

**POLITECNICO DI TORINO**

**Master's Degree in ICT for Smart Societies**



**Master's Degree Thesis**

**Detecting Interference and Classification  
in Global Navigation Satellite Systems  
Based on Machine Learning Techniques**

**Supervisor**

**Prof. Fabio DOVIS**

**Candidate**

**Iman EBRAHIMI MEHR**

**April 2021**

# Summary

The Global Navigation Satellite Systems (GNSSs) have been established as one of the most critical infrastructures in today's world. GNSS positioning, navigation, and timing services have played a significant role in various applications. These GNSS signals are transmitted from satellites that are over 22000 km away from the earth. The power of the received signals at the GNSS receivers' antenna is extremely weak, and the signals are vulnerable to interference, which can cause degraded positioning accuracy or even a complete lack of position availability. Thus, it is essential for GNSS applications to detect interference and further recognize the interference types for the interference mitigation in GNSS receivers to guarantee reliable solutions. Various types of intentional interference can disturb the GNSS signals in the receivers. In this thesis, the focus is on detecting chirp signals, known as one of the most common and disruptive interfering signals.

Digital signal processing techniques are traditional ways in order to detect the presence of interference. The common drawback of such techniques is that they rely on complex and computationally expensive operations. Recent researches have shown that machine learning methods can be utilized to detect interferences. Machine learning is the methodical study of intelligent algorithms and systems that improve their knowledge or performance by experience. For this reason, the use of machine learning within the framework of Radio Frequency Interference (RFI) detection and mitigation can be multifold. The proposed algorithms are unsupervised machine learning (K-Means clustering) and Convolutional Neural Network (CNN).

Features of signals extracted for k-means solution are in the time domain, frequency domain, and time-frequency domain. Different feature selection approaches are implemented, including principal component analyses, correlation matrix evaluation, and supervised techniques.

Another investigated classifier is CNN based on multi-layer neural networks. The images of signals in the time-frequency domain using Wigner-Ville transform

are fed to the CNN algorithms to classify the different shapes of chirp signals.

# Acknowledgements

Prima di tutto vorrei ringraziare **l'Italia**, un Paese che mi ha dato l'opportunità di crescere e di raggiungere i miei obiettivi.

Ringrazio **Edisu** e il **Politecnico** per i servizi di cui ho usufruito nell'arco della mia carriera magistrale.

Ringrazio **Prof. Magli** per la motivazione datami.

Ringrazio **Prof. Dosis** per le conoscenze che mi ha trasmesso nel campo GNSS.

Vorrei ringraziare l'ingegnere **Qin Wenjian**, per la sua gentile supervisione.

Uno speciale ringraziamento va alla **mia famiglia** che, anche se lontana, è sempre stata vicina a me.

Ringrazio inoltre **Stella, Sajad, Sara** e **ARIO**, le persone a me più care qui in Italia, che sono la mia seconda famiglia.

Un ulteriore ringraziamento va ai miei compagni di università: **Giulia, Gennaro, Duilio, Antonio, Mersedeh, Mostafa** e **Dr. Rasoul**.

Infine ringrazio **me stesso** per gli sforzi e l'impegno nel mio lavoro.



*Iman Ebrahimi Mehr (Arman)*

Torino 2021

# Table of Contents

<b>List of Figures</b>	VI
<b>Acronyms</b>	VIII
<b>1 Overview of GNSS Interference</b>	1
1.1 Global Navigation Satellite systems . . . . .	1
1.1.1 GNSS Signal and its Band . . . . .	2
1.2 Interference Concept . . . . .	4
1.2.1 Interference Characteristics . . . . .	5
1.3 Interference Sources . . . . .	5
1.3.1 Natural sources of Interference . . . . .	6
1.3.2 Artificial sources of Interference: Unintentional . . . . .	6
1.3.3 Artificial sources of Interference: Intentional . . . . .	7
1.4 Impact of Interference on GNSS Receivers . . . . .	9
1.5 State of the art . . . . .	10
1.5.1 Traditional Detection Technique . . . . .	10
1.5.2 Machine Learning Based Detection Technique . . . . .	12
<b>2 Machine learning Review</b>	14
2.1 What is Machine Learning? . . . . .	14
2.1.1 Supervised Machine Learning . . . . .	16
2.1.2 Unsupervised Machine Learning . . . . .	16
2.2 Feature Selection Techniques . . . . .	17
2.2.1 Unsupervised Feature Selection . . . . .	18
2.2.2 Supervised Feature Selection . . . . .	20
2.3 Neural Network . . . . .	22
2.3.1 Learning Process of Neural Networks . . . . .	22
2.3.2 Minimization Problem . . . . .	24
2.3.3 Backpropagation algorithm . . . . .	25
2.3.4 Cost Function . . . . .	26
2.3.5 Regularization and Dropout . . . . .	27

