Universidad de Málaga

Escuela Técnica Superior de
Ingeniería de Telecomunicación

TRABAJO FIN DE MÁSTER

A Machine Learning based NLOS Detector for Robust GNSS Positioning in Urban Environments

Máster en Ingeniería
de Telecomunicaciones

Juan Carlos Ruiz Sicilia
Málaga, 2021

This master thesis has been developed at the Institute of Communications and Navigation of the German Aerospace Center (DLR) in Munich, under the supervision of Omar García Crespillo.

Este trabajo fin de máster se ha desarrollado en el Instituto de Comunicaciones y Navegación del Centro Aeroespacial Alemán (DLR) en Munich, bajo la supervisión de Omar García Crespillo.

Universidad de Málaga
E.T.S. de Ingeniería de Telecomunicación

Detector de Señales NLOS basado en Machine Learning para Posicionamiento GNSS Robusto en Entornos Urbanos

Autor: Juan Carlos Ruiz Sicilia

Tutor: Mari Carmen Aguayo Torres

Departamento: Ingeniería de Comunicaciones (IC)

Titulación: Máster en Ingenieria de Telecomunicaciones

Palabras clave: GNSS, Machine Learning, NLOS, Reliability

Resumen

En la actualidad, garantizar la fiabilidad es uno de los principales retos en el posicionamiento GNSS, especialmente, en entornos urbanos donde se pueden presentar múltiples amenazas, como las señales multi-camino o sin visión directa (NLOS). Estas fuentes de error son complejas de modelar y aislar y, por tanto, de predecir o detectar. Para afrontar este problema, se han empezado a utilizar algoritmos de aprendizaje automático (ML) como solución para manejar esta complejidad debido a su potencia para encontrar relaciones entre descriptores. Sin embargo, actualmente el estado del arte es muy limitado y se basa en descriptores dudosos. Además, los algoritmos propuestos dependen en gran medida de los datos de entrenamiento y, por lo tanto, pueden no ser robustos frente a variaciones en el escenario o la instalación.

En este trabajo se presenta un detector de LOS/NLOS basado en regresiones logísticas. El algoritmo propuesto incluye descriptores obtenidos a partir de señales a varias frecuencias. Estos descriptores son procesados siguiendo una metodología para que el detector sea menos dependiente de la antena y el receptor específicos y de los datos utilizados para el entrenamiento. Además, se ha incorporado el detector en un estimador de Posición, Velocidad y Tiempo (PVT). Los resultados muestran que el detector propuesto presenta ventajas significativas respecto a la literatura, y sus predicciones reducen el error de posicionamiento final.

Universidad de Málaga
E.T.S. de Ingeniería de Telecomunicación

A Machine Learning based NLOS Detector for Robust GNSS Positioning in Urban Environments

Author: Juan Carlos Ruiz Sicilia

Supervisor: Mari Carmen Aguayo Torres

Department: Ingeniería de Comunicaciones (IC)

Degree: Máster en Ingenieria de Telecomunicaciones

Keywords: GNSS, Machine Learning, NLOS, Reliability

Abstract

Ensuring reliability is currently one of the main challenges in Global Navigation Satellite System (GNSS) positioning, especially in challenging urban environments with the presence of multiple local threats, such as multipath or non-line-of-sight (NLOS) signals. These error sources are difficult to model and isolate and, therefore, to predict or detect. To solve this problem, Machine Learning (ML) algorithms have started to be used as a manner of handling this complexity due to their strength in finding relations between features. However, the current state of the art is limited and is based on questionable features. Moreover, proposed algorithms are highly dependent on the training data and, therefore, they may not be robust against variations in the scenario or the installation.

In this work, a LOS/NLOS detector based on logistic regressions will be presented. The proposed algorithm includes features from multiple frequencies and a methodology to make the detector less dependent on the specific antenna and receiver and the specific data used for training. Additionally, the ML algorithm results are incorporated in the measurement model of a Position, Velocity and Time (PVT) estimator. Results show that the proposed detector presents significant benefits over the literature, and its output improves the final position computation error.

Acknowledgements

I would like to express my gratitude to Omar Garcia Crespillo who supervised my activity at DLR. After these six months under his guidance, I have learned from him how the researching work should be. His critical analysis and his rigorous methodology are some of the skills that I will try to continue applying in the future steps of my professional career.

Additionally, an especial thanks to the DLR staff who have shared their time and knowledge with me during my work. In particular, to Ana Kliman who has been always willing to assist me in many tasks being key for the development of this work.

My sincere appreciation to Mari Carmen Aguayo for being my mentor since the bachelor. Her orientation and support during these years have contributed considerably to my personal and professional development.

Also, I have to mention here the small Spanish community in Unterbrunn. I can not imagine this experience without Lorena, Paco and Pablo. We have shared a lot of experiences, worries and achievements together.

Finally, I can not express how grateful I am to my parents. They have made all they could to give me the best opportunities and their unwavering support has been fundamental for being where I am today.

Acronyms

ADC	Analog-Digital Converter
ARNS	Aeronautical Radionavigation Service
BOC	Binary Offset Carrier
BPSK	Binary Phase Shift Keying
CDMA	Code Division Multiple Access
COTS	Commercial-Off-The-Shelf
DC	Direct Current
DLL	Delay Lock Loop
DLR	Deutsches Zentrum für Luft- und Raumfahrt e.V.
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
LOS	Line of Sight
LR	Logistic Regression
LS	Least Squares
MAR	Missing At Random
MEO	Medium Earth Orbit
ML	Machine Learning
MNAR	Missing Not At Random
NLOS	Non-Line of Sight

MSE	Mean Squared Error
PDF	Probability Density Function
PRC	Pseudorange Rate Consistency
PLL	Phase Lock Loop
PPP	Precise Point Positioning
PPS	Precise Positioning Service
PRN	Pseudo-Random Noise
PVT	Position, Velocity and Time
RINEX	Receiver INdependent EXchange
RNSS	Radio Navigation Satellite Services
SDR	Software Defined Radio
SPS	Standard Precise Service
SVM	Support Vector Machine
ToA	Time of Arrival
WLS	Weighted Least Squares

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Nowadays, Global Navigation Satellite Systems (GNSSs) are broadly extended for positioning purposes. Due to its proven capabilities, it is used for most of the applications related to non-safety navigation or location-based services. These systems, for instance, Galileo or Global Positioning System (GPS), are worldwide available and open access which are some of the advantages that make them superior to any navigation system

However, future applications such as autonomous cars or trains require a high level not only of accuracy but also of reliability and robustness. At present, GNSSs can not provide this performance, specially in land applications where the system can suffer a degradation due to the existence of several threats. Furthermore, this degradation is even higher in urban scenarios, such as urban canyons, where the buildings and other elements act as obstacles for the direct line vision between the satellites and the GNSS receiver. Examples of these threats are Non-Line of Sight (NLOS) signals, multipath (i.e., reflections of the signals intro nearby objects) and intentional or unintentional interferences.

These sources of error are difficult to model and predict and, for this reason, their detection and mitigation is still an open challenge.

In this context, the Institute of Navigation and Communication of the German Aerospace Center (DLR) is studying different approaches to improve the reliability and robustness of GNSS positioning in urban scenarios. In particular, this work is focused on detecting and mitigating NLOS signals as they are one of the greatest threats for land applications.

1.1 Motivation

The motivation behind this thesis is given by the increasing interest in autonomous cars and other safety-of-life applications that require higher reliability and robustness for GNSS positioning in challenging signal visibility scenarios.

This performance can only be ensured if the threats that especially appear in urban scenarios are correctly detected and mitigated. In the case of NLOS signals, this is still an open problem. The complexity of modelling this threat has complicated the development of a robust detector using traditional approaches.

Nevertheless, the impulse of Machine Learning (ML) algorithms in recent years has allowed to consider them as a valuable candidate to handle this complexity and develop robust detectors. In particular, these algorithms are especially strong finding hidden patterns or relations between examples and building a generalized model from them. The development of these algorithms requires a first phase of training in which the algorithm learns from the examples.

Despite its potential, the application of machine learning for GNSS implies some challenges that need to be tackled. Firstly, the designed inputs of the algorithms named features have to be robust against variations in the conditions given in the training phase. For instance, systems based on Machine Learning must work in scenarios different from the training location and with different antennas. Furthermore, some of these algorithms require labelled data for its training phase. This implies to have a reference ground truth for the presence of GNSS threat under study. This labelling process is not easy in general for GNSS applications because isolating those errors is a complex task even in a pos-processing phase.

In recent years, a few works have proposed Line of Sight (LOS)/NLOS detector based on some of these Machine Learning techniques. However, this thesis will show that those detectors have some limitations in terms of robustness.

1.2 Objectives

The main objective of this thesis are the following:

- Develop an ML algorithm and related methodologies to design a robust LOS/NLOS detector.
- Include the detector in a Position, Velocity and Time (PVT) algorithm to improve the final position estimation.

1.3 Methodology

In order to achieve the goals of this thesis, the following tasks need to be carried out:

- A deep study of the current state of art about the application of Machine Learning for GNSS. Furthermore, a critical analysis of the works related to LOS/NLOS detectors.
- Design the most relevant features that can be useful for distinguishing between LOS and NLOS. Moreover, they must not be affected by variations in the scenario or the installation.
- Select or develop the ML algorithm that can seize better the information given by the features.
- Design a labelling methodology for LOS/NLOS signals.
- Carry out a measurement campaign to train and test the algorithms.
- Test the developed detector not only in the same scenario as trained but also in different ones.

1.4 Brief Overview: Structure of the Thesis

The rest of this thesis is organized as follows: Chapter 2 is an introduction of the basic notions about GNSS. In the same way, Chapter 3 gives an overview of Machine Learning and provides a background of the specific algorithms and concepts used in the rest of the thesis. Then, the current state of the art about the application of Machine Learning for GNSS with a particular focus on LOS/NLOS detectors is explained in Chapter 4. After contextualizing the work, Chapter 5 develops the theoretical study of the problem and present the NLOS detector designed in this work. Chapter 6 details the measurement campaign carried out to train the algorithms and test their performance. The analysis of those results can be found in Chapter 7. Finally, some conclusions and possible future steps are presented in Chapter 8.

Chapter 2

Fundamentals of GNSS

This chapter provides the theoretical notions related to GNSS which are essential in order to understand the work presented in this thesis.

Firstly, the GNSS technology is explained, mainly focusing on GPS and Galileo. Then, the raw measurements that can be extracted from the satellites signals also known as observables are presented. Finally, the PVT solver that computes the position based on those observables is introduced.

2.1 GNSS Systems

The Global Navigation Satellite System (GNSS) term comprises all the technologies for location and positioning based on a constellation of satellites. The fundamental idea behind these systems is to measure the ranges between the user receiver and those satellites processing their transmitted signals. The knowledge of this distance and of the position of the visible satellites can reduce the user position computation to a geometrical problem.

The first two fully functional systems were the GPS developed by the United States of America and the Russian GLONASS. Both were fully functional around 1995. Although it was firstly military-oriented, it was discovered later its potential for civil purposes. In the last years, the Chinese Beidou and the European Galileo have been deployed.

In general, all the GNSS systems have the same architecture based on three parts called segments which are the following:

Space segment This segment comprises the satellites. Their function is to transmit the signals that are used for measuring the ranges. Moreover, in that signal, it is sent the navigation message. This message contains the information necessary for the position acquisition such as the satellite clock bias and the ephemerids for the satellite position.

Control segment This segment consists of ground stations that monitor the health of the satellites and update the information they transmit in the navigation message.

User segment This segment involves the user receiver that tracks the satellites and computes the PVT.

Nevertheless, each system has its specific characteristics. Considering this work is focused on GPS and Galileo, both systems are going to be explained in more detail.

Global Positioning System

GPS has a constellation of 32 satellites which are distributed on Medium Earth Orbits (MEOs). The constellation orbits have a repeat cycle of one day.

All the GNSS satellites transmit in the Radio Navigation Satellite Services (RNSS) allocated spectrum. However, each system has its own bands. In particular, GPS signals were initially transmitted only in two frequencies, L1 (1575.42 MHz) and L2 (1227.60 MHz), and two services were available:

- The Standard Precise Service (SPS) which is civil service that allows a single-frequency position acquisition using L1.
- The Precise Positioning Service (PPS) which is restricted for military use and provide more precision by seizing L1 and L2 with dual-frequency receivers.

The signals for all the satellites are multiplexing using a Code Division Multiple Access (CDMA) scheme where the code used named Pseudo-Random Noise (PRN) is unique for each satellite. In this way, the receiver can recognize each satellite correlating its code with the received signal. Furthermore, if the result of this correlation is successful, the receiver will be able to extract the data transmitted by that satellite. For this purpose, the receiver must know the codes of all the satellites. Regarding the transmitted data, it is used a Binary Phase Shift Keying (BPSK) modulation scheme with a binary rate of 50 bps.

In this initial version of GPS, two types of PRN codes are used: the Coarse/Acquisition Codes and the Precision Codes. On the one hand, the C/A codes are used in SPS and are public. Those codes contain 1023 bits repeated each millisecond, spreading the signal bandwidth to 1.023 MHz. On the other hand, the Precision Codes are reserved for military use and, therefore, they are not publicly known. These codes used for PPS are very protected and spread the bandwidth to 10.230 MHz.

The commercials necessities of an improved GNSS system was one of the factors that motivated a GPS modernization. In this modernization, a third band L5 (1176.42 MHz) was included for civil use. That allowed the development of dual-frequency receivers, which are capable of giving a more precise position. In this band, PRN codes of 10 MHz are used. Although these codes with higher rates are more difficult to track, they are more robust against multipath.

Moreover, L5, as well as L1, are reserved spectrum for Aeronautical Radionavigation Service (ARNS). Hence, they are more protected against interference than L2. A visual summary of the allocated bands for GPS is shown in Fig. 2.1.

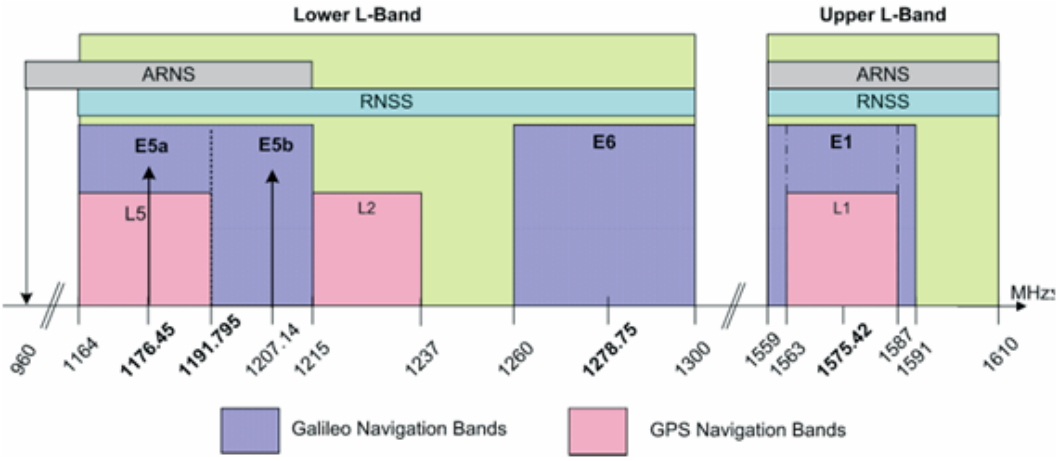


Figure 2.1: Allocated spectrum for GPS and Galileo. Resource from [1].

