform current approaches, especially in the moderate to high signal-to-noise ratios.

Future works will revolve around improving the results in terms of lower false alarm probability while keeping detection probabilities large in the presence of the spoofer. Also, the Texas Spoofing test Battery (TEXBAT) the real dataset can be used with the proposed method to compare its results with the simulated dataset results. Another research area can be investigating the DNN methods for multipath mitigation strategies in the presence and absence of the spoofer.

Bibliography

- [1] D. Dardari, E. Falletti, and M. Luise, *Satellite and terrestrial radio positioning techniques: a signal processing perspective*. Academic Press, 2011.
- [2] M. G. Amin, P. Closas, A. Broumandan, and J. L. Volakis, “Vulnerabilities, threats, and authentication in satellite-based navigation systems [scanning the issue],” *Proceedings of the IEEE*, vol. 104, no. 6, pp. 1169–1173, 2016.
- [3] D. Dardari, P. Closas, and P. M. Djurić, “Indoor tracking: Theory, methods, and technologies,” *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, pp. 1263–1278, 2015.
- [4] Y. J. Morton, F. van Diggelen, J. J. Spilker Jr, B. W. Parkinson, S. Lo, and G. Gao, *Position, Navigation, and Timing Technologies in the 21st Century: Integrated Satellite Navigation, Sensor Systems, and Civil Applications*. John Wiley & Sons, 2021.
- [5] N. Williams, P. B. Darian, G. Wu, P. Closas, and M. Barth, “Impact of Positioning Uncertainty on Connected and Automated Vehicle Applications,” *SAE International Journal of Connected and Automated Vehicles*, vol. 6, no. 12-06-02-0010, 2022.
- [6] D. Borio, “A statistical theory for GNSS signal acquisition,” *PhD Dissertation Polytecnico di Torino*, 2008.
- [7] E. Kaplan and C. Hegarty, *Understanding GPS: principles and applications*. Artech house, 2005.
- [8] P. Misra and P. Enge, “Global Positioning System: signals, measurements and performance second edition,” *Global Positioning System: Signals, Measurements And Performance Second Editions*, 2006.

BIBLIOGRAPHY

- [9] J. B. Tsui, *Fundamentals of global positioning system receivers: a software approach*. John Wiley & Sons, 2005, vol. 173.
- [10] A. Lehner and A. Steingass, “A novel channel model for land mobile satellite navigation,” in *Institute of Navigation Conference ION GNSS*, 2005, pp. 13–16.
- [11] D. Borio and P. Closas, “A fresh look at GNSS anti-jamming,” *Inside GNSS*, vol. 12, pp. 54–61, 2017.
- [12] D. Borio, “Robust signal processing for GNSS,” in *Proc. of the 2017 European Navigation Conference (ENC)*, Lousanne, Switzerland, May 2017, pp. 150–158.
- [13] —, “Myriad Non-Linearity for GNSS Robust Signal Processing,” *IET Radar Sonar and Navigation*, vol. 11, no. 10, pp. 1467–1476, Oct. 2017. [Online]. Available: <http://digital-library.theiet.org/content/journals/10.1049/iet-rsn.2016.0610>
- [14] D. Borio and P. Closas, “Complex Signum Non-Linearity for Robust GNSS Signal Mitigation,” *IET Radar Sonar and Navigation*, pp. 1–10, Apr. 2018. [Online]. Available: <http://digital-library.theiet.org/content/journals/10.1049/iet-rsn.2017.0552>
- [15] D. Borio, H. Li, and P. Closas, “Huber’s Non-Linearity for GNSS Interference Mitigation,” *Sensors*, vol. 18, no. 7, p. 2217, 2018.
- [16] D. Borio and P. Closas, “Robust Transform Domain Signal Processing for GNSS,” *Navigation*, 2019.
- [17] H. Li, D. Borio, and P. Closas, “Dual-Domain Robust GNSS Interference Mitigation,” in *Proceedings of the International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2019)*, Miami, FL, 16-20 Sept 2019.
- [18] D. Borio and C. Gioia, “GNSS interference mitigation: A measurement and position domain assessment,” *NAVIGATION, Journal of the Institute of Navigation*, vol. 68, no. 1, pp. 93–114, 2021.
- [19] P. Borhani-Darian and P. Closas, “Deep Neural Network Approach to GNSS Signal Acquisition,” in *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, 2020, pp. 1214–1223.
- [20] F. Dovis, *GNSS interference threats and countermeasures*. Artech House, 2015.