2.3.6	Convolutional Neural Networks . . . . .	28
<b>3</b>	<b>Simulation</b>	<b>30</b>
3.1	GNSS Signal . . . . .	30
3.2	Chirp Signal . . . . .	31
3.2.1	Chirp Classifications . . . . .	32
3.3	Combining with CHIRP . . . . .	34
<b>4</b>	<b>Feature Extraction</b>	<b>36</b>
4.1	Features in Time Domain . . . . .	36
4.2	Features in Frequency Domain . . . . .	37
4.3	Features by Wavelet Transform . . . . .	40
4.4	Features by Wigner-Ville Transform . . . . .	45
4.5	Final Dataset . . . . .	47
<b>5</b>	<b>Methodology and Result</b>	<b>48</b>
5.1	K-means Clustering . . . . .	49
5.1.1	Data Preparation . . . . .	50
5.1.2	Feature Selection . . . . .	51
5.1.3	Modelling . . . . .	53
5.1.4	Evaluation and Result . . . . .	54
5.2	Convolutional Neural Network . . . . .	58
5.2.1	Images Dataset . . . . .	59
5.2.2	Model Structure . . . . .	60
5.2.3	Configuration and Result . . . . .	63
<b>6</b>	<b>Conclusion</b>	<b>66</b>
<b>A</b>	<b>Chirp Shapes</b>	<b>68</b>
A.1	Wide sweep . . . . .	68
A.2	Narrow sweep . . . . .	70
A.3	Triangular . . . . .	71
A.4	Triangular wave . . . . .	72
A.5	Sawtooth . . . . .	73
A.6	Hooked sawtooth . . . . .	74
A.7	Tick . . . . .	75
A.8	Multi tone . . . . .	76
	<b>Bibliography</b>	<b>77</b>

# List of Figures

1.1	Trilateration between satellites and receiver [2]	1
1.2	L Frequency Band [4]	3
1.3	GPS L1 signal [4]	3
1.4	Wave Interference [5]	4
1.5	Radio Frequency Band [11]	7
1.6	Artificial Sources	9
1.7	General architecture of a GNSS receiver [7]	11
2.1	Machine Learning Procedure	15
2.2	Machine Learning Classification	15
2.3	Sigmoid, Hyperbolic Tangent and Relu activation functions	24
2.4	Local Receptive Fields [36]	28
2.5	Sample of Max Pooling [36]	29
2.6	Complete Layers of CNN (Number of layers can be rather large)[36]	29
3.1	Simple Chirp Signal	31
3.2	Chirp Signal Classifications	32
3.3	Narrow-band Chirp Signal [68]	33
3.4	Narrow-band Chirp	33
3.5	Wide-band Chirp	33
3.6	Pseudocode of Combining GNSS signal with Chirp Signals	35
4.1	Time Domain vs Frequency Domain [71]	38
4.2	Magnitude and Phase of Complex Number	40
4.3	Comparison of Different Transformations [75]	42
4.4	Multi-Level Decomposition [76]	43
4.5	Sym4 Wavelet	43
4.6	Wavelet analyses over original and combined signal with different types of chirp	44
4.7	Wigner Ville analyses over Original and combined signal with different types of chirp	46

5.1	Methodology Diagram . . . . .	48
5.2	Correlation between all features (the image is imported as pdf file, you can zoom in to see more details) . . . . .	51
5.3	Cumulative explained variance of different number of components . . . . .	53
5.4	Taxonomy of Evaluation Methods Covering Internal and External Validation Approaches . . . . .	55
5.5	Sample of images in dataset . . . . .	60
5.6	AlexNet Architecture . . . . .	62
5.7	Accuracy and Lost in the Training Process of AlexNet . . . . .	63
5.8	Confusion Matrix of CNN Algorithm . . . . .	64
A.1	Wide sweep - Slow(2 to 3 chirps per 100 microseconds) [68] . . . . .	68
A.2	Wide sweep - Medium(4 to 6 chirps per 100 microseconds) [68] . . . . .	69
A.3	Wide sweep - Fast(8 to 12 chirps per 100 microseconds) [68] . . . . .	69
A.4	Wide sweep - Rapid(more than 12 chirps per 100 microseconds) [68] . . . . .	69
A.5	Narrow sweep [68] . . . . .	70
A.6	Triangular [68] . . . . .	71
A.7	Triangular Wave [68] . . . . .	72
A.8	Sawtooth [68] . . . . .	73
A.9	Hooked Sawtooth [68] . . . . .	74
A.10	Tick [68] . . . . .	75
A.11	Multi Tone [68] . . . . .	76