Galileo

The Galileo constellation consists of 36 satellites distributed in MEOs. The repeat cycle for their orbits is ten days.

The satellite signals are also multiplexed using CDMA. However, as Galileo design and deployment was carried out later than GPS, it has already included in its initial version some modern concepts. For instance, it includes three open service bands to allow dual-frequency schemes or even triple frequency services. These bands are L1, E5a which is approximately at the same frequency as L5 and E5b (1207.14 MHz). All these bands are part of the ARNS spectrum. Even though Galileo was mainly designed for civil use, they reserved a band E6 (1278.75 MHz) for commercial and government purposes.

As Galileo shares some bands with GPS, data is modulated using a more complex scheme named Binary Offset Carrier (BOC) in order to reduce the overlapping in the spectrum domain between their signals.

GNSS receiver scheme

As GPS and Galileo have similar signals and use CDMA, the receivers are simple because they have to carry out almost the same processing tasks for both constellations. In particular, a simplified diagram of the structure of a single frequency receiver is shown in Fig. 2.2.

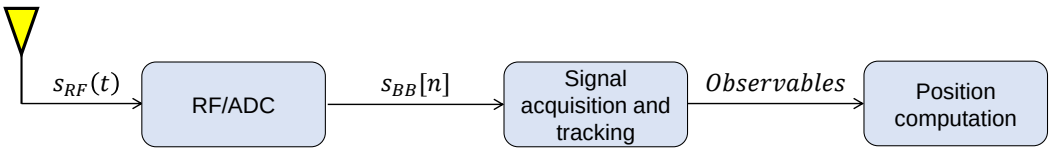


Figure 2.2: Diagram of GNSS receiver.

As it is shown in the block diagram, the antenna specifically designed for the RNSS bands receives the electromagnetic waves $s_{RF}(t)$ which are down-converted to an intermediate frequency or, even, baseband, and sampled by the Analog-Digital Converter (ADC) obtaining $s_{BB}[n]$.

Once in this point, it is carried out a signal acquisition and tracking for identifying the signal of each satellite. In this process, it is obtained the Time of Arrival (ToA) of each satellite as well as some more information named as observable that are necessary for the PVT.

The acquisition phase consists of a rough estimation of the code delay and the Doppler frequency of a specific satellite signal. Then, the signal is tracked with a Delay Lock Loop (DLL) and a Phase Lock Loop (PLL) respectively. Therefore, in the tracking phase, these parameters are refined using both loops. Through both phases, the receiver extracts the ToA, which is used for computing the pseudoranges and a precise estimation of the Doppler frequency.

2.2 Observables

The observables that could be measured from the satellite i and the band j are:

- The pseudorange $\rho_{i,j}$ which is the apparent distance between the satellite and the receiver. However, it contains errors that need to be correcting to compute the position. It is measured in metres.
- The Doppler frequency $f_{D,i,j}$ which is the difference between the nominal central frequency of the band and the received frequency. This measurement has a direct relation with the relative velocities between the satellite and the user. It is measured in Hz.

- Finally, the last observable is the carrier phase which is the difference between the phase of the received signal and the phase of the local replica of the code. It is measured in radians. In this work, it is not going to be used, but more information about it can be found in [2].

Additionally, it can be included in the information extracted from the correlator the Carrier-to-Noise ratio $C/N_{0,i,j}$. It is a measurement of the signal power divided by the noise power density after the correlators. Therefore, it is measured in $Watt/Hz$ or more often in dB/Hz.

Below, the observables of interest are explained in detail.

Pseudorange

Considering all the factors that can affect to the real range estimation, the pseudorange of the satellite i and the frequency j can be expressed as:

$$\rho_{i,j} = R_i + c(dt^u - dt_i) + Tr_i + I_{i,j} + M_{i,j} + \epsilon_{i,j} \tag{2.1}$$

Where:

- ρ_i is the real distance between the satellite and the user.
- c is the speed of light.
- dt^u is the receiver clock bias.
- dt_i is the satellite clock bias.
- Tr_i is the tropospheric delay which does not depend on the frequency.
- $I_{i,j}$ is the frequency-dependent ionospheric delay.
- $M^P_{i,j}$ is the multipath component.
- $\epsilon^P_{i,j}$ is the receiver noise.

In order to obtain the final position, it is necessary to isolate the real distance between the satellite and the receiver. The satellite clock bias can be computed based on models and known keplerian equations with some information from the navigation message. In single-frequency receivers, the ionospheric and the tropospheric delays can be estimated using models. Finally, the user clock bias is included as one of the unknowns that must be solved in the PVT. The multipath and the noise can not be estimated with classical approaches, so it will unfortunately produce an error in

the position domain. It is important to mention that NLOS effect can be included in the multipath term since it can be considered worst case of multipath possible. Therefore, the error in position domain due to NLOS signals will be dramatically higher.

Doppler frequency

The Doppler shift measurement is usually given in units of m/s. Hence, it is necessary to convert to this unit by multiplying with the wavelength of its band. The resulting pseudorange rate can be modelled as:

$$\dot{\rho}_{i,j} = -\lambda_j f_{D_{i,j}} = (v^u - v_i)e_i^u + c\left(\frac{\partial dt^u}{\partial t} - \frac{\partial dt_i}{\partial t}\right) + \varepsilon_{i,j} \tag{2.2}$$

Here:

- $(v^u - v_i)$ is the speed difference between the user and the satellite in ECEF coordinates.
- e_i^u is the LOS pointing vector from the receiver antenna to the satellite.
- $\epsilon_{i,j}^r$ is the noise in the measurement which could be caused by second-order factors.

2.3 PVT: Weighted Least Squares Estimator

PVT is one of the most classical algorithms for the position computation. The main goal of this algorithm will be to compute not only the position of the receiver (x_u, y_u, z_u) but also the user clock bias dt^u. It is normally computed using a least-squares process with the pseudoranges as input measurements. Since there are 4 unknowns, it is requires at least 4 visible satellites.

As the navigation message contains the information necessary to compute sat clock and position and to compute ionospheric corrections based on Klobuchar model [3], it is possible to simplify the expression in Eq. (2.1) to:

$$\rho_i = \sqrt{(x^i - x_u)^2 + (y^i - y_u)^2 + (z^i - z_u)^2} + c \cdot dt^u + \epsilon_i'; \tag{2.3}$$

where the last term comprises the residuals terms: the noise, the multipath and the error of the aforementioned models. Furthermore, the satellite position (x^i, y^i, z^i) is computed based on the ephemeris from the navigation message. Although the pseudorange measured is different for each band, the PVT is normally computed

with the pseudorange of one frequency, so the notation has been simplified to make it more understandable.

As the least-squares solution is a linear estimator, it is therefore, necessary to linearize the unknowns in Eq. (2.3). This is generally done by expanding the Taylor series of the non-linear term around an initial position $(x_{u,0}, y_{u,0}, z_{u,0})$ which initial distance to the satellite is $R_{i,0}$:

$$\rho_i \approx R_{i,0} - \frac{(x^i - x_{u,0})}{\rho_{i,0}}\Delta x + \frac{(y^i - y_{u,0})}{\rho_{i,0}}\Delta y + \frac{(z^i - z_{u,0})}{\rho_{i,0}}\Delta z + c \cdot dt^u + \epsilon_i'. \tag{2.4}$$

Now, it is possible to build the equation system with the N visible satellites:

$$\underbrace{\begin{bmatrix} -\frac{(x^1-x_{u,0})}{R_{1,0}} & -\frac{(y^1-y_{u,0})}{R_{1,0}} & -\frac{(z^1-z_{u,0})}{R_{1,0}} \\ -\frac{(x^2-x_{u,0})}{R_{2,0}} & -\frac{(y^2-y_{u,0})}{R_{2,0}} & -\frac{(z^2-z_{u,0})}{R_{2,0}} \\ -\frac{(x^3-x_{u,0})}{R_{3,0}} & -\frac{(y^3-y_{u,0})}{R_{3,0}} & -\frac{(z^3-z_{u,0})}{R_{3,0}} \\ \vdots & \vdots & \vdots \\ -\frac{(x^N-x_{u,0})}{R_{N,0}} & -\frac{(y^3-y_{u,0})}{R_{3,0}} & -\frac{(z^3-z_{u,0})}{R_{3,0}} \end{bmatrix}}_{\boldsymbol{H}} \underbrace{\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ dt^u \end{bmatrix}}_{\boldsymbol{\Delta \hat{x}}} = \underbrace{\begin{bmatrix} \rho_1 - R_{1,0} \\ \rho_2 - R_{2,0} \\ \rho_3 - R_{3,0} \\ \vdots \\ \rho_N - R_{N,0} \end{bmatrix}}_{\boldsymbol{z}}; \tag{2.5}$$

This system of equations can be easily computed with the pseudo-inverse:

$$\boldsymbol{\Delta \hat{x}} = (\boldsymbol{H}^T \boldsymbol{H})^{-1} \boldsymbol{H}^T \boldsymbol{z} \tag{2.6}$$

With this result, the initial position assumed can be corrected. However, the accuracy of the result obtained is very dependent on the initial position chosen. Therefore, it is interesting to seize the iterative structure of the algorithm in which each iteration k the position is updated:

$$x_{u,k+1} = x_{u,k} + \Delta x_{k+1} \tag{2.7}$$

$$y_{u,k+1} = y_{u,k} + \Delta y_{k+1} \tag{2.8}$$

$$z_{u,k+1} = z_{u,k} + \Delta z_{k+1} \tag{2.9}$$

This is the basic Least Squares (LS) algorithm to compute the position. However, in this case, the position will be computed assuming that the error in the pseudorange measurement is independent and identically distributed. This is not necessarily true considering, for instance, that some models can be less precise for satellites with a low elevation or whether a satellite is NLOS. In order to allow the inclusion of these considerations, it is widespread the use of the Weighted Least Squares (WLS). This solver includes a weighting matrix $\boldsymbol{W}$ which is generally related to the inverse of the covariance matrix of the error in the pseudoranges and its expression is:

$$\boldsymbol{\Delta \hat{x}}_{WLS} = (\boldsymbol{H}^T \boldsymbol{W}^{-1} \boldsymbol{H})^{-1} \boldsymbol{H}^T \boldsymbol{W}^{-1} \boldsymbol{z} \tag{2.10}$$

This covariance matrix can be estimated based on different models as it will be explained more in detail in 5.5.

Chapter 3

Machine Learning Algorithms

This chapter provides with an overview of the basics about Machine Learning and some of the most used algorithms.

Firstly, the philosophy and the methodology that machine learning follows are presented. Then, the specific algorithms used in this work are explained in detail with a particular focus on Logistic Regression (LR). The last section of this chapter includes an overview of the ML framework used in this work which is called Scikit-learn.

3.1 Fundamentals

Machine Learning (ML) algorithms have acquired an increasing ìmportance in recent years. This group of algorithms is especially interesting for those situations where there are many data somehow related between them, but the precise relationship is unclear. In other words, these techniques try to obtain a general model by "learning" from some examples. ML comprises a group of algorithms that can be used for regression or classification. In the first case, the algorithm output must be a continuous variable while, in the second case, the result is the prediction between a discrete group of classes.

There is another classification of the ML algorithms:

Supervised learning These algorithms fit a mapping function between some inputs and an output. In order to fit its parameters, it is required labelled data with the ground truth that want to be predicted.

Unsupervised learning These algorithms do not need labelled data. In contrast, they try to find hidden patterns between data.

Reinforcement learning In this case, the algorithms are trained to make a sequence of decision in a complex environment. Its behavior is corrected based

on rewards and punishments.

This thesis is based on supervised learning. The operational flow with these algorithms is firstly to carry out a training phase and then evaluate it in a testing phase. In order to ensure that the trained algorithm is well generalized, the whole dataset must be split into independent parts. In this way, it will be proved that the algorithm works right not only for the training data but also for all the possible data.

In the case of classification, there are some metrics that are usually employed to evaluate the performance of an ML implementation. The first step after running the test is to count the true positives tp, the true negatives tn, the false positives fp and the false negatives fn. Fig.3.1 clarifies their definitions using a matrix representation known as *confusion matrix*.

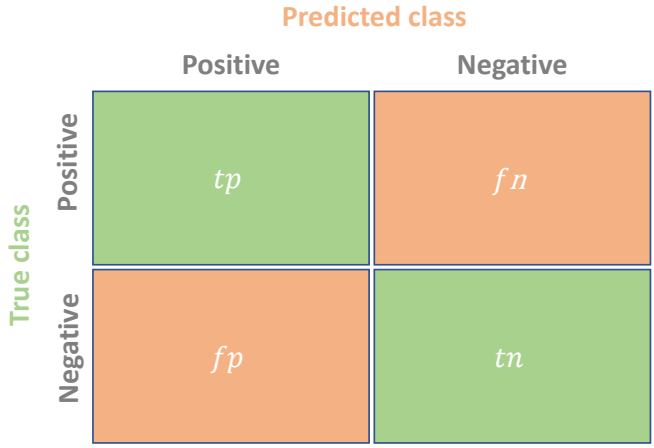


Figure 3.1: Confusion matrix with ML classification results.

With these elements, almost all the metrics can be computed. In particular, the most common is the accuracy which is the ratio of correct predictions to the total number of evaluated samples. Based on the aforementioned elements, the accuracy is computed with:

$$Accuracy = \frac{tp + tn}{tp + fp + tn + fn} \tag{3.1}$$

However, one value is not enough to extract insights about the performance and behavior of the algorithm. In contrast, the confusion matrix gives insights into how balanced is the error between the positives and the negatives. The matrix can be filled not only with the absolute values as Fig. 3.1 but also with the relative ratios as shown in Fig. 3.2.

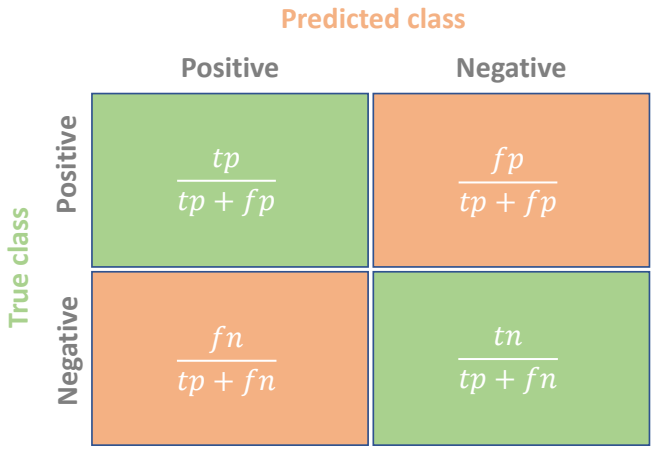


Figure 3.2: Statistics of ML performance.

3.2 Decision Trees

The decision tree is a supervised algorithm for classification or regression. It is widely used because the final flowchart of the trained model can be visually analyzed. This allows extracting insights about the behavior of the algorithm.

As its name indicates, the structure of this algorithm is a tree where each leaf node represents a condition or decision rule. This decision rule is generally a threshold for a feature fitted as a function of the dataset. An example of a decision tree for predicting whether it will rain is shown in Fig. 3.3. The more depth the tree has, the better the tree will be fitted. However, a very deep tree could cause overfitting to the training dataset losing generalization and, therefore, accuracy in the testing dataset.