BIBLIOGRAPHY

- [21] A. Siemuri, H. Kuusniemi, M. Elmusrati, P. Välisuo, and A. Shamsuzzoha, “Machine Learning Utilization in GNSS—Use Cases, Challenges and Future Applications,” in *2021 International Conference on Localization and GNSS (ICL-GNSS)*. IEEE, 2021, pp. 1–6.
- [22] A. A. Abdallah and Z. M. Kassas, “Deep learning-aided spatial discrimination for multipath mitigation,” in *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)*. IEEE, 2020, pp. 1324–1335.
- [23] G. Zhang, P. Xu, H. Xu, and L.-T. Hsu, “Prediction on the Urban GNSS Measurement Uncertainty Based on Deep Learning Networks With Long Short-Term Memory,” *IEEE Sensors Journal*, vol. 21, no. 18, pp. 20 563–20 577, 2021.
- [24] P. Huang, C. Rizos, and C. Roberts, “Satellite selection with an end-to-end deep learning network,” *GPS Solutions*, vol. 22, no. 4, pp. 1–12, 2018.
- [25] H. Li, P. Borhani-Darian, P. Wu, and P. Closas, “Deep Learning of GNSS Signal Correlation,” in *Proceedings of the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2020)*, 2020, pp. 2836–2847.
- [26] ———, “Deep neural network correlators for GNSS multipath mitigation,” *IEEE Transactions on Aerospace and Electronic Systems*, 2022.
- [27] C. Savas and F. Dosis, “Multipath Detection based on K-means Clustering,” in *Proceedings of the 32nd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2019)*, 2019, pp. 3801–3811.
- [28] T. Suzuki, K. Kusama, and Y. Amano, “NLOS Multipath Detection using Convolutional Neural Network,” in *Proceedings of the 33rd International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+ 2020)*, 2020, pp. 2989–3000.
- [29] E. Munin, A. Blais, and N. Couellan, “Convolutional neural network for multipath detection in GNSS receivers,” in *2020 International Conference on Artificial Intelligence and Data Analytics for Air Transportation (AIDA-AT)*. IEEE, 2020, pp. 1–10.
- [30] G. Caparra, P. Zoccarato, and F. Melman, “Machine Learning Correction for Improved PVT Accuracy,” in *Proceedings of the 34th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2021)*, 2021, pp. 3392–3401.

BIBLIOGRAPHY

- [31] M. R. Manesh, J. Kenney, W. C. Hu, V. K. Devabhaktuni, and N. Kaabouch, "Detection of GPS spoofing attacks on unmanned aerial systems," in *2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2019, pp. 1–6.
- [32] P. Borhani-Darian, H. Li, P. Wu, and P. Closas, "Deep Neural Network Approach to Detect GNSS Spoofing Attacks," in *Proceedings of the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2020)*, 2020, pp. 3241–3252.
- [33] S. Tohidi and M. R. Mosavi, "Effective detection of GNSS spoofing attack Using A multi-layer perceptron neural network classifier trained by PSO," in *2020 25th International Computer Conference, Computer Society of Iran (CSICC)*. IEEE, 2020, pp. 1–5.
- [34] R. Calvo-Palomino, A. Bhattacharya, G. Bovet, and D. Giustiniano, "Short: LSTM-based GNSS Spoofing Detection Using Low-cost Spectrum Sensors," in *2020 IEEE 21st International Symposium on "A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*. IEEE, 2020, pp. 273–276.
- [35] S. Semanjski, A. Muls, I. Semanjski, and W. De Wilde, "Use and validation of supervised machine learning approach for detection of GNSS signal spoofing," in *2019 International Conference on Localization and GNSS (ICL-GNSS)*. IEEE, 2019, pp. 1–6.
- [36] R. Morales Ferre, A. de la Fuente, and E. S. Lohan, "Jammer classification in GNSS bands via machine learning algorithms," *Sensors*, vol. 19, no. 22, p. 4841, 2019.
- [37] A. Louis and M. Raimondi, "Neural Network based Evil WaveForms Detection," in *Proceedings of the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2020)*, 2020, pp. 1984–1989.
- [38] D. Brum, M. R. Veronez, E. M. de Souza, I. É. Koch, L. Gonzaga, I. Klein, M. T. Matsuoka, V. F. Rofatto, A. M. Junior, G. E. dos Reis Racolte *et al.*, "A Proposed Earthquake Warning System Based on Ionospheric Anomalies Derived From GNSS Measurements and Artificial Neural Networks," in *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2019, pp. 9295–9298.
- [39] M. Alshaye, F. Alawwad, and I. Elshafey, "Hurricane tracking using Multi-GNSS-R and deep learning," in *2020 3rd International Conference on Computer Applications & Information Security (ICCAIS)*. IEEE, 2020, pp. 1–4.