# Acronyms

**GNSS** Global Navigation Satellite system

**RFI** Radio Frequency Interference

**EMI** Electromagnetic Interference

**MEO** Medium earth orbit

**GPS** Global Positioning System

**GDOP** Geometrical Dilution of Precision

**SNR** Signal to Noise Ratio

**NBI** Narrow-band Interference

**WBI** Wide-band Interference

**TEC** Total Electron Content

**SIS** Signal in Space

**RNSS** Radio Navigation Satellite System

**PRN** Pseudo-Random Noise

**AGC** automatic gain control

**ADC** Analog to Digital Converter

**PVT** position, velocity, and precise time

**CNN** Convolutional Neural Network

**ML** Machine Learning

**DSP** Digital Signal Processing

**MAD** Mean Absolute Deviation

**AF** Autocorrelation Function

**FFT** Fast Fourier Transform

**DFT** Discrete Fourier transform

**STFT** Short Time Fourier transform

**DWT** Discrete Wavelet Transform

**WVT** Wigner-Ville transform

**PCA** Principal Component Analyses

**ANOVA** Analysis of Variance

**RELU** REctified Linear Units

**SGD** Stochastic Gradient Descent

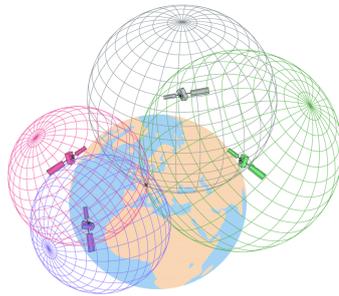
# Chapter 1

## Overview of GNSS Interference

### 1.1 Global Navigation Satellite System

Global Navigation Satellite System (GNSS) refers to the constellation of satellites placed in Medium Earth Orbit(MEO), around 20,000 km far from the earth and provides Positioning, Navigation, and Timing (PNT) services on a global basis. America's Global Position System(GPS), Europe's Galileo, Russia's Global Navigation Satellite System(GLONASS) and China's BeiDou provide these services.

Each GNSS receiver, to determine its location, has to estimate the distance between itself and the satellites, create the line of position by measuring the signal's propagation time (time difference between signal transmission time by satellite and the received time in the receiver). Then by solving trilateration between at least four satellites, it can obtain X, Y, Z, and  $\delta_t$  (misalignment time between the receiver and GNSS time by considering all the satellites are synced) [1].



**Figure 1.1:** Trilateration between satellites and receiver [2]

The Coordinated Universal Time (UTC) is the global time reference that contains hundreds of atomic clocks located at different calibrations and aligned with the time reference from the constellation of satellites[3]. GNSS consists of the three main segments:

- Space segment: refers to satellites constellation on the orbit.
- Control segment: includes a network of monitoring stations distributed worldwide and checks the status of satellites and signals. The master stations keep the control of time scale, and other stations communicate to the satellites to upload data or adjust the position of satellites.
- User segment: represents different types of receivers that are all common in a certain task, positioning based on velocity and time.

As shown in Figure 1.1, the availability of the position depends on the signals from at least four satellites, as well as the distribution of the satellites in the sky, so the geometry of satellites is an impact of quality which is known as Geometrical Dilution of Precision (GDOP).

GNSS receivers are designed to receive a set of signals from a satellite over a specific range of frequencies. The frequency ranges that the receiver needs to track are typically defined by a series of bandpass filtering inside the receiver, called pass-band.

### 1.1.1 GNSS Signal and its Band

GNSS signal is transmitted continuously from the satellites in two or more frequencies in the L band and its range is between 1 to 2 GHz. This L band is shared between all GNSS constellations such as GPS, Galileo, GLONASS, and others.

As it represents in Figure 1.2, Radio Navigation Satellite System (RNSS) almost occupy all band of L band region. There is also another communication system, Aeronautical Radio Navigation Service (ARNS). Note that Figure 1.2 only shows three constellations, since there exists more like BeiDou (Chinese), IRNSS (Indian), QZSS(Japanese).

GNSS signal comprises three main components that allow the user to compute its position. Formula 1.1 states the GPS L1 Signal, and Figure 1.3 shows the components of the signal.

$$X_{RF}(t) = \sqrt{2P_c}c(t)d(t)\cos(2\pi f_L t + \theta_{L1}) \quad (1.1)$$

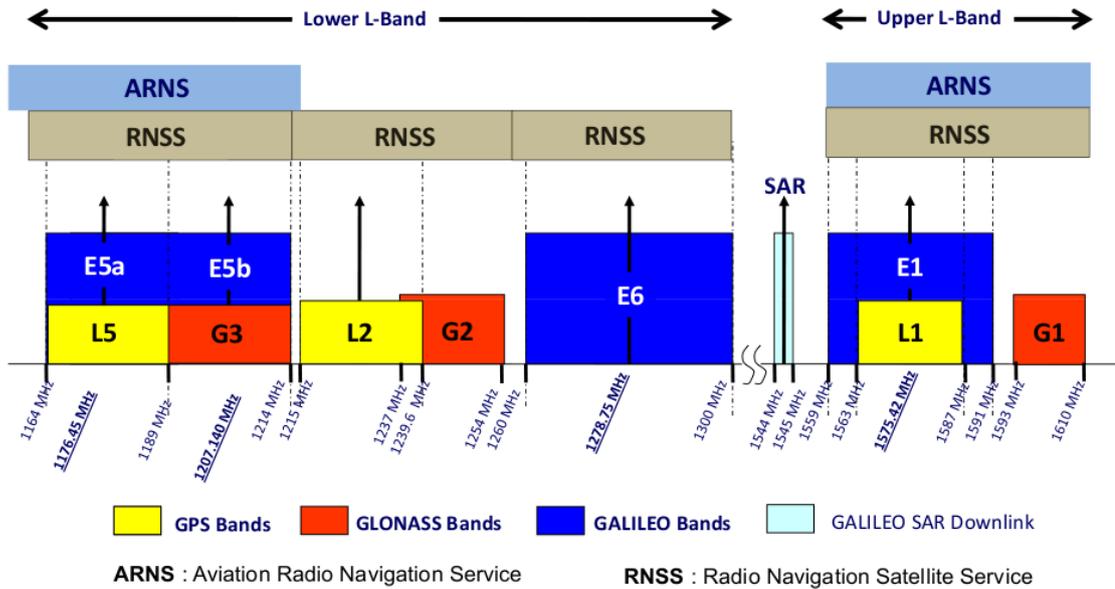


Figure 1.2: L Frequency Band [4]