The main advantages of decision trees are not only their visual representation but also their robustness against outliers. Moreover, decision trees give insights into which features are the most important. However, they can not seize all the features simultaneously, but instead, they prioritize only the main important ones. Another problem of this algorithm is its propensity to overfitting and the resulting unstable models where a slight variation in the input can entirely modify the output.

3.3 Support Vector Machines

The Support Vector Machine (SVM) is a supervised algorithm which can work both for regression and classification. In contrast to the decision tree, this algorithm can consider a large number of features to obtain the best possible prediction. This is done by finding a hyperplane with the same dimensionality as the number of

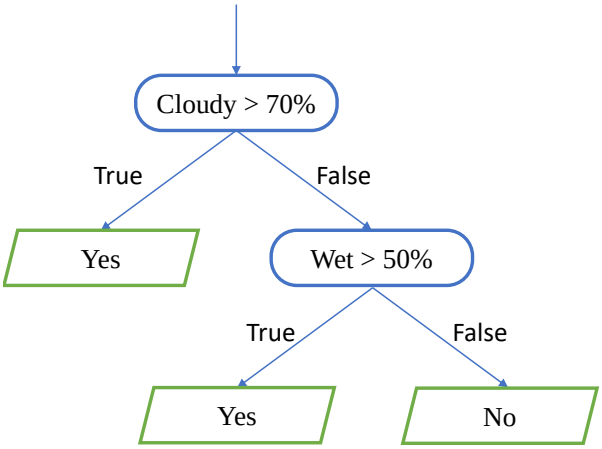


Figure 3.3: Example of decision tree for predicting whether it will rain.

features that divides the two possible labels. Moreover, this can be extended to multiple classes by computing more hyperplanes.

Although the aforementioned hyperplane is the decision bound between both labels, multiple hyperplanes can be fitted with the same data by slight rotations or shifting. As this is not convenient, an additional criterial is included to find the optimal hyperplane. It consists in finding the hyperplane for which the distance with the closest points for each label must be maximal. An example for the case of two dimensions is shown in Fig. 3.4.

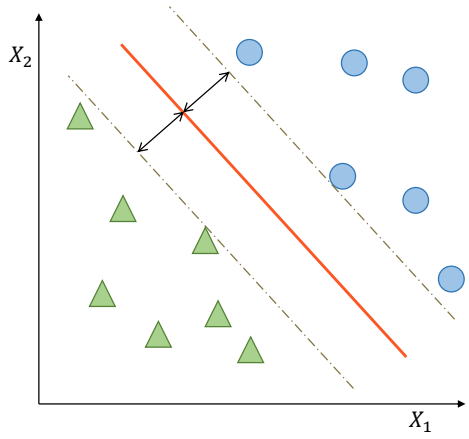


Figure 3.4: Example of SVM fitting for two dimensions.

If the most accurate bound is not linear, other shapes can be used as quadratic or radials by processing the features with the corresponding functions called kernels. However, computing complexity will increase with kernels different from the linear.

In any case, even with the linear kernel, complexity is one of the main disadvantages of SVM and this restricts the amount of data that can be used for training.

3.4 Logistic Regression

The logistic regression is one of the most used ML algorithms for classification. This algorithm has as output not only the predicted class but also the estimated probability of being in that class. This information could be helpful for some applications, as it will be shown in this work.

The sigmoid is the mapping function for the logistic regression as the target of this algorithm is to estimate a probability and the sigmoid is bounded between 0 and 1. In particular, its mathematical expression is:

$$Pr(\boldsymbol{x};\boldsymbol{\beta}) = \frac{1}{1+e^{-\boldsymbol{\beta}^T\boldsymbol{x}}}; \tag{3.2}$$

where $\boldsymbol{x}$ is the vector of features used as input and $\boldsymbol{\beta}$ is the vector of coefficients that must be fitted. The final predicted class is based on this estimated probability. It is considered positive if $Pr(\boldsymbol{x}) < 0.5$ and negative if not. The sigmoid and the decision bounds are shown in Fig. 3.5.

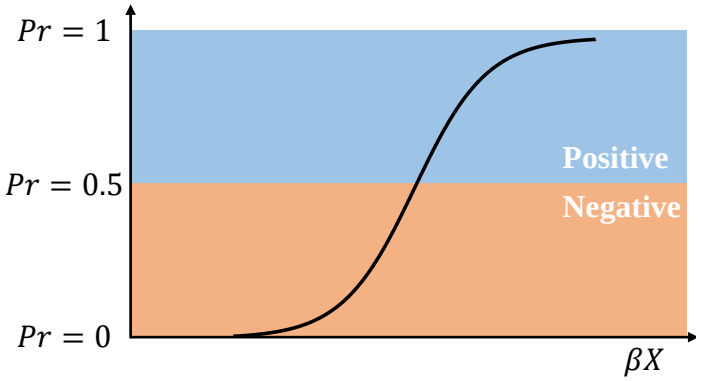


Figure 3.5: Logistic regression decision bounds.

At this point, the test phase of the LR has been explained but not the training phase. This is usually done by fitting $\boldsymbol{\beta}$ in order to minimize a cost function typically the Mean Squared Error (MSE):

$$J(\boldsymbol{\beta}) = \frac{1}{M}\sum_{i=1}^{M}(Pr(\boldsymbol{x}_i;\boldsymbol{\beta}) - y_i)^2; \tag{3.3}$$

where M in the size of the training dataset and y_i is the label of the data.

The typical approach to tackle this optimization problem is to use gradient-descendent algorithms. More information about it can be found in [4].

3.5 The Scikit-learn Framework

Implementing ML algorithms is a complex work. However, different frameworks have been developed to offer general-purpose software that handles this complexity in recent years. In this way, researchers and developers can focus on the features design and data processing for their specific work. Following this philosophy of simplicity, most of the frameworks have been developed with Python, which is a relatively easy language to learn. Examples of this type of framework are Pytorch [5] and Keras[6] for neural networks and Scikit-learning [7] for machine learning which is the used in this work.

Scikit-learn is a popular and powerful framework used which implements the main ML algorithms such as decision tree, SVM and logistic regression. It contains implementations for supervised and unsupervised algorithms as well as for classification and regression purposes. Furthermore, it has additional valuable functions for preprocessing the data, for instance, for splitting the data into the training and the testing dataset and for balancing the amount of data for each label.

Chapter 4

Machine Learning in GNSS Survey

In this chapter, a summary of the main state of about the application of Machine Learning in GNSS is presented. In particular, the first section is related to an overview of the current state of the art about all the proposed applications in GNSS. The second section is focused on a deep study and a critical analysis of the LOS/NLOS detectors which are the main topic of this work.

4.1 General Overview

As the development of ML techniques is recent, there are just a few works related to its application in the GNSS field. They are focus on the detection of some threats to improve the reliability of GNSS positioning for land applications. The current state of the art has not robust detectors for those sources of error using classical approaches. However, it is considered that ML techniques are a good candidate for carrying out these detectors [8]. In particular, in the state of the art, these techniques has been applied to handle the following threats:

NLOS signals It refers to the situation when the path between the satellite and the receiver is obstructed partially or entirely by one or more obstacles. However, some reflected signals reach the receiver, leading to an error in the receiver about the real range.

Multipath In this case, although there is line of sight between the satellite and the receiver, some reflected rays also reach the receiver. This skew the correlation peak misleading the receiver, in the same way, about the real distance between satellite and receiver.

Ionospheric scintillations The electromagnetic waves can be sometimes distorted passing through the ionosphere. It happens because, in this region of the

atmosphere, some small scale irregularities in the electron density can appear due to the ionization of the sun. This produces rapid fluctuations in the refraction index and, therefore, variations in the observables.

Spoofing This is one of type of intentional attack to GNSS systems to degrade their performance. A malicious transmitter generates the same signals as the satellites. In this way, the attacker can give false information to the receiver which can potentially mislead it.

In Fig. 4.1, it can be observed a representation of the aforementioned threats.

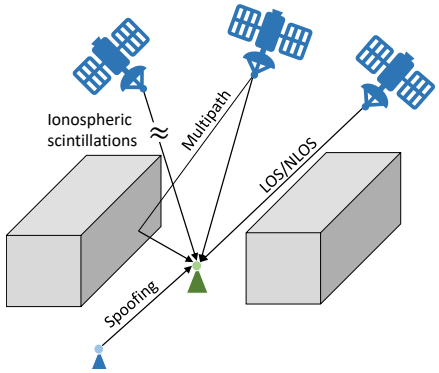


Figure 4.1: Threats to GNSS positioning for land applications.

In the literature, the detectors are carried out at different levels in the receivers. For instance, some works propose detectors for multiple threats using samples obtained from the correlator output. In [9], a multipath detector is based on the creation of a 2D image combining the information of the DLL and the PLL. The detection is carried out using this image as the input of Convolutional Neural Network, a very popular ML technique for image processing.

The most typical approach for carrying out spoofing attacks is based on re-emitting an authentic satellite signal with a certain delay [10]. This produces strong multipath at the receiver that can be detected processing the delays in the correlation peaks computed in the DLL. However, information at this low level is not usually provided in Commercial-Off-The-Shelf (COTS) receivers. Hence, developments at this level only can be done using self-developed GNSS Software Defined Radio (SDR).

At the level of basebands signals, in [11], some statistics are extracted from baseband in-phase and quadrature signals. Then, they are used as features in a decision tree for detecting ionospheric scintillations and multipath. However, since baseband signals are recorded before the correlators, it is not possible to distinguish which particular satellite signals suffer multipath. It is only possible to detect if

any or some of the satellites are affected by it. Therefore, this approach does not allow to discard the affected measurements or to mitigate somehow the effect of those threats on the corresponding satellite measurements. Furthermore, some commercial receivers can deliver this information but with a low sampling frequency and a low bit resolution. For example, in the case of the receiver used in this work, the Septentrio Mosaic-X5, the baseband sampling frequency is 20 Hz and the bit resolution is 8 bits.

Finally, most of the papers related to ML in GNSS use information obtained at the observable and position domain. Almost all the receivers allow recording this information and there are file formats such as Receiver INdependent EXchange (RINEX) that standardize the storage and interchange of the observables. In [11], it is proposed a detector with some features that belong to this level based on decision trees for ionospheric scintillations. Another example is [12] where it is presented a detector of anomaly satellite signals which are those signals considered for some reason not healthy. This consideration is carried out through a labelling methodology that discovers this anomaly behavior based on a clustering algorithm. Moreover, all the papers presented in the next section are based on information at this level.

4.2 LOS/NLOS Detectors in the Literature

There are some papers related to the detection of NLOS signals in the literature. Some of them try not only to distinguish between LOS and NLOS but also multipath as an intermediate case.

The features that have been considered in previous works are:

- C/N_0 is the carrier to noise ratio
- $\Delta C/N_0$ is the difference between two consecutive errors of the carrier to noise ratio.
- θ is the elevation of the satellite.
- PR is the pseudorange residual. This value is obtained after the position computation by subtracting to the pseudorange, the estimated range and all the known terms such as the clock error and the ionospheric and tropospheric error. If the position estimation were perfect, the pseudorange residual would be only the error due to the noise and the multipath.
- Pseudorange Rate Consistency (PRC) is the difference between the changing rate of pseudorange measurements ΔR and the Doppler, which is more robust

against multipath and NLOS signals. Hence, it is a way to isolate the effect of those effects in the pseudorange rate.

$$PRC = |\Delta R - (-\lambda f_D)\Delta t| \tag{4.1}$$

Additionally, Table 4.1 summarizes all the state of the art found about this specific topic.

Goal	Feature vector	ML technique	Reference
LOS/NLOS	$[C/N_0, \theta, PR]$	Decision tree	[8]
LOS/NLOS	$[C/N_0, \theta, PR, PRC]$	SVM	[13]
LOS/Multipath/NLOS	$[C/N_0, \Delta C/N_0, PR, PRC]$	SVM	[14]
LOS/Multipath/NLOS	$[C/N_0, \theta, PR]$	Decision tree	[15]

Table 4.1: Summary of the state of the art of NLOS detectors.

As it is shown in Table 4.1, the most used algorithms for NLOS detection are decision trees and SVM. Furthermore, the proposed feature vectors have some similarities, such as the C/N_0.

However, some of the proposed features depend on the scenario and the installation used for the training. This installation comprises the specific setup used for recording the data such as the antenna type and the receiver model. One of these questionable features is the elevation θ. Although it is more likely to find NLOS signals from satellites at low elevations, this is not necessarily true. Moreover, the decision of an algorithm with this feature would be conditioned by the elevation of the obstacles in the training dataset used.

Another case of these debatable features is the pseudorange residual. As has been mentioned previously, this is obtained after the position computation. Therefore, it depends on the specific algorithm used for the position, so it would make the detector not valid for different estimators. Moreover, since the computed position is different depending on how many satellites are used and how healthy their signals are, the residuals would change for each situation. This could lead to considering LOS or NLOS because of the conditions not only of the satellite on test but also of the rest of satellites.

The C/N_0 is clearly an important feature to distinguish between LOS and NLOS considering that the Probability Density Functions (PDFs) of each class are noticeably different as explained in [8]. However, this feature has a strong dependency on the specific installation used for recording the data. For instance, if an algorithm that uses this feature is trained with a specific installation and later, the receiver or the antenna arec changed, the detector will considerably degrade its performance. Furthermore, the C/N_0 also depends on the elevation due to the radiation pattern of the antenna and other factors, as it is shown in Section 7.1.

Regarding the labelling methodology, in [13], [14] and [15] 3D building models are proposed to compute using ray-tracing techniques whether a signal is LOS or NLOS. However, it is complex to generate 3D building models and their precision is essential for the performance of the detector. In contrast, in [8] it is proposed to measure some physical distance and angles to the obstacles around the receiver in order to characterize them and, therefore, compute the labels. This procedure is easier to carry out but its performance strongly depends on the precision measuring. In particular, in [8], the sensors and camera of a mobile are used to take the measurement. This low-cost solution could not be the most precise one for obtaining the labels. Moreover, this methodology also depends on the accuracy of the obstacles characterization. Therefore, scenarios with elements as trees must be avoided as their behavior with electromagnetic waves are almost unpredictable.

Finally, it is important to mention that most of the papers do not consider how to use the predicted information to improve the position estimation. After the state-of-art review, only [12] proposes to discard the compromised measurements improving the computed position. However, if they are correctly handled, even NLOS signals can help to improve the position estimation, so a softer methodology would be more desirable.

Chapter 5

Non Line of Sight Detector

The main objective of this thesis is the development of a LOS/NLOS detector for GNSS satellites. As explained in the previous chapter, there are some works in the state of the art related to this specific type of detector. However, they generally use questionable features which are not robust against variations in the installation or the scenario.

In this chapter, it is detailed the methodology that has been followed to design and implement the detector. It starts by presenting the overall scheme of the ML chain used. Then, it follows with the methodology that has been followed to discover the ground truth. This ground truth has served to label the data in order to train the supervised ML algorithms. Once the data is labelled, the features and algorithm that have been selected and designed are presented. Finally, it is shown a methodology to include the LOS/NLOS detector in the PVT estimation.