BIBLIOGRAPHY

- [40] Q. Yan and W. Huang, "Sea ice sensing from GNSS-R data using convolutional neural networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 10, pp. 1510–1514, 2018.
- [41] N. Linty, A. Farasin, A. Favenza, and F. Dovis, "Detection of GNSS ionospheric scintillations based on machine learning decision tree," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 55, no. 1, pp. 303–317, 2018.
- [42] Y. Liu, Y. Morton, and Y. Jiao, "Application of Machine Learning to Characterization of GPS L1 Ionospheric Amplitude Scintillation," in *2018 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, 2018, pp. 1159–1166.
- [43] M. O. Selbesoglu, "Prediction of tropospheric wet delay by an artificial neural network model based on meteorological and GNSS data," *Engineering Science and Technology, an International Journal*, vol. 23, no. 5, pp. 967–972, 2020.
- [44] J. Vilà-Valls, N. Linty, P. Closas, F. Dovis, and J. T. Curran, "Survey on signal processing for GNSS under ionospheric scintillation: Detection, monitoring, and mitigation," *NAVIGATION, Journal of the Institute of Navigation*, vol. 67, no. 3, pp. 511–536, 2020.
- [45] M. Sun, Y. Qin, J. Bao, and X. Yu, "Gps spoofing detection based on decision fusion with a k-out-of-n rule," *IJ Network Security*, vol. 19, no. 5, pp. 670–674, 2017.
- [46] E. Shafiee, M. Mosavi, and M. Moazedi, "Detection of spoofing attack using machine learning based on multi-layer neural network in single-frequency gps receivers," *The Journal of Navigation*, vol. 71, no. 1, pp. 169–188, 2018.
- [47] S. Semanjski, I. Semanjski, W. De Wilde, and S. Gautama, "Use of supervised machine learning for gnss signal spoofing detection with validation on real-world meaconing and spoofing data—part ii," *Sensors*, vol. 20, no. 7, p. 1806, 2020.
- [48] J. Li, X. Zhu, M. Ouyang, W. Li, Z. Chen, and Q. Fu, "Gnss spoofing jamming detection based on generative adversarial network," *IEEE Sensors Journal*, vol. 21, no. 20, pp. 22 823–22 832, 2021.
- [49] H. Mathis, P. Flammant, and A. Thiel, "An analytic way to optimize the detector of a post-correlation FFT acquisition algorithm," *Quadrature*, vol. 1000, p. 1, 2003.
- [50] A. Whalen, *Detection of signals in noise*. Academic press, 2013.

BIBLIOGRAPHY

- [51] P. Borhani-Darian, H. Li, P. Wu, and P. Closas, “Deep Learning of GNSS Acquisition,” *Sensors*, vol. 23, no. 3, p. 1566, 2023.
- [52] T. O’Shea, T. Roy, and T. Clancy, “Over-the-air deep learning based radio signal classification,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 168–179, 2018.
- [53] Q. Yan, W. Huang, and C. Moloney, “Neural networks based sea ice detection and concentration retrieval from GNSS-R delay-Doppler maps,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 8, pp. 3789–3798, 2017.
- [54] S. Liu and W. Deng, “Very deep convolutional neural network based image classification using small training sample size,” in *2015 3rd IAPR Asian conference on pattern recognition (ACPR)*. IEEE, 2015, pp. 730–734.
- [55] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [56] Activation functions in neural networks. <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>.
- [57] Mathworks, “Create Simple Deep Learning Network for Classification,” <https://www.mathworks.com/help/deeplearning/examples/create-simple-deep-learning-network-for-classification.html>.
- [58] F. Pastor, J. García-González, J. M. Gandarias, D. Medina, P. Closas, A. J. García-Cerezo, and J. M. Gómez-de Gabriel, “Bayesian and Neural Inference on LSTM-Based Object Recognition from Tactile and Kinesthetic Information,” *IEEE Robotics and Automation Letters*, vol. 6, no. 1, pp. 231–238, 2020.
- [59] E. Kaplan and C. Hegarty, *Understanding GPS: principles and applications*. Artech house, 2005.
- [60] P. Misra and P. Enge, “Global positioning system: signals, measurements and performance second edition,” *Global Positioning System: Signals, Measurements And Performance Second Editions*, vol. 206, 2006.
- [61] J. B.-Y. Tsui, *Fundamentals of global positioning system receivers: a software approach*. John Wiley & Sons, 2005, vol. 173.

BIBLIOGRAPHY

- [62] M. L. Psiaki and T. E. Humphreys, “GNSS spoofing and detection,” *Proceedings of the IEEE*, vol. 104, no. 6, pp. 1258–1270, 2016.
- [63] H. Mathis, P. Flammant, and A. Thiel, “An analytic way to optimize the detector of a post-correlation fft acquisition algorithm,” *Quadrature*, vol. 1000, p. 1, 2003.
- [64] A. D. Whalen, *Detection of signals in noise*. Academic press, 2013.
- [65] T. O’Shea, T. Roy, and T. Clancy, “Over-the-air deep learning based radio signal classification,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 168–179, 2018.
- [66] Q. Yan, W. Huang, and C. Moloney, “Neural networks based sea ice detection and concentration retrieval from GNSS-R delay-Doppler maps,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 8, pp. 3789–3798, 2017.
- [67] S. Sharma, “Activation functions in neural networks,” *Towards Data Science*, vol. 6, 2017.
- [68] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*. Springer, 2006, vol. 4.
- [69] P. Closas, J. Arribas, and C. Fernández-Prades, “Spoofing detection by a reduced acquisition process,” in *Proceedings of the Precise Time and Time Interval Systems and Applications Meeting (ION PTTI 2016)*, 2016.