- **Carrier:**  $f_L$  Radio-frequency sinusoidal signal at a given frequency.
- **Ranging code:**  $c(t)$  is a binary sequence (zeroes and ones) and unique for each satellite. It is also known as Pseudo-Random Noise sequences or PRN codes which allow the receiver to estimate the pseudo-range.
- **Navigation data:**  $d(t)$  is a binary-coded message providing complementary information (satellite ephemeris, almanac, satellite health status) about the satellite that the receiver needs to know.

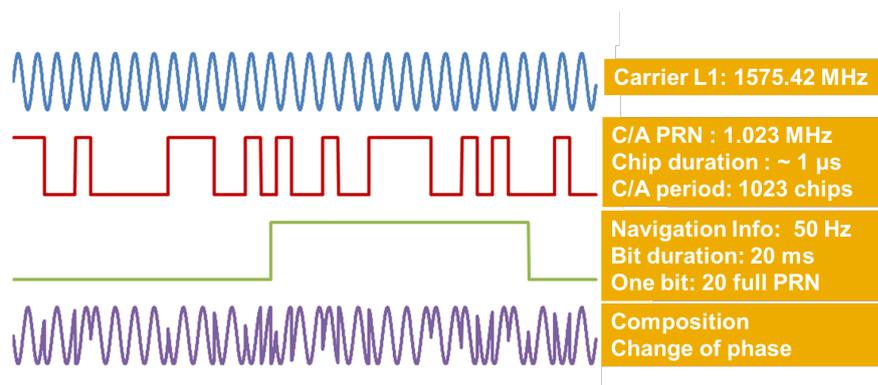
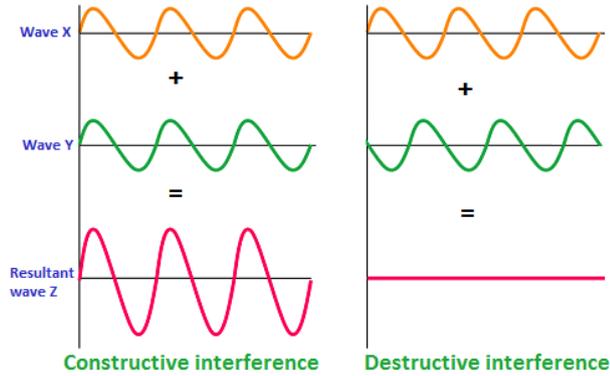


Figure 1.3: GPS L1 signal [4]

Each GNSS signal occupies a range of frequency in this band called GNSS bandwidth ( $B_{GNSS}$ ). In GPS L1 case, the central frequency (carrier frequency) is 1575.42 MHz, and the bandwidth is 24 MHz which means the maximum and minimum frequencies of these signals are 1587 MHz and 1563 MHz, respectively (for more details in the bandwidth of each signal, refer to [4]).

## 1.2 Interference Concept

According to the Cambridge dictionary, in Physics, interference between two waves occurs when they have same frequency and produce a force that is either stronger or weaker than one wave alone. The stronger wave is called constructive, and the weaker wave is destructive interference. Figure 1.4 represents a straightforward example of wave interference.



**Figure 1.4:** Wave Interference [5]

In the field of telecommunications, data can be transmitted between a source and destination over a radio wave with a specific frequency in the range of 3 kHz up to 300 GHz. Any electromagnetic source out of transmitter interacting or disturbing with the radio wave is known as Radio Frequency Interference (RFI) or Electromagnetic Interference (EMI). The disturbance might decrease the communication performance and increase the error rate or even completely stop it from functioning and cause loss of data.

According to the International Telecommunication Union [6], RFI is defined as:

“The effect of unwanted energy due to one or a combination of emissions, radiations, or inductions upon reception in a radiocommunication system, manifested by any performance degradation, misinterpretation, or loss of

information which could be extracted in the absence of such unwanted energy.“

As it has mentioned in Section 1.1, constellations of satellites are placed quite far the earth, and the power of the received signals in the receiver’s antenna is extremely weak. The weakness of power looks like looking for the light emitted from a 100W lamp at a distance of 20 000 km. It means that the signal could be easily disturbed. Therefore, even small disturbance could be a potential source of interference for GNSS signal. Due to the GNSS signal’s weakness and crowded frequency spectrum, the GNSS signal is vulnerable to other communication systems [7].