5.1 Methodology Overview

The development of the detector has been carried out by considering the scheme in Fig 5.1. Two differentiated phases can be distinguished: the pre-processing phase and the ML phase.

Pre-processing Phase

This part is related to the signal processing required to convert the raw data from the receiver into a dataset that can be managed by an ML algorithm. All this process is carried out using MATLAB. It comprises the following blocks:

Data preprocessing All the data collected by the GNSS receiver is generally in a proprietary format but commercial brands usually gives software to convert it into RINEX. After it, the observables and navigation messages stored as

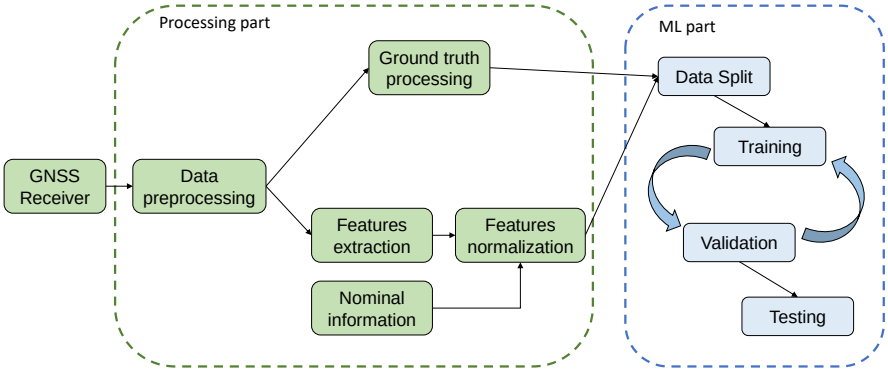


Figure 5.1: Diagram of the ML chain.

RINEX need to be parsed to a format understandable for Matlab. Furthermore, elevation and azimuth of the satellites need to be computed using the navigation message.

Ground truth processing As supervised learning algorithms require labelled data, a labelling methodology has to be carried out to discover this ground truth.

Features extraction This block comprises all the signal processing required to obtain the features utilized by the detector.

Features normalization This block represents the normalization carried out to remove the dependency of some features on the scenario and the installation. It is done using *Nominal information* obtained in an open-sky scenario.

The RINEX parser and the functions for computing the elevation and azimuth of the satellites from the ephemeris were already done by DLR.

Machine Learning Phase

In this part, the ML algorithm is trained and evaluated. However, both processes must not be carried out with the same data as it would potentially lead to overfitting. Therefore, the whole dataset is split into three parts:

- The training data is used to compute the coefficients that fit the algorithm.
- The validation data contains data recorded in the same scenario as the training dataset.
- The testing data is recorded in different locations different from the training scenario. It serves to evaluate how robust is the algorithm against variations in the scenario.

This division is necessary in order to avoid that the algorithm fits perfectly with the training data but it is not well generalized for a different dataset such as the validation data is not good enough. It is opportune to mention that the algorithm training and its evaluation with the validation data can be used as an iterative process to choose the final system with the best performance.

5.2 Labelling LOS/NLOS

One of the main difficulties for implementing ML algorithms for solving GNSS problems is the necessity of finding the ground truth. Regarding this particular topic, two methods that have been proposed in some papers. One is based on the application of ray-tracing algorithms in 3D models and the other solution is the estimation of the horizon considering the scenario as a geometric problem. Due to the complexity of designing 3D models, the second option has been selected as the methodology in this work.

The main idea consists in modelling the horizon of the elements around the receiver that potentially can be obstacles for the LOS signal between the GNSS satellites and the receiver. The main limitation of this method is that complex elements such as trees or traffic signals are challenging to model. Moreover, the behavior of the electromagnetic waves which go through these elements is quite unpredictable. Therefore, this type of element should be avoided in the selected scenarios for recording the data. However, in typical urban locations such as urban canyons, the main structures that fix the horizon for LOS signals are buildings whose shape can be decomposed in one or more surfaces.

For a particular azimuth φ_i, the first step is to find to which surface that azimuth belongs, i.e. $\varphi_i \in [\varphi_1, \varphi_2]$ where φ_1 and φ_2 are the azimuths at the ends of the surface. Then, the problem will be simplified to the sketch shown in Fig 5.2.

For obtaining the elevation θ_i of the horizon for each φ_i, it has been considered that the distances A, D_1 and D_2 are known. In the same way, at least one of the elevations at the sides of the building, e.g. θ_1 , and $\alpha = \varphi_2 - \varphi_1$ must be known. The measurement procedure for these magnitudes is explained in Section 6.2. In this way, the estimation of the horizon is reduced to a trigonometric problem computable following the Algorithm 1.

5.3 Features Extraction

One of the key elements for achieving a good performance with a ML algorithm is to select the most important features which allow distinguishing between the different labels or classes. Depending on the precise goal of the predictor, different features

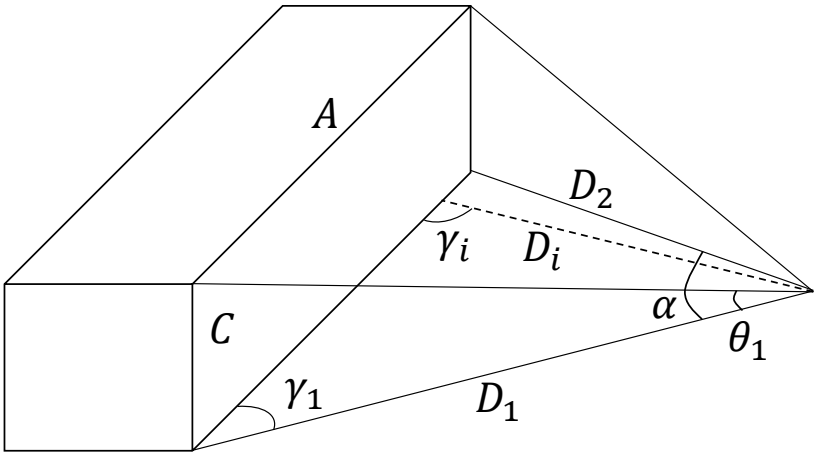


Figure 5.2: Sketch of the horizon determination.

Algorithm 1 Horizon determination considering a surface as an obstacle.

$C \leftarrow D_1/sin(\theta_1)$
$\gamma_1 \leftarrow arcsin(D_2 sin(\alpha)/A)$
for each $\varphi_i \in \varphi$ **do**
  $\alpha_i \leftarrow \varphi_i - \varphi_1$
  $\gamma_i \leftarrow \pi - \alpha_i - \gamma_1$
  $B_i \leftarrow D_1 sin(\gamma_1)/sin(\gamma_i)$
  $\theta_i \leftarrow atan(C/D_i)$
end for

would be more suitable. This work proposes to use the following features.

Carrier To Noise Ratio

As it is explained in the literature, the C/N_0 is expected to be one of the main descriptors for the LOS/NLOS prediction. However, in this work, it has been included as a novelty the C/N_0 not only of one band but also of some more. In particular, for GPS it has been taken into account L1, L2 and L5 while for Galileo, E1, E5a and E5b. However, the utilization of these measurements generates some problems that need to be tackled.

The first one is that the C/N_0 is an observable that highly depends on the installation used. For example, if one algorithm is trained using a specific installation, that is an specific setup of antenna and receiver, and an RF splitter is included, it would lead to an increment in the LOS signals predicted as NLOS signals due to the reduction of the C/N_0. Hence, in this work, it has been considered important to carry out a normalization process for increasing the robustness of the features.

The information for this normalization is extracted from data recorded in an open-sky scenario, from now on, the nominal state. In this way, GNSS threats as multipath or NLOS signals are avoided and it is possible to isolate the behavior of the signals due to the installation. From this scenario, it is possible to extract, for an elevation θ and the band j, the mean $\mu_j(\theta)$ and the standard deviation $\sigma_j(\theta)$ of the C/N_0 for each band. These statistics allow the normalization of the signal from the satellite i through:

$$\overline{C/N_{0,j}} = \frac{C/N_{0,j} - \mu_j(\theta_i)}{\sigma_j(\theta_i)} \tag{5.1}$$

Where all the C/N_0 are in units of dB/Hz.

Lock Time

In addition to the C/N_0, it has been included as a feature the lock time T_{lock} which is the time since the signal is tracked. It is expected that a short T_{lock} could be an indicator of NLOS. However, it has been detected that this parameter in raw could potentially generate overfitting problems in case of not having a huge amount of data. This is caused due to the large range of values that this parameter can have. Since the critical range where the feature is valuable for distinguishing between LOS and NLOS is in the first seconds, the solution for this problem is to saturate the maximum value:

$$T'_{lock} = \begin{cases} T_{lock}, & \text{if } T_{lock} < 15. \\ 15, & \text{otherwise.} \end{cases} \tag{5.2}$$

This value is set heuristically and its meaning is that the signal is on track from a long time ago.

Pseudorange Rate Consistency

À modified version of the PRC defined in Eq. (4.1) is included. The reason for this modification is that with the state-of-the-art definition, the algorithm would not be robust against variations in the sampling frequency. Therefore, it is proposed to define it as:

$$\overline{PRC} = \left| \frac{\Delta\rho}{\Delta t} - (-\lambda f_D) \right|. \tag{5.3}$$

Since f_D is robuster against multipath and other GNSS threats than the pseudorange variations, it is expected that higher values of this feature could potentially indicate the NLOS nature of the signal.

Final Feature Vector

Therefore, the selected feature vector for GPS and Galileo are respectively:

$$X_{GPS} = [\overline{C/N}_{0,L1}, \overline{C/N}_{0,L2}, \overline{C/N}_{0,L5}, T'_{lock}, \overline{PRC}] \tag{5.4}$$

$$X_{GAL} = [\overline{C/N}_{0,E1}, \overline{C/N}_{0,E5a}, \overline{C/N}_{0,E5b}, T'_{lock}, \overline{PRC}] \tag{5.5}$$

5.4 ML Algorithm Selection

Most of the papers related to NLOS detectors in GNSS propose to use SVM or decision trees as the ML algorithm for their implementation. However, these algorithms only give a binary decision between the labels. This is a limitation because the output is the same for a feature vector far from the decision bound and for one in its limit.

An alternative to these algorithms is the Logistic Regression (LR). It provides a continuous range of values between 0 and 1 as it models a probability. Hence, with the LR it is possible to consider not only the predicted class but also how clear it is for the algorithm prediction. In this way, it would provide a LOS probability estimation P_{LOS}.

However, the algorithm can not be applied immediately since other problems need to be tackled in the algorithm design. This problem is that sometimes the codes from some frequencies are not on track and, because of that, some C/N_0 are missing in the feature vector. Considering that the logistic regression requires a complete feature vector to make the prediction, a solution is required.

In the literature of Machine Learning, this situation where a feature is sometimes missing because its value is out of range is known as Missing Not At Random (MNAR)[16]. In other situations as Missing At Random (MAR), where some values are missing without any reason related to the data, these values can be imputed, i.e. estimated by inference. However, in the case of MNAR values, this is not an option so the literature proposes as the most common solution for this situation to define a default value for each of the features in case of being missing. For instance, if the number of square meters of a garage were a feature of an algorithm, a default value of 0 would be assigned to those houses without a garage. However, this solution does not fit in the NLOS detector because if a low default value is assigned to the missing values of both LOS and NLOS signals, the dataset would be distorted, and, therefore, the detector would be biased degrading its performance.

The alternative carried out in this work consists of some logistic regression organized in a structure named *Branched Scheme*. This system is represented in Fig. 5.3 for GPS but the same structure can be replicated for Galileo. As illustrated in

the figure, 4 logistic regression algorithms are trained with different feature vectors, each one depending on the available measurements. Therefore, when the system is evaluated with a vector of values, there is firstly a phase of searching the logistic regression that corresponds to the tracking frequencies and, then, a second phase of computing P_{LOS} with that LR. In this way, the distortion due to missing values is attenuated as it will show in the Chapter 7.

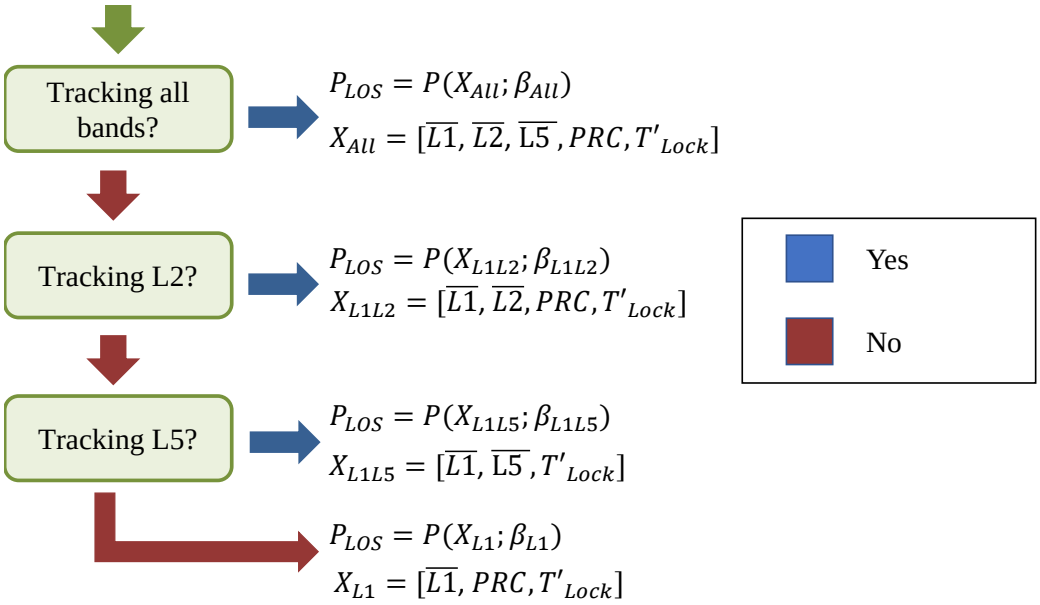


Figure 5.3: Logistic regression decision bounds.

As a fine tune, there is a problem caused because some branches are less likely than the rest. For that reason, their training dataset is dramatically smaller and, because of that, the coefficients of their LR are not perfectly fitted, degrading the performance. It happens, for instance, with the branch of tracking L1 and L5 because L5 is still pre-operational phase and, at the moment of publishing this thesis, only 16 satellites transmit at this frequency [17], so it is more likely to have on track L1 and L2. This can be corrected by applying an asymmetric scheme for training and evaluation. That is to train the branch with all the epochs that have on track L1 and L5 without caring whether L2 is on track or not but, later, evaluate the system with the same scheme shown in Fig. 5.3. This approach maximizes the training dataset utilization.

5.5 Position Domain Application

If the position is computed using a WLS estimator, the classical approach for the weighting matrix $\boldsymbol{W}$ is to use the inverse of the covariance matrix of the error in the pseudorange measurements. This covariance matrix is modelled as a diagonal matrix considering that the error of the measurements is independent between them. Each of those errors is the sum of all the sources of error that are implicated. In particular, a classical model for those errors is [18]:

$$\sigma^2_{classic} = \sigma^2_{URA} + \sigma^2_{ion} + \sigma^2_{tro} + \sigma^2_{nom}; \tag{5.6}$$

where σ^2_{URA} is the variance of the error estimating the satellite position and clock, σ^2_{ion} is the error of the ionospheric model, σ^2_{tro} is the error of the tropospheric correction and σ^2_{nom} is the noise and multipath in open-sky nominal conditions.