### 1.2.1 Interference Characteristics

Each interference can be categorised based on its characteristics in the frequency domain as follows [7]:

- **Narrow-band Interference (NBI):** It refers to interference whose bandwidth is much narrower compared to bandwidth of interest and disturbs some portion of GNSS bandwidth. ( $B_{inf} \ll B_{GNSS}$ )
- **Wide-band Interference (WBI):** The interference’s bandwidth is almost the same as GNSS bandwidth. ( $B_{inf} \approx B_{GNSS}$ )
- **Continuous-Wave Interference (CWI):** This type has constant amplitude and frequency and appears as a single tone in the frequency domain. ( $B_{inf} \approx 0$ ) Continuous-wave interference (CWI) can induce significant ranging error for a spread spectrum receiver [8].
- **Swept-frequency interference (chirp signal):** It is a signal that its frequency changes with time and variation of frequency can be linear or non-linear. Most chirp signals have appeared as wideband interference in the frequency domain, and they are broadcast by jammers to disrupt the GNSS signal [9]. Section 3.2 will discuss more in details about this type.

## 1.3 Interference Sources

Any undesired radio signals are classified as RFI and reduce the efficiency of a wireless receiver or even interrupt the receiving. RFI signals have a complex nature and can be generated unintentionally, mostly harmonics from lower frequency bands, inter-modulation products, and out-of-band emissions or intentionally, often named jamming. The RFI signals are bothersome for any receiver, but they mainly warn

for applications based on low power signals. This points to the applications that rely on the Global Navigation Satellite Systems (GNSS), such as GNSS-Reflectometry (GNSS-R)[3].

### 1.3.1 Natural Sources of Interference

GNSS signal travels from a satellite in space to reach the destination on earth. There are some natural sources which can affect the signal. During propagation of the signal in the atmosphere, Atmospheric impairment has to be considered. The atmosphere consists of two layers, the upper layer containing a region of ionized gases called the ionosphere and the lower troposphere containing clouds. The free electron density changes signal propagation speed and introduces a delay in propagation because there is an interaction between charged particles and the electromagnetic wave of the signal. The troposphere is the humid part of the atmosphere and introduces attenuation power in a signal, this phenomenon is less critical than the ionosphere effect [7].

The ionization distribution is not uniform in the ionosphere layer around the earth, both in space and time. This incident is caused by space weather which involves a complex interaction between solar radiation, solar wind, and the earth's magnetic field. So when the signal crosses this non-uniform area, it is similar to what is called fading in wireless communications and induces a frequency shift in the signal carrier. This event in the GNSS domain is called ionospheric scintillation, that is tough to predict and known as natural interference [7] [10].

### 1.3.2 Artificial Sources of Interference: Unintentional

In addition to natural sources of interference which are introduced as unintentional interference, other human-made types of interference have existed that can affect and degrade the accuracy of position. Some of unintentional artificial sources are sorted below:

- **Radio Spectrum:** It is part of the electromagnetic spectrum in the range of 30 Hz (super low frequency) up to 300 GHz (extremely high frequency). Nowadays, with the increasing number of wireless communication infrastructures, the radio spectrum becomes more and more crowded, as it is shown in Figure 1.5. Thus, it might overlap or interfere between signals used to communicate with satellites and those used for terrestrial networks such as mobile phone systems. The main effects are caused by high-order harmonic or spurious components

generated by intermodulation products in the communication transmitter. (For more details and examples, refer to book [7])

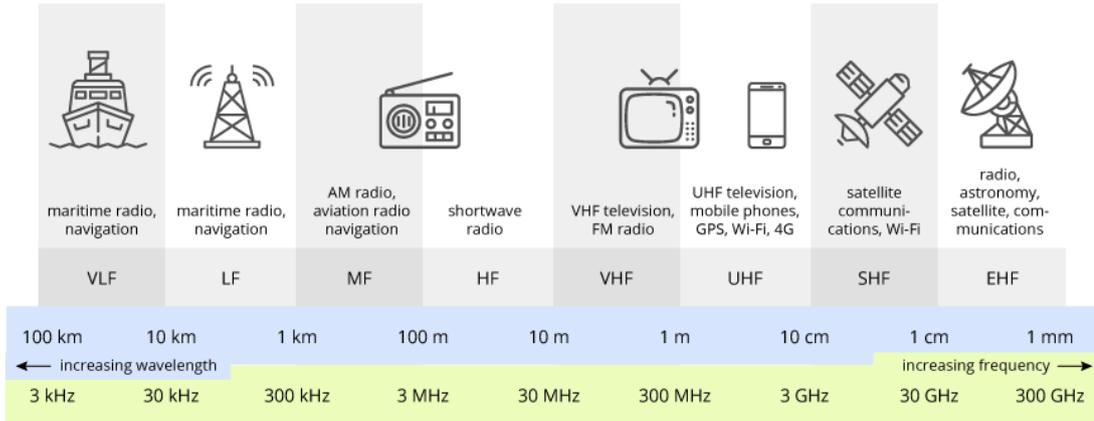


Figure 1.5: Radio Frequency Band [11]

- **Intersystem and Intrasystem:** GNSS constellations grow fastly with more signals at higher power levels in the same frequency bands. Due to the growing number of satellites in view for GNSS receivers, this implies increasing inter and intrasystem interference which should be considered in the design phase of signal [12].
  - **Intrasystem:** Each satellite signal is designed to be orthogonal with respect to other satellites in the same constellation. Thus, they can be distinguished for receivers to recognize from which satellite comes. However, this orthogonality might not be perfect, and some residual power generates interference known as intrasystem interference [7].
  - **Intersystem:** It is due to using the same bandwidth for satellite communication between the different constellations. For instance, the L1 signal from GPS and E1 from Galileo use the same carrier frequency, while they use different modulation to be distinguishable from each other. Therefore some power of the signal of another system can disrupt the signal of interest [7].