As it can be appreciated, sources of error that can appear in challenging scenarios are not taken into account. This limitation degrades the position error as the variance error do not consider them.

In order to include the P_{LOS} probability estimated by the machine learning NLOS detector, it is proposed to include a multiplier to the nominal variance in the following way::

$$\sigma^2_{LOS} = \left(\sigma^2_{URA} + \sigma^2_{ion} + \sigma^2_{tro} + \sigma^2_{nom}\right) e^{1-P_{LOS}}. \tag{5.7}$$

In this way, if P_{LOS} is 1 or close to it, the variance is the same as in the classical model. However, the lower the LOS probability is, the higher the error variance is for that satellite. Hence, it can be concluded that, in some sense, the WLS trusts less in NLOS estimated measurements.

The proposed model was implemented in a PVT estimator already developed by DLR.

Chapter 6

Experimental Methodology

This chapter explain the details about the measurement campaign carried out to collect the data necessary for training and testing the ML algorithms.

Firstly, the experimental installation is explained. Then, it is justified the location selection and details about the measurement campaign are given. Finally, it is presented the labelling methodology carried out.

6.1 Experimental Setup

The installation used to collect the data consisted of an antenna, a tripod and a box for storing the receivers and batteries as shown in Fig. 6.1.

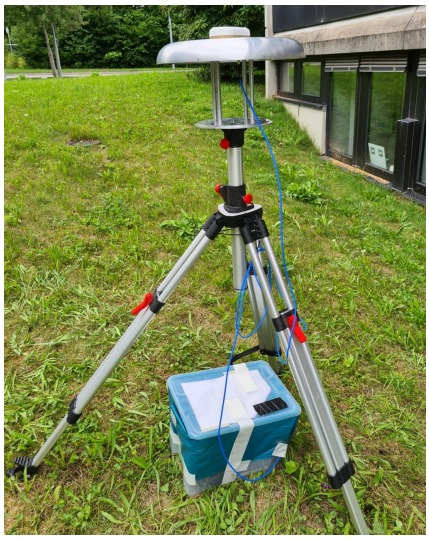


Figure 6.1: Installation for collecting the data.

The antenna is an own design at DLR which includes a metallic surface for avoiding the diffraction. The antenna is set on the tripod which allows leveling and adjusting the height of the installation. The rest of the setup is contained on the box and consists of the elements shown in 6.2.

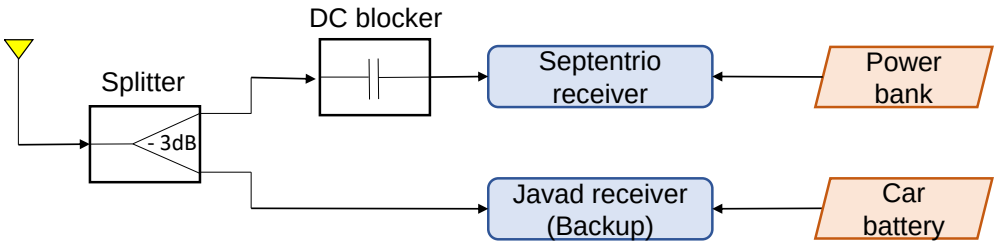


Figure 6.2: Diagram of the installation setup.

As illustrated in the figure, a splitter allows the connection of two GNSS receivers: a *Septentrio mosaic-X5* (Fig. 6.3a) and a *Javad Delta-3* (Fig. 6.3b). Although the work has been carried out with the data collected with the *Septentrio* receiver, it is a good practice in measurement campaign to include a second receiver as backup in case the primary receiver fails.

(a) *Septentrio mosaic-X5*. Resource from: [19].

(b) *Javad Delta-3*.. Resource from: [20].

Figure 6.3: Receivers in the measurement campaign.

As the antennas for GNSS are generally active and the receiver are prepared to feed them, a Direct Current (DC) blocker avoided that both receivers fed the antenna at the same time damaging their own circuits. The DC blocker was set on the *Septentrio* receiver because the car battery that powered the *Javad* has more

battery than the power bank that fed the *Septentrio* receiver. Finally, all this setup was organized in a plastic box as shown in 6.4 in order to protect it against rain.

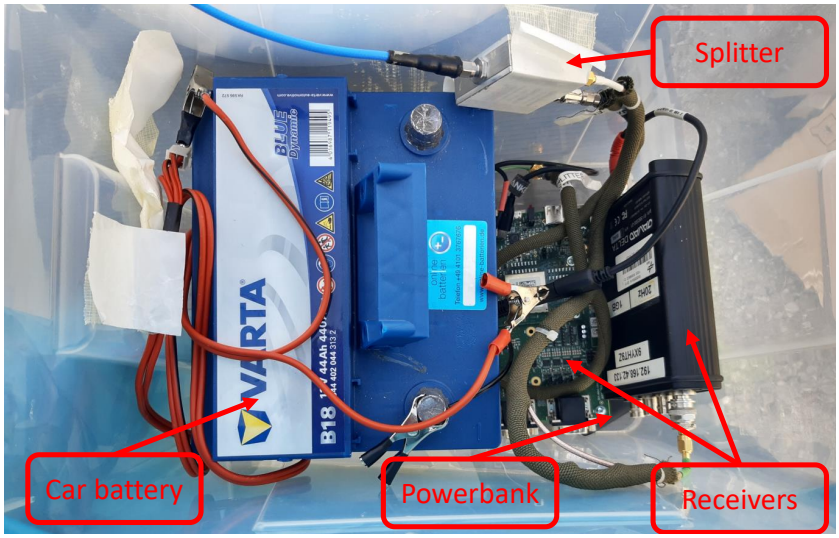


Figure 6.4: Organization of the batteries and receivers.

6.2 Measurement Campaign

A total of 4 different locations were selected to collect data, all of them inside the DLR facilities. In particular, Fig. 6.5 shows those locations in a map.

Figure 6.5: Locations where the data was collected.

On one hand, the calibration point was selected as far as possible of possible obstacles. After pre-analysis of the data by using Code-minus-Carrier (CMC) techniques [21], it is observed that measurements from satellites located at specific azimuth and elevation angles with respect to the receiver, presented additional errors. This can be due to the extra multipath received, probably by reflections of a nearby fence. Since the azimuth and elevation angles with affected measurements were easily identifiable, the data is pre-screened to remove the affected measurements from the rest of the data processing. Like that, it is ensured that the measurements are well representative of the expected open-sky scenario.

On the order hand, the challenging scenarios were searched trying to find locations next to buildings whose horizon is possible to model using the algorithm of Section 5.2. In contrast, locations with trees or other complex elements difficult to models were avoided as much as possible.

A precise positioning of the receivers in the location was needed to compute the ground truth and to test the effect of the LOS/NLOS detector in the position. However, the PVT has a poor precision in these challenging scenarios with an error in the level of tens of meters. The solution is to use Precise Point Positioning (PPP) estimators which, in a post-processing stage, are able to compute the position with a decimeter level of precision using Kalman Filters and precise products for the satellite position and clock. In this case, *Inertial Explorer* was the software used for this purpose.

It has been recorded on all the locations continuous streams of more than 24 hours in order to have, at least for GPS, the full repeating orbit cycle.

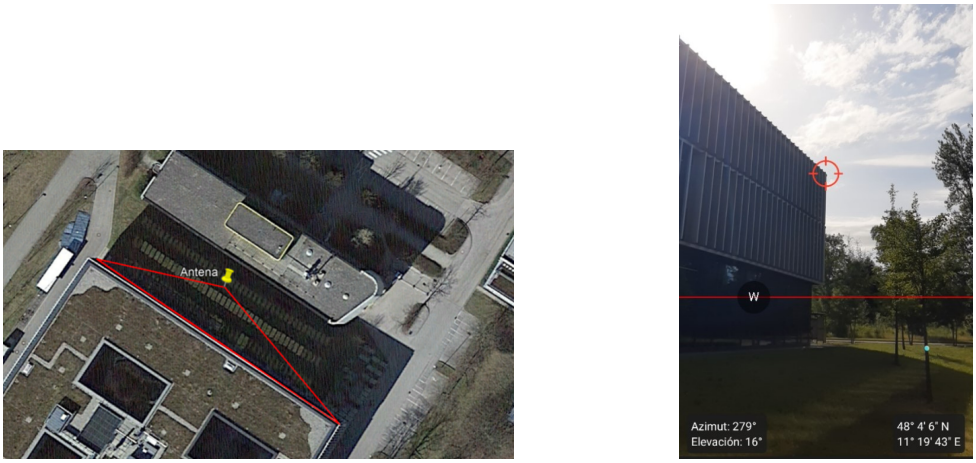
6.3 Horizon Estimation Methodology

As explained in Section 5.2, the horizon estimation for each building requires to measure its width as well as the distances and angles to its corners.

The first approach was to follow [8] where it is proposed to use a mobile application for measuring the elevation at each building corner. For this purpose, it was used an application called *Satellite Pointer* [22]. In the case of the distance and the width of the building, it can be provided by *Google Earth*. An example of how both tools have been used is shown in Fig. 6.6.

This methodology can be categorize as low cost. However, the resolution of the application is one degree so it is not very precise. Moreover, it is difficult to measure with precision the distances in *Google Earth* because the image resolution was poor.

In order to obtain better measurements, a second approach was carried out. It was based on a *Leica TPS1200* tachymeter which deliver measurements with a centimeter level precision. The tachymeter is able to measure angles and distances between a reference station which was located in the tripod and a prism. In Fig.

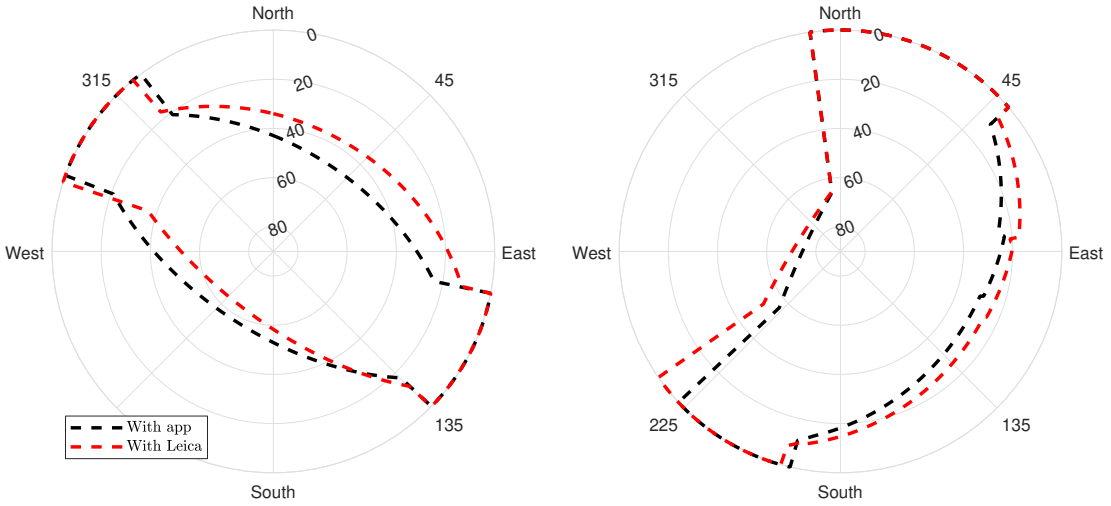


(a) Measuring with *Google Earth*. (b) Mesuring with *Satellite Pointer*.

Figure 6.6: First approach for the horizon estimation.

6.7, it is shown how corners are measured.

Processing these measurements, it is obtained the estimated horizon of each scenario. For instance, the panoramic view for the receiver in the *Training Location* and in the *Testing Location 2* are shown in Fig. 6.8 while Fig. 6.9 is shown the estimated horizons for both scenarios.



(a) *Training Location*. (b) *Testing Location 2*.

Figure 6.9: Skyplot of the estimated horizons.

The results in the skyplots intuitively agree with the panoramic views. Furthermore, the estimations using both approaches are approximately similar. That

(a) Reference station pointing to the prism. (b) The prism set on the corner of a building.

Figure 6.7: Second approach for the horizon estimation.

(a) *Training Location.*

(b) *Testing Location 2.*

Figure 6.8: Panoramic view from the receivers.

means that the low cost methodology with the app has a relatively good performance. However, the precision with the tachymeter is supposed to be higher so for the rest of the thesis this will be the labelling methodology.

Chapter 7

Results and Evaluation

The performance of the LOS/NLOS detector and its application in the PVT solver will be presented in this chapter.

First, the nominal C/N0 model is obtained, and some insights are extracted. A comparison between the main ML algorithm developed in this thesis based on logistic regressions is compared with the ones found in the literature. Finally, the impact on the position domain of using the LR to weight measurements in the PVT is evaluated.

7.1 Nominal Carrier-To-Noise Ratio Model

The nominal C/N_0 is extracted from the 24 hours open-sky data recorded during the measurement campaign with the same installation used in the challenging scenarios. After processing the data, the statistics $\mu_j(\theta)$ and $\sigma_j(\theta)$ presented in Section are obtained. For instance, in Fig. 7.1 it is presented the raw C/N_0 for L1 as a function of the elevation as well as the resulting mean and standard deviation.

As it is shown, due to the antenna radiation pattern and other factors, the $\mu_j(\theta)$ increases with the elevation while the $\sigma_j(\theta)$ decreases. Considering this behavior, a low C/N_0 would be less significant of NLOS for low elevation satellites. This justifies the necessity of the normalization.

These statistics are different for each GNSS signal due to multiple factors such as the antenna gain at each frequency and the different modulations of each signal. For instance, in Fig. 7.2 the mean value obtained for each GNSS signal used in this work is shown.

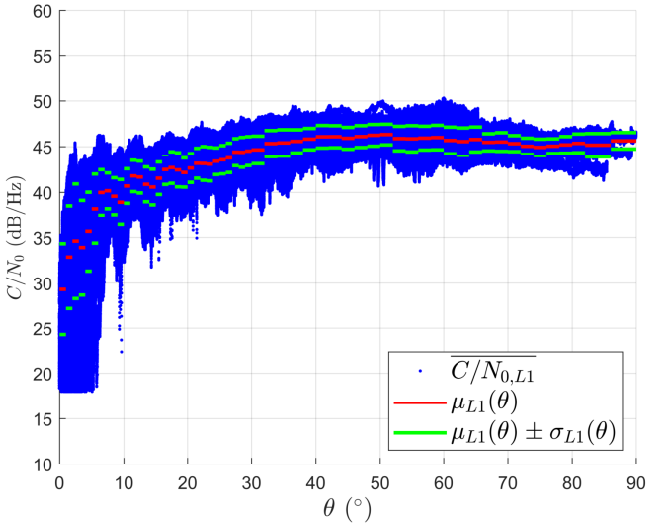


Figure 7.1: Nominal C/N_0 for L1.

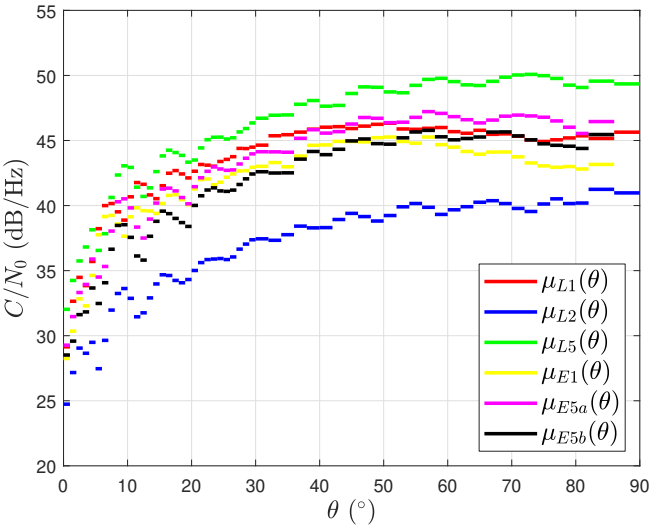


Figure 7.2: $\mu_j(\theta)$ for all the signals of interest.

7.2 Feature analysis

The performance of ML algorithms are strongly related to the behavior of the features for each label. Therefore, it is interesting to examine how different are those behaviors for each class.

For instance, in Fig. 7.3, it is shown the estimated PDFs of $C/N_{0,L1}$ for LOS and NLOS labelling with the horizon obtained with Leica and the mobile app. Although it is difficult to evaluate the quality of the labelling, it is possible to appreciate that the shape of both PDFs is approximately the same as expected in [8] which based its explanation on [23]. This behavior is the same for other GNSS signals. As a last conclusion of this figure, the shape of these PDFs allow concluding that a high C/N_0 will be most likely from LOS signals while the opposite is not as clear.

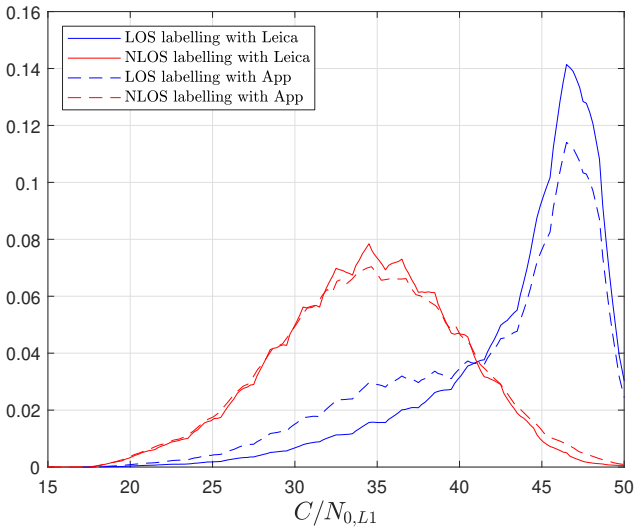


Figure 7.3: The PDF of $C/N_{0,L1}$ for LOS and NLOS signals.

In the case of the PRC, in Fig. 7.4 is presented the evolution of this feature on the dataset for the satellite PRN=24 of GPS. In the figure, it is clearly shown that for NLOS signals the range of values for the PRC is larger than for LOS. However, low values of PRC are shared by LOS and NLOS signals so only this feature is not enough to detect NLOS.

In general, the information that each feature give about the signal nature is not enough for classifying it. However, the combination of all of them in the ML algorithm can potentially manage the problem.

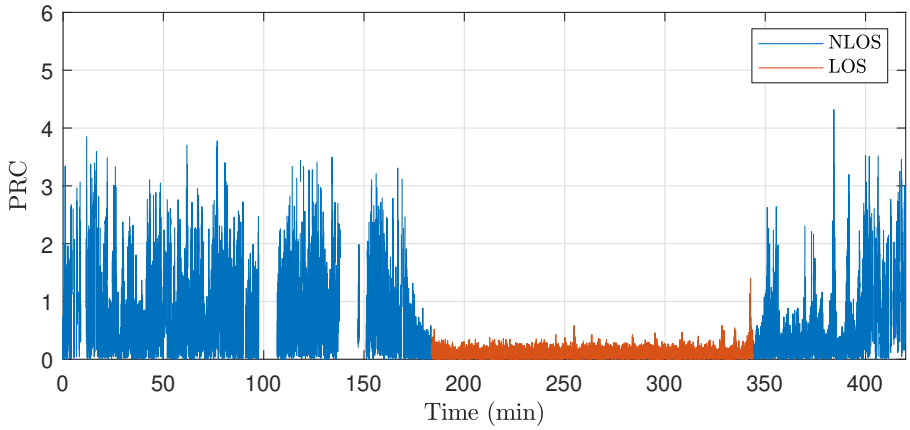


Figure 7.4: Evolution of the recorded PRC for the satellite PRN = 24

7.3 Performance of the Detectors in the Literature

Before presenting the performance of the final proposed detector, it is going to be presented some tests performed with state-of-the-art designs. In particular, the most used ML algorithms in the state-of-the-art are the decision tree and the linear SVM.

These state-of-the art algorithms have been applied without normalization and only considering the C/N_0 of one band as it is done in the literature. Therefore, the feature vector for them in the case of GPS is:

$$X_{GPS} = [C/N_{0,L1}, T'_{lock}, \overline{PRC}] \tag{7.1}$$

In Table 7.1 and Table 7.2, the confusion matrix from the training phase is presented for GPS and Galileo respectively. As it is observed in both tables, the decision tree is slightly better than the SVM in terms of accuracy. However, the SVM has a more balanced distribution of the error. This is most noticeable for the case of GPS.

Ground Truth	SVM detector		Decision tree detector	
	LOS	NLOS	LOS	NLOS
LOS	76%	24%	83%	17%
NLOS	14%	86%	21%	79%
Accuracy	80.8%		82.1%	

Table 7.1: Confusion matrix for the state-of-art algorithms for GPS.

Ground Truth	SVM detector		Decision tree detector	
	LOS	NLOS	LOS	NLOS
LOS	86%	14%	76%	24%
NLOS	21%	79%	17%	83%
Accuracy	80.6%		81.2%	

Table 7.2: Confusion matrix for the state-of-art algorithms for Galileo.

7.4 Logistic Regression Performance For GPS

In this section, the detectors proposed in this work are evaluated. In particular, the effect of the normalization, the scheme of branches carried out for handling the C/N_0 of multiple bands and the overall system performance are analyzed.

The scheme of branches regarding the tracking signals is going to be compared with the most typical MNAR solution. This state-of-art solution consists in using just one LR for all the cases defining a default value in case of not tracking one of the signals. This model was explained in detail in 5.4.

As a first step, it is presented in Table 7.3 the confusion matrix of the single LR with default values named from now on as the classic LR with and without normalization. In the same way, it is shown for the branched LR in the Table 7.4 For evaluating the accuracy of the models between them, it has been used the same scenario for training and validating and the data has been labelled using Leica.

Ground Truth	Classic LR (not norm.)		Classic LR (norm.)	
	LOS	NLOS	LOS	NLOS
LOS	85%	15%	87%	13%
NLOS	10%	90%	19%	81%
Accuracy	87.7%		84.3%	

Table 7.3: Confusion matrix for the classic LR

The results from the tables show that the inclusion of the C/N_0 from additional bands clearly outperform the state-of-art proposals. Moreover, the proposed branched scheme for the LR manages better the situation when one or more signals are not on track.

Regarding the normalization, the LR works worse with it than without it in the scenario where the models were trained. However, it is generally expected that a model for specific conditions works better in those conditions than a generalized

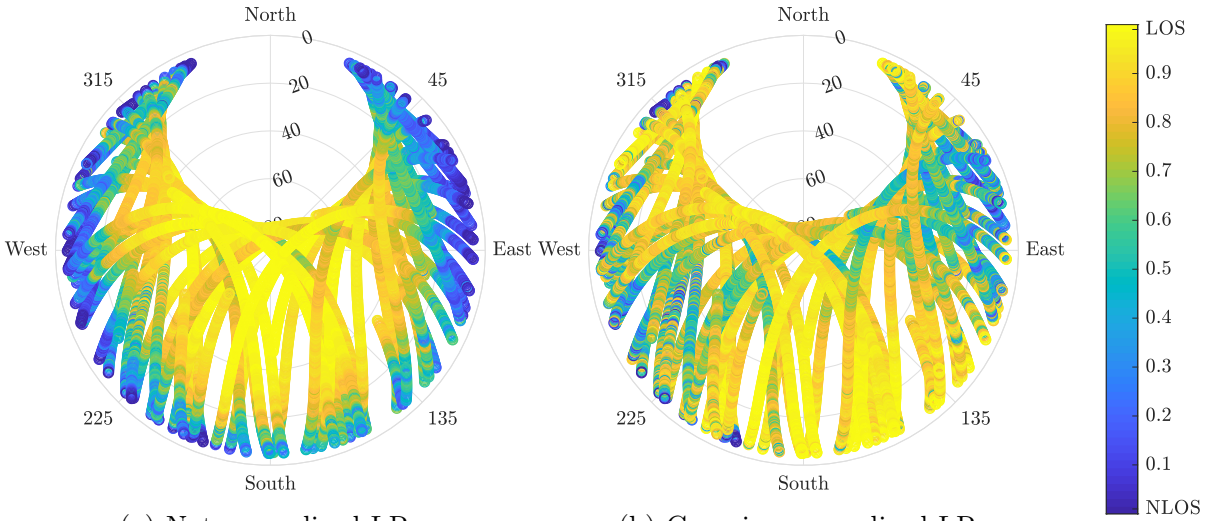
Ground Truth	Branched LR (not norm.)		Branched LR (norm.)	
	LOS	NLOS	LOS	NLOS
LOS	92%	8%	89%	11%
NLOS	14%	86%	14%	86%
Accuracy	89.0%		87.6%	

Table 7.4: Confusion matrix for the branched LR.

model. If the environment or conditions change, then it is expected that the generalized models outperform the not normalized ones.

In order to justify the aforementioned explanation, both cases are compared in an open-sky scenario where the LOS probability should always be $p_{LOS} > 0.5$. The confusion matrix for this test is exhibited in Table 7.5. Moreover, in order to extract insights from the model in a more visual way and to examine the result not only as a classification problem but also as a regression problem, Fig. 7.5 shows the estimated LOS probability in that open-sky scenario. As expected, the not normalized version consider NLOS satellites as those with low elevation. This is caused by the reduction of the C/N_0 just because of the elevation shown in Section 7.1.

It is important to comment that all the skyplots and tests show from now on have been carried out using 24 hours continous recordings. In this way, the full cycle of GPS orbits are exposed but not the complete Galileo orbits, which require more days.



(a) Not normalized LR. (b) Gaussian normalized LR.

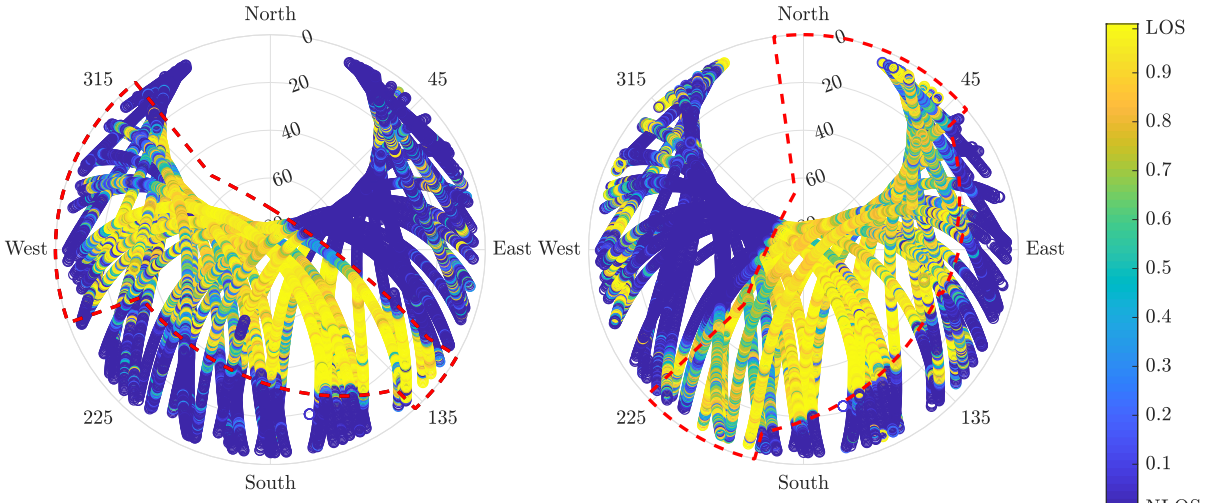
Figure 7.5: Estimated LOS probability using LR in open-sky for GPS.

Ground Truth	Branched LR (not norm.)		Branched LR (norm.)	
	LOS	NLOS	LOS	NLOS
LOS	74%	26%	91%	9%
NLOS	-	-	-	-
Accuracy	74.2%		90.7%	

Table 7.5: Confusion matrix of the branched LR in open-sky for GPS

As it is shown numerically in Table 7.5, the normalized branched LR clearly outperforms the not normalized one. The difference in terms of accuracy could potentially be even higher but the scenario for recording the data is not a perfect open-sky scenario. In particular, the east area on the skyplot of Fig. 7.5b shows a lower P_{LOS} than it should be. This area is the same from which some satellites had to be removed as explained in Section 6.2 due to a fence and, sometimes, airplanes.

In order to check the performance of the proposed model in different scenarios, the detector has been tested in other different scenarios which are more challenging than the one used for training. The skyblock for those scenarios are shown in Fig. 7.6 and the table 7.6 contains their confusion matrix.



(a) *Testing Location 1.* (b) *Testing Location 2.*

Figure 7.6: Estimated LOS probability using LR in different scenarios for GPS.

As it is observed in the figures, the algorithm works well in different scenarios. However, the accuracy is lower than it was in the scenario where the model was trained. It could be caused by different factors such as a higher error labelling the data due to the difficulties for modelling complex scenarios. More precisely, in the

Ground Truth	*Testing Location 1*		*Testing Location 2*	
	LOS	NLOS	LOS	NLOS
LOS	94%	6%	88%	12%
NLOS	30%	70%	20%	80%
Accuracy	84.1%		91.4%	

Table 7.6: Confusion matrix of the branched LR in challenging scenarios for GPS.

case of Fig. 7.6a, the errors at low elevation on the west could be caused by the trees and elements in that area. Despite the effort done in the location search during the measurement campaign, it has been impossible to avoid all the elements which horizon modelling was extremely complicated to do. Therefore, this problem has inevitably distorted some results.