### 1.3.3 Artificial sources of Interference: Intentional

As mentioned, GNSS signals are really at risk of being interfered; therefore, they might be under attack by a person or an organization to fool the system or disrupt positioning [13]. This section describes the different types of intentional interference.

According to the book *GNSS Interference Threats and Countermeasures* [7], from a general perspective, an intentional attack on GNSS receivers might state in two different layers:

- **Non-signal attack:** is directly on the receiver. Typically this type includes tampering with the receiver's information or the alteration of the position reported by the receiver to a service provider or a control centre. Non-signal (cyber) attacks need no RF equipment at all.
- **Signal attack:** is at Signal in Space(SIS). This category includes a deliberate attack on GNSS signal and generally is categorized into three forms: Meaconing, Spoofing, and Jamming.

Intentional interference in GNSS is becoming a huge threat for current systems and applications based on positioning. Intentional interference can be categorized into three groups as follows:

- **Meaconing:** This attack refers to devices that receive a signal from the satellite and rebroadcast the signal with a delay, and therefore, the measure of propagation time is not correct; consequently, the position is wrong. The existing GNSS system offers two services; the first one is authenticated signal (P(y) code in case of GPS) in which the signal is encrypted and only used for military purposes, while the second service with no need for authentication refers to civil users (C/A code in case of GPS). In both cases, meaconing without knowing the sequence of encrypting can affect the solution of positioning. Meaconing can be detected by analyzing the GNSS receiver's clock bias [14].
- **Spoofing:** Spoofing is a transmission of a counterfeit GNSS signal similar to the original one, forcing the receiver to provide an erroneous position without disrupting the receivers' operation. This type of attack can be implemented in civil user signals because they are easy to regenerate and not encrypted. A spoofing attack happens when there are several fake signals in the GNSS receiver, and the power of these signals is more potent than the power of the signal from the satellite. Thus, the receivers, by using them, will face a wrong position [7] [1].
- **Jamming:** Jamming refers to transmit energy at the same frequency of GNSS signal to mask it by noise. Thus, the signal-to-noise ratio decreases and causes failure in tracking the signal of interest. The jammer device can mask or jam a specific portion of the GNSS signal band, disrupting receivers' operation. Jamming is more common than the other two types of attack, and a jammer device can be bought easily from the internet, but the usage of these types of devices is illegal. One of the most common of these devices is an in-car

jammer powered by a car's cigarette lighter, and it allows the driver to avoid being tracked [15] [16] [7] .

All of the artificial sources are summarized in Figure 1.6

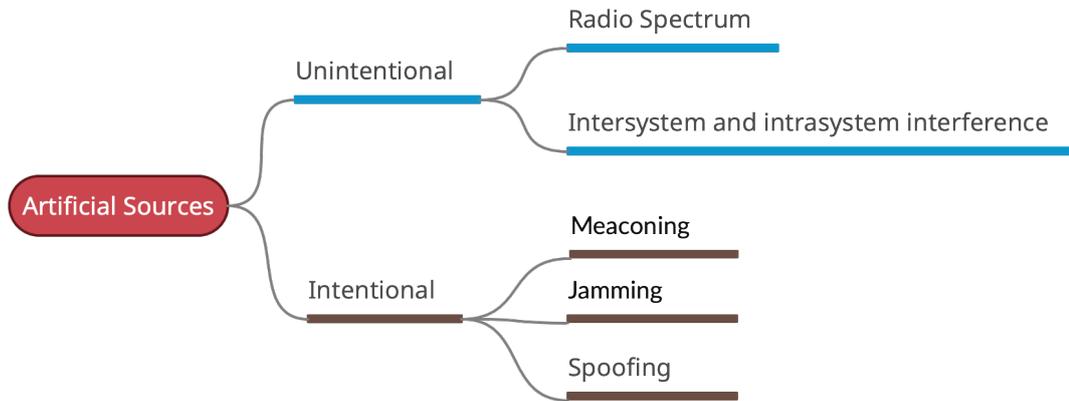


Figure 1.6: Artificial Sources

## 1.4 Impact of Interference on GNSS Receivers

Intentional interference by jammers makes GNSS-based systems unavailable due to the fact that strong interference may completely make the GNSS receivers inoperable. Although in some cases, the interference is not strong enough to cause the receiver to face a problem in positioning, it can increase the error of the pseudorange and/or phase measurements and, as a consequence, the accuracy of the positioning solution reduces. The presence of jamming signal affects both acquisition and tracking stages in the receiver which is leading to loss of quality of the GNSS satellite signals. Consequently, GNSS receivers cannot provide accurate positioning according to velocity and time services [17].

An analysis based on changing the observed carrier-to-noise density (C/N0) values of the received signals represented that regardless of the use of an L1/E1 jammer, the GLONASS and Galileo signals will be affected due to the increment of C/N0 or even the loss of signal. In fact, a jammer can considerably impact the quality of positioning by both incorrect position or lack of final position [18].