As the last point in this section, some insights related to how the model behaves with features changes will be presented. In particular, it is interesting, considering two features of interest and the rest fixed, to find the regions in which the classification problem consider LOS or NLOS. For this purpose, it is necessary to derive the Eq. (3.2) considering $p_{LOS} = 0.5$. Under that restriction, it is obtained the following equation:

$$X_{GPS}\beta = 0 \tag{7.2}$$

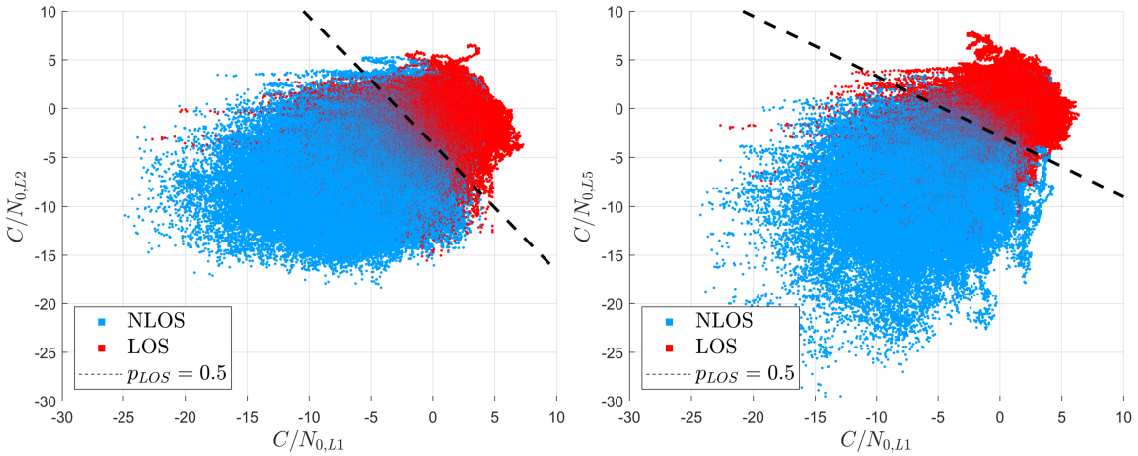
For visualizing this hyper-surface, it is necessary to fix some of the features. The criteria decided for it is to use the mean in those terms. The result is shown in Fig. 7.7 where the training dataset is plotted as points, and the black line is the separation that makes the LR between a LOS decision and a NLOS decision. In particular, in Fig. 7.7a it is tested the branch for the case of tracking only L1 and L2, while in Fig. 7.7b it is tested for the case of only L1 and L5.

The figure shows how, despite some outliers, most of the points are correctly split into two regions. Moreover, observing the slant of the $p_{LOS} = 0.5$ line, it is appreciated that all the C/N_0 are useful for differentiating between LOS and NLOS.

7.5 Logistic Regression Performance For Galileo

In this section, the performance of the model for Galileo will be examined. As the spectral shape and the modulation of each band is different from GPS, the model can not have the same coefficients for both constellations.

The first evaluation is done with the validation dataset in the same scenario where it is trained. This time, in order to simplify the results, it has been considered that



(a) Branch of L1 and L2.

(b) Branch of L1 and L5.

Figure 7.7: Decision areas for different branches of the model.

the branched scheme outperforms the scheme of just one LR with default values. In table 7.7, it is shown the resulting confusion matrix.

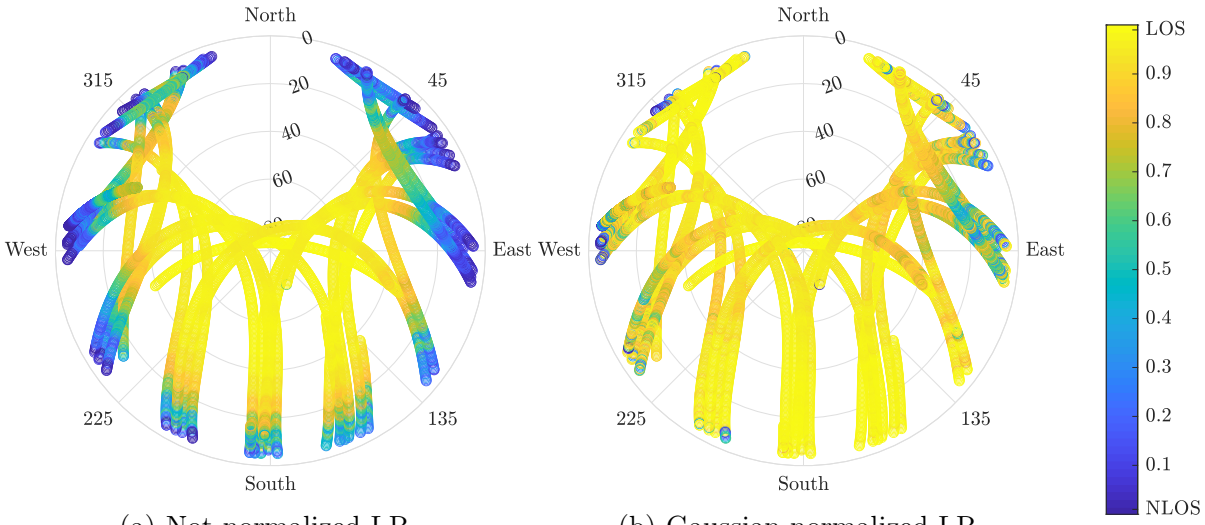
Ground Truth	Branched LR (not norm.)		Branched LR (norm.)	
	LOS	NLOS	LOS	NLOS
LOS	94%	6%	91%	9%
NLOS	11%	89%	14%	86%
Accuracy	91.5%		88.9%	

Table 7.7: Confusion matrix in validation for the Galileo detector.

As it is appreciated, the LOS/NLOS detector is slightly better for Galileo than for GPS. The reason for that could be associated with the bands used in each constellation. Nevertheless, the difference is not very significant.

As it was done with GPS, in Fig. 7.8 it is compared the open sky LOS probability estimation using normalization and not using it. In the same way, it is presented in table 7.8 the confusion matrix for both cases. These results show the necessity of the normalization for correcting the C/N_0 considering the elevation as happened in GPS.

In the same way, the Fig. 7.9 illustrates the skyplot of the estimated LOS probability in challenging scenarios. Moreover, the confusion matrix for those cases are presented in the table 7.9. As it was explained in the case of GPS the estimation in Fig. 7.9a is affected by the elements complex to model. Although this problem is attenuated because it is less usual that orientation for the orbits of Galileo satellites,



(a) Not normalized LR. (b) Gaussian normalized LR.

Figure 7.8: Estimated LOS probability using LR in open-sky for Galileo.

Ground Truth	Branched LR (not norm.)		Branched LR (norm.)	
	LOS	NLOS	LOS	NLOS
LOS	77%	23%	98%	2%
NLOS	-	-	-	-
Accuracy	76.6%		97.9%	

Table 7.8: Confusion matrix of the branched LR in open-sky for Galileo.

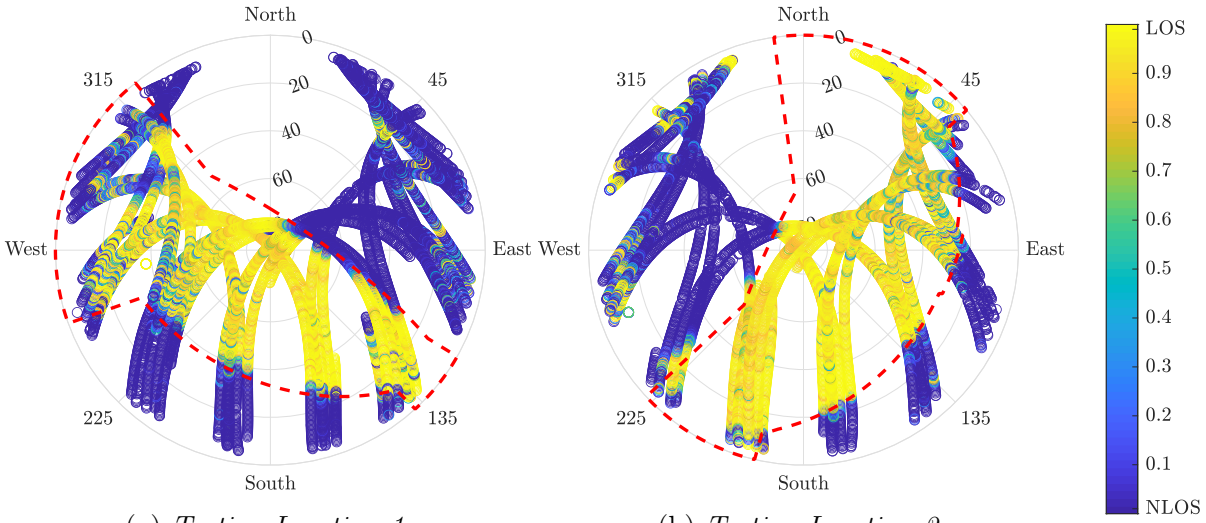
it still produces an increment in false negatives in that direction. Nevertheless, in both cases, the model predicts the horizon of the building correctly.

7.6 Impact at Position Domain

After showing the performance of the model that has been carried out, this section will show how this model has been used for improving the final position in a PVT estimator. For this analysis, a 6 hours GNSS recording will be the test dataset.

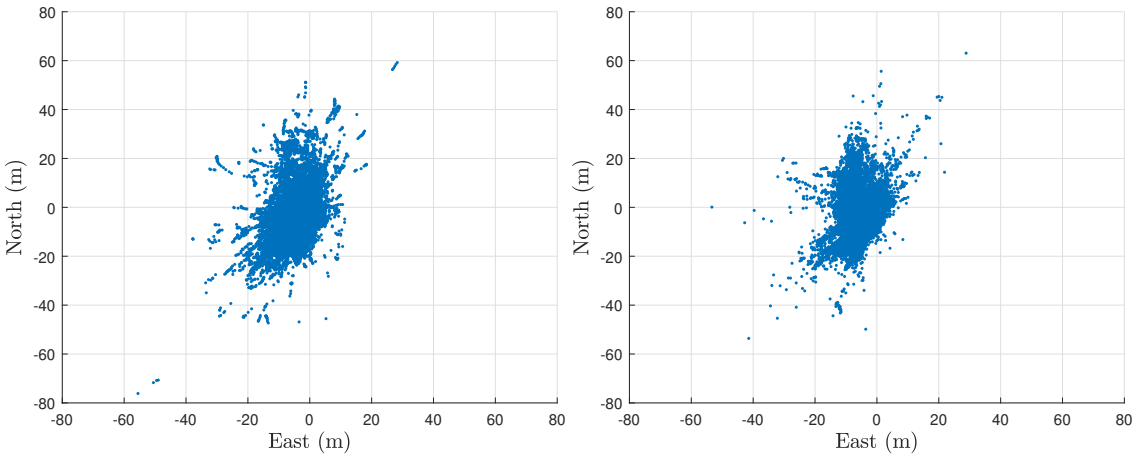
First of all, in Fig. 7.10 it is illustrated a comparison of the position error using only GPS in the case of the classic error variance shown in Section 2.3 and the case of the proposed model.

As it is appreciated in Fig. 7.10, increasing the error variance for the NLOS satellites through the developed model reduces the number of outliers. However,



(a) *Testing Location 1.* (b) *Testing Location 2.*

Figure 7.9: Estimated LOS probability using LR in different scenarios for Galileo.



(a) Classic variance model. (b) LOS variance model.

Figure 7.10: Scatterplot of the position error for GPS.

Ground Truth	*Testing Location 2*		*Testing Location 2*	
	LOS	NLOS	LOS	NLOS
LOS	95%	5%	92%	8%
NLOS	24%	76%	10%	90%
Accuracy	84.7%		91.4%	

Table 7.9: Confusion matrix of the branched LR in challenging scenarios for Galileo.

this improvement is even more apparent in the standard deviation of the error in each axis as shown in Table 7.10. Moreover, in this way, it is shown how the model also improves the error in the height axis. This axis is generally the one with the higher error because the LS solver is ill-conditioned. The reason for it is geometrical considering that all the satellites are above the receiver.

	North (m)	East(m)	Height(m)
Classic model	4.78	9.89	13.73
LOS model	4.13	8.47	11.84
Improvement	13.62%	14.39%	13.75%

Table 7.10: Standard deviation of the position error using GPS.

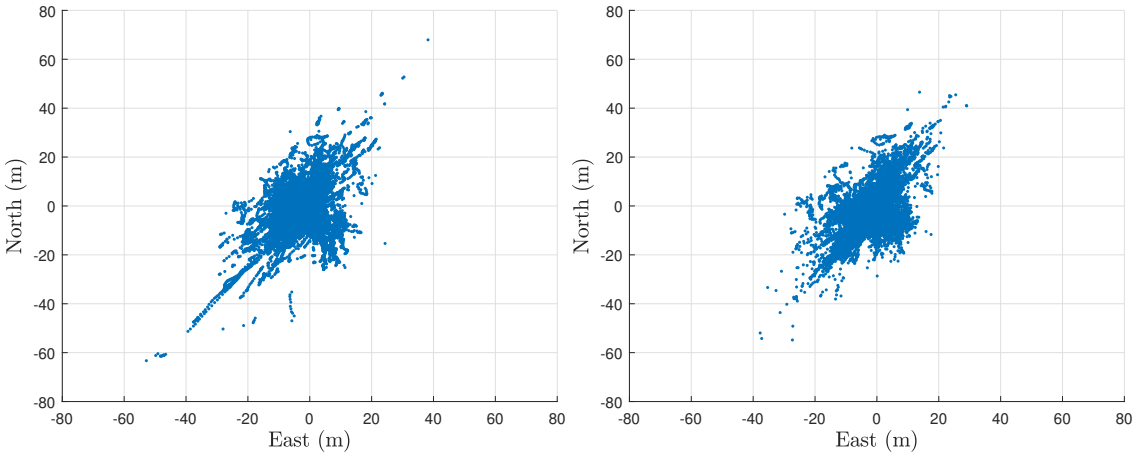
As it is shown in the table, the proposed model improves the position error in the three axis.

In the case of Galileo, it is possible to apply the same procedure with its respective model. In this way, the improvement in the position computation is shown in Fig. 7.11 and in the Table 7.11.

	North (m)	East(m)	Height(m)
Classic model	6.84	9.75	14.46
LOS model	5.95	8.23	14.07
Improvement	12.96%	15.62%	2.67%

Table 7.11: Standard deviation of the position error using Galileo.

The Table 7.11 reflects a more imbalanced correction of the error than in the case of GPS but, in any case, the results still show an improvement in all the axis when the estimated LOS probability is included. Therefore, these results validate the designed model for the error variance in the weighted LS.



(a) Classic variance model.

(b) LOS variance model.

Figure 7.11: Scatterplot of the position error for Galileo.

Chapter 8

Achievements, Conclusions and Outlook

8.1 Achievements

In this thesis, a LOS/NLOS detector based on Machine Learning for GNSS positioning has been developed. The system extracts some robust features from the observables of Galileo and GPS receivers and is able to predict a probability of a satellite measurement to come from a line of sight satellite signal. This detector has been tested using different challenging scenarios to prove that is robust and well generalized. Furthermore, a methodology to include the LOS/NLOS detector in the PVT estimation has been proposed and tested.

To carry out this thesis, some tasks have been achieved in different domains. Fig. 8.1 shows the main contributions of this thesis with respect to the most important areas of work as well as an indication of the previous work and material available at DLR.

Among all these tasks, it is possible to point out the most important achievements:

- Survey and critical analysis of the current state of the art.
- Definition of a precise labelling methodology for LOS/NLOS truth.
- Design robust features with a normalization process.
- Inclusion the detector in the PVT estimator.

First, after a deep study of the current state of the art, it has been observed that most of the proposals used questionable features. Algorithms with those features

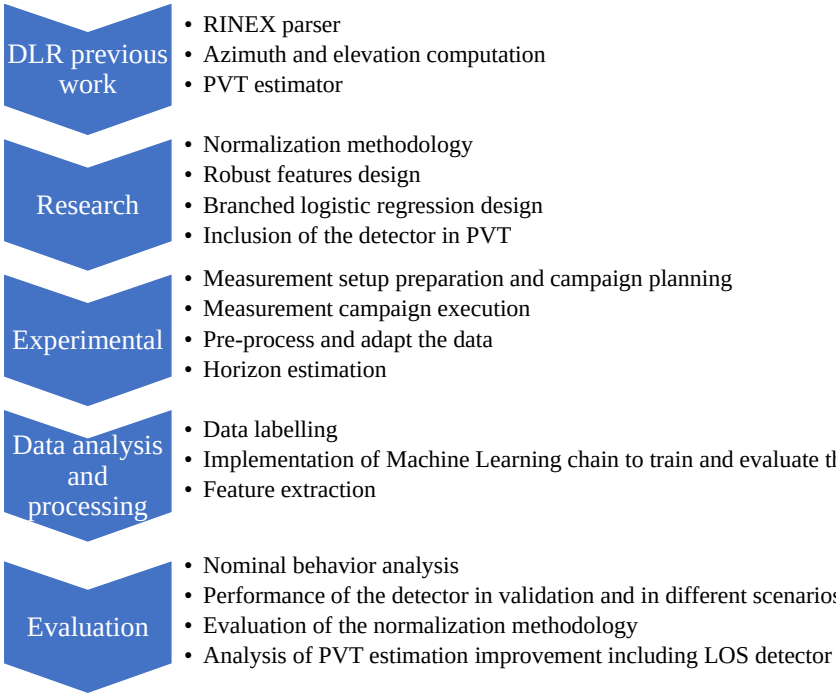


Figure 8.1: Summary of the achieved tasks.

could work in the training scenario but it is debatable that the performance remains the same in a different one.

Second, a labelling methodology based on two different measurement devices has been carried out. Although the low-cost solution based on a mobile application is less precise, it has been shown that the estimated horizon is not far from the one obtained with the tachymeter.

Third, the proposed LOS/NLOS detector outperforms the state of the art, even in the same scenario of the training. In particular, it has been obtained an accuracy of 87.6% for GPS and 88.9% for Galileo. As the C/N_0 is the key feature, including some observables from more frequencies gives a valuable extra information. The use of multiple frequencies presented new challenges. In particular the fact that some frequency observables are sometimes not available. A new approach has been develop to handle this situation based on a branched logistic regression algorithm. Furthermore, the utilization of logistic regressions is more potent for some applications as it provides a continuous range of values instead of a single binary decision.

Additionally, the nominal model obtained empirically has demonstrated that the C/N_0 is conditioned by the elevation. Furthermore, the normalization of the C/N_0 has been demonstrated as a valuable way to increase the applicability of the detector for different scenarios.

The effort to increase the robustness has allowed to keep a good performance changing the scenario. In both testing scenarios, the detectors for GPS and Galileo show a similar performance to the one in validation.

Finally, a methodology has been developed to include the output of the detector in the PVT estimation. This idea of modelling the covariance matrix to consider the predictions has a huge potential and can be used for other detectors. In particular, the proposed model for the pseudorange error variance, including the LOS probability as a variance multiplier has improved the positioning error for both constellations.

8.2 Outlook

Some future improvements and further analysis for the work in this thesis can be carried out:

- *Testing the detector with different installations*: It has been shown that the algorithm is well generalized for different scenarios. It is expected that the normalization also serves to make the detector valid for different antenna and receivers. However, it has not been tested.
- *A deeper study of the model for the pseudorange error variance*: It has been shown that the inclusion of the detector output in the WLS improves the positioning error. However, the model for the variance with the LOS probability was not the main target of this work. Hence, a deeper study of this topic might allow the development of a better one.
- *Test the detector with dynamic measurements*: Labelling the data in a dynamic scenario is challenging. However, it would be interesting to evaluate the positioning error in a dynamic scenario with the WLS including the LOS/NLOS detector.

Additionally, some new research directions can address:

- *LOS/NLOS detector based on Deep Learning*: Deep Learning is a variant of Machine Learning which is more powerful than Machine Learning but requires more data for the training. Its application as the algorithm for the detector can improve the overall performance.
- *Multipath detector*: It would be interesting to develop a predictor for the multipath variance since it is more straightforward to include it in the pseudorange variance error.

Juan Carlos Ruiz Sicilia
September 24, 2021

Capítulo 9

Logros, Conclusiones y Propuestas Futuras

9.1 Logros

En esta tesis se ha desarrollado un detector LOS/NLOS basado en Machine Learning para posicionamiento GNSS. El sistema extrae algunas características robustas de los observables de los receptores Galileo y GPS y es capaz de predecir la probabilidad de que una medición de satélite provenga de una señal de satélite en línea de visión. Este detector ha sido probado utilizando diferentes escenarios desafiantes para demostrar que es robusto y bien generalizado. Además, se ha propuesto y probado una metodología para incluir el detector LOS/NLOS en la estimación PVT.

Para llevar a cabo esta tesis, se han realizado algunas tareas en diferentes dominios. En la Fig. 9.1 se muestran las principales aportaciones de esta tesis con respecto a las áreas de trabajo más importantes, así como una indicación de los trabajos previos y el material disponible en el DLR.

De entre todas estas tareas destacan los siguientes logros:

- Recolección y análisis del estado del arte actual.
- Definición de una metodología precisa para etiquetar los datos como LOS o NLOS.
- Diseño the características robustas mediante un proceso de normalización.
- Inclusión del detector en el estimador de PVT.

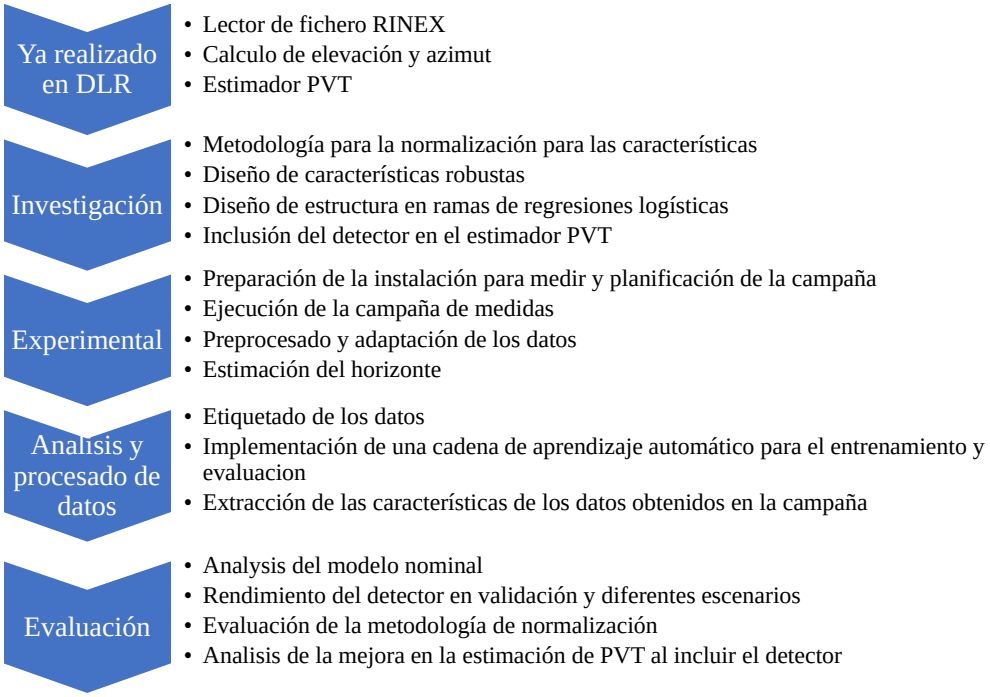


Figura 9.1: Resumen de las tareas realizadas.

9.2 Conclusiones

En primer lugar, tras un profundo estudio del estado actual de la técnica, se ha observado que la mayoría de las propuestas utilizan características cuestionables. Los algoritmos con esas características podrían funcionar en el escenario de entrenamiento, pero es discutible que el rendimiento siga siendo el mismo en uno diferente.

En segundo lugar, se ha llevado a cabo una metodología de etiquetado basada en dos dispositivos de medición diferentes. Aunque la solución de bajo coste basada en una aplicación móvil es menos precisa, se ha demostrado que el horizonte estimado no dista mucho del obtenido con el taquímetro.

En tercer lugar, el detector LOS/NLOS propuesto supera al estado del arte, incluso en el mismo escenario del entrenamiento. En particular, se ha obtenido una precisión del 87,6 % para GPS y del 88,9 % para Galileo. Dado que el C/N_0 es la característica clave, la inclusión de algunos observables de más frecuencias proporciona una valiosa información extra. El uso de múltiples frecuencias presenta nuevos retos. En particular, el hecho de que algunas frecuencias observables a veces no están disponibles. Se ha desarrollado un nuevo enfoque para manejar esta situación basado en un algoritmo de regresión logística ramificada. Además, la utilización de regresiones logísticas es más potente para algunas aplicaciones, ya que proporciona

un rango continuo de valores en lugar de una única decisión binaria.

Por otra parte, el modelo nominal obtenido empíricamente ha demostrado que la C/N_0 está condicionada por la elevación. Además, la normalización de la C/N_0 se ha demostrado como una herramienta útil para aumentar la aplicabilidad del detector para diferentes escenarios.

El esfuerzo por aumentar la robustez ha permitido mantener un buen rendimiento cambiando de escenario. En ambos escenarios de prueba, los detectores para GPS y Galileo muestran un rendimiento similar al de la validación.

Finalmente, se ha desarrollado una metodología para incluir la salida del detector en la estimación de PVT. Esta idea de modelar la matriz de covarianza para considerar las predicciones tiene un enorme potencial y puede ser utilizada para otros detectores. En concreto, el modelo propuesto para la varianza del error de pseudorango, incluyendo la probabilidad LOS como multiplicador de la varianza, ha mejorado el error de posicionamiento para ambas constelaciones.

9.3 Propuestas Futuras

Algunas mejoras y analisis adicionales sobre el trabajo realizado en esta tesis podrían ser:

- *Verificar el rendimiento del dectector con diferents instalaciones*: Se ha demostrado que el algoritmo está bien generalizado para diferentes escenarios. Se espera que la normalización también sirva para que el detector sea válido para diferentes antenas y receptores; sin embargo, esto no se ha probado.

- *Un estudio más profundo sobre el modelo de la varianza del error del pseudorango*: Se ha demostrado que la inclusión de la salida del detector en el WLS mejora el error de posicionamiento. Sin embargo, el modelo para la varianza incluyendo la probabilidad de LOS no era el objetivo principal de este trabajo. Por lo tanto, un estudio más profundo de este tema podría permitir el desarrollo de uno mejor.

- *Verificar el rendimiento del detector en escenarios dinámicos*: Etiquetar los datos en un escenario dinámico es complejo. Sin embargo, sería de gran interés evaluar el error de posicionamiento en un escenario dinámico con el WLS incluyendo el detector LOS/NLOS.

Por otra parte, algunas nuevas lineas de investigación podrían ser:

- *Diseño de un detector de LOS/NLOS basado en Deep Learning*: *Deep Learning* es una variante *Machine Learning* que puede llegar a resultar más potente que

esta si es entrenada con gran cantidad de datos. Su aplicación para detectar señales NLOS podría mejorar el rendimiento del detector.

- *Detector de señales multicamino*: Sería interesante desarrollar un predictor para la varianza del error multicamino ya que este detector sería notablemente más fácil de incluir en el modelo de la varianza de error del pseudorango.

Bibliography

[1] G. S. ICD, "Galileo open service signal in space interface control document," 2010, http://galileognss.eu/wp-content/uploads/docs/galileo-os-sis-icd-issue1-revision1_en.pdf, Visited at: 07/09/2021.

[2] J. J. Z. J. Sanz Subirana and M. Hernández-Pajares, *GNSS Data Processing, Vol. 1: Fundamentals and Algorithms*. ESA Communications, 2010.

[3] J. A. Klobuchar, "Ionospheric time-delay algorithm for single-frequency gps users," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-23, no. 3, pp. 325–331, 1987.

[4] T. M. Mitchell, "Machine learning," 1997.

[5] R. Collobert, K. Kavukcuoglu, and C. Farabet, "Torch7: A matlab-like environment for machine learning," in *BigLearn, NIPS Workshop*, 2011.

[6] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[7] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux, "API design for machine learning software: experiences from the scikit-learn project," in *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 2013, pp. 108–122.

[8] R. Yozevitch, B. Ben-Moshe, and A. Weissman, "A robust gnss los/nlos signal classifier: Gnss shadow matching algorithms," *Navigation*, vol. 63, pp. 429–442, 12 2016.

[9] 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)*, 2020, pp. 1–10.

[10] E. Shafiee, M. R. Mosavi, and M. Moazedi, "Detection of spoofing attack using machine learning based on multi-layer neural network in single-frequency GPS receivers," *Journal of Navigation*, vol. 71, no. 1, pp. 169–188, aug 2017.

[11] R. Imam and F. Dovis, "Distinguishing ionospheric scintillation from multipath in GNSS signals using bagged decision trees algorithm," in *2020 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)*. IEEE, oct 2020.

[12] Y. Xia, S. Pan, X. Meng, W. Gao, F. Ye, Q. Zhao, and X. Zhao, "Anomaly detection for urban vehicle GNSS observation with a hybrid machine learning system," *Remote Sensing*, vol. 12, no. 6, p. 971, mar 2020.

[13] H. Xu, A. Angrisano, S. Gaglione, and L.-T. Hsu, "Machine learning based LOS/NLOS classifier and robust estimator for GNSS shadow matching," *Satellite Navigation*, vol. 1, no. 1, may 2020.

[14] L. Hsu, "Gnss multipath detection using a machine learning approach," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017, pp. 1–6.

[15] R. Sun, G. Wang, W. Zhang, L.-T. Hsu, and W. Ochieng, "A gradient boosting decision tree based gps signal reception classification algorithm," *Applied Soft Computing*, vol. 86, p. 105942, 11 2019.

[16] S. van Buuren, *Flexible Imputation of Missing Data*. Chapman and Hall/CRC, 2018.

[17] U. S. Force, "Civil Navigation (CNAV) Message," 2021, https://www.gps.gov/systems/gps/modernization/cnav/. Visited at: 20/9/2021.

[18] O. G. Crespillo, A. Andreetti, and A. Grosch, "Design and evaluation of robust m-estimators for gnss positioning in urban environments," in *Proceedings of the 2020 International Technical Meeting of The Institute of Navigation, San Diego, CA, USA*, 2020, pp. 21–24.

[19] Septentrio, "Mosaic-X5," 2021, https://www.septentrio.com/. Visited at: 20/9/2021.

[20] Javad, "Delta-3," 2021, https://www.javad.com/. Visited at: 20/9/2021.

[21] M. Caamano, O. G. Crespillo, D. Gerbeth, and A. Grosch, "Detection of gnss multipath with time-differenced code-minus-carrier for land-based applications," in *2020 European Navigation Conference (ENC)*. IEEE, 2020, pp. 1–12.

[22] Cappsule, "Satellite Pointer," 2021, https://www.satellitepointer.com/. Visited at: 20/9/2021.

[23] A. T. Irish, J. T. Isaacs, F. Quitin, J. P. Hespanha, and U. Madhow, "Belief propagation based localization and mapping using sparsely sampled gnss snr measurements," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 1977–1982.