Furthermore, in order to acquire high accuracy in positioning, high reliability, and prevent lack of positioning, it is critical to **DETECT and MITIGATE**

interference in GNSS signal. The GNSS applications need to detect interference and further recognize the types of interference to mitigate and guarantee reliable solutions. Nevertheless, RFI detection in GNSS is like finding a needle in the haystack [7] which means it is a challenging task due to the large search space with respect to time and space.

## 1.5 State of the art

During the past decades, the number of applications based on GNSS has highly increased [19], and GNSS has established itself as part of critical infrastructure in today's world. In fact, it is widely used to provide positions in Unmanned Aerial Vehicles (UAV) and Unmanned Ground Vehicles (UGV), and it combines with the other information provided by other sensors[20]. Hence, RFI can be appointed as one of the most significant topics on GNSS-based devices not only for navigation, positioning, and timing but also for Earth Observation purposes such as GNSS-Reflectometry (GNSS-R)[19].

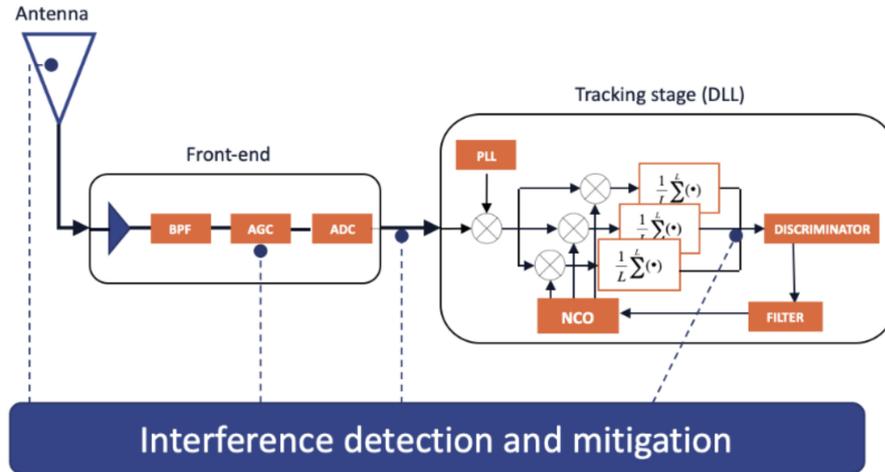
The existence of GNSS interference makes the communication environment complex. To reduce the interference effect, many researchers have begun to detect and classify the interference signals. Consequently, effective interference detection and classification are critical aspects to enhance anti-jamming techniques, and efficient interference identification enables us to control the electromagnetic spectrum [21]. In literature, it is possible to find many methods of interference detection. Some of them refer to past years [22] [23]. The topic of interference monitoring has constantly been relevant in recent years.

### 1.5.1 Traditional Detection Technique

Depending on characteristics and power, RFI can affect the procedure of positioning in receivers at different stages. According to the book [7], detection and countermeasures can be done at different stages as shown in Figure 1.7.

Various traditional techniques are used to detect RFI at different stages of receivers and some of them sorted as follows:

- **AGC monitoring at Front-end Stage:** Automatic Gain Control (AGC) is a critical component in GNSS receivers, where the signal power is below that of the thermal noise floor, the AGC is driven by the noise environment rather than the signal power. AGC denotes the main equipment for assessment of the operating environment in the GNSS receiver. The AGC advantage is the ability of interference detection and estimation. Moreover, the AGC may



**Figure 1.7:** General architecture of a GNSS receiver [7]

increase the dynamic range, and it can be introduced as an adaptive variable gain amplifier with the main role to minimize quantization losses. The AGC operation is acted as an analog-to-digital converter (ADC) [7] [24] [25].

- **Time-domain statistical analysis at Front-end Stage:** The techniques based on the statistical characteristics are sensitive to the presence of interference and can be used for recognition purposes. Hence, Time-varying interference can be detected by performing statistical analysis methods based on observing the time fluctuations of the signal sample distribution at the ADC output [26]. Normally, time domain techniques can only converse with the narrow interference due to the fact that the wideband interference cannot be recognized simply from Gaussian white noise [27].
- **Spectral Monitoring:** An interfering signal that influences the input signal during the antenna's reception process is expected to be distinguished via spectral analysis by comparing the estimated power spectral density of the received signal with a spectral mask that appropriately represents the nominal interference-free conditions. A spectral assessment of the received signal is expected to interpret the front-end's equivalent transfer function, multiplied by the noise variance which passed through the analog front-end. The interference-free expected spectral evaluation can be considered a known value after appropriate calibration of the antenna and RF front-end system [7].
- **C/N0 Monitoring at Post-Correlation:** is popular due to its low complexity implementation, but, as previously remarked, it can be due to several factors other than interference [28].

- Techniques based on **Multi-Correlators** aim at identifying distortions of the correlation function of the signal assessing several metrics and comparing them to a threshold. While they might be useful for detecting the outliers, they might not clarify the actual RFI presence or its features [29].

There are some other techniques such as post-correlation statistical analysis [7] [30], pseudo-range monitoring [7] and PVT solution.

### 1.5.2 Machine Learning Based Detection Technique

The limitations of the traditional approaches can be reduced by using machine learning methods, which can be learned based on specific processes and provide automatic detection and classification with high validity. The objective of machine learning is to recognize the best algorithm and extract the most appropriate set of features to build a model that could achieve the desired goal in detection with high precision [31].

Machine learning is the consistent study of intelligent algorithms and systems that assist in developing system performance. According to IBM Co, Machine learning focuses on applications that learn from experience and improve their decision-making or predictive accuracy over time. In general concept, the machine learning process refers to the ability to solve a difficult task by processing the right features according to a model. The use of machine learning in Location-Based Services is also motivated by the increasing volume of available data collected at remote sites through low-cost GNSS software receivers [16].

Nowadays, by increasing computing ability, the hard problem can be solved using Machine Learning techniques [32]. Therefore an automated approach is useful and functional to detect RFI. Many work of literature have addressed using machine learning to detect RFI in general domain of signal, [33] [34]. Given the increased prevalence of interferences in GNSS bands, the GNSS literature's focus in recent years has shifted towards integrity methods and procedures.

For GPS interference signals, Liu et al. implemented the fast independent component analysis method to extract the interference characteristic factors in the frequency domain, time domain, and time-frequency domain. Tao et al. proposed the Welch method and the fractional Fourier transform to find the interference parameters. Liu et al. performed a new algorithm based on a support vector machine (SVM) by extracting the signal's characteristics and analyzing them [21].

In another paper, machine-learning techniques are implemented to detect ionospheric scintillation events affecting the GNSS signal amplitude. Based on this method, the ML algorithms are learned historical pre-classified data (decision tree) to perform automatic classification on new data. The result indicates that this approach performs better with respect to the traditional methods in terms of accuracy. Moreover, the achievements show that signal-based features have better performance in contrast to the observable-based features. Machine learning can simplify analyzing big GNSS data sets affected by the scintillation phenomenon [31].

With reference to one research, the k-means algorithm was performed to identify the GNSS signals suffering from multipath, therefore by eliminating the affected signals from the PVT solution, the accuracy of positioning was improved [20].

A methodology to classify jammer types was proposed by using two different ML algorithms: SVM and CNN on the set of generated images of time-frequency analysis. According to the conclusion in the classification of the interference and interference-free scenario, it is stated that with a small set of images and not excessively complex parameters, a high accuracy result is obtained [35].

Due to the acceptable result of different investigations aforementioned and DSP limitations, machine learning will aid to achieve better performances on analyzing and classification of GNSS interferences.

As explained, various types of intentional interference can disturb the GNSS signals in the receivers (Section 1.3.3). This thesis focuses on detecting chirp signals, known as one of the most common and disruptive interfering signals. The proposed algorithms are unsupervised machine learning (k-means clustering) and Convolutional Neural Network (CNN).

Features of signals extracted for the K-means solution are in the time domain, frequency domain, and time-frequency domain. Different feature selection approaches are implemented, including principal component analyses, correlation matrix evaluation, and supervised techniques.

Another investigated classifier is CNN which is based on multi-layer neural networks. The images of signals in the time-frequency domain using Wigner-Ville transform are fed to the CNN algorithms to classify the different shapes of chirp signals.

# Chapter 2

## Machine learning Review

### 2.1 What is Machine Learning?

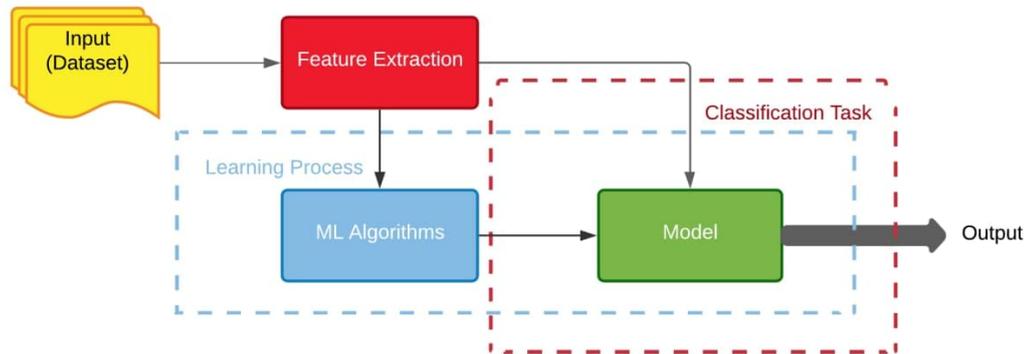
Machine Learning(ML) is a set of techniques to detect data patterns and describe characteristics of the data, and therefore, it is possible to predict new data's behavior. Data analysis methods are employed to process data and extract useful information from them [32].

- ML based on statistics: uses training data to determine the parameters of a statistical model fitting, it uses the model to make prediction on new data [36].
- ML based on numerical models: uses training data to determine the parameters of a generic numerical model of a given type (e.g. deep neural network, using a very large number of parameters) [36].

ML is the systematic study of intelligent algorithms and systems that improve their knowledge or performance by experience. In its general concept, the machine learning process refers to the ability to solve a task, processing the right features describing the domain of interest, according to a model[31]. Figure 2.1 shows the procedure of a machine learning problem that needs to be solved and the ML algorithm's main elements are sorted as follows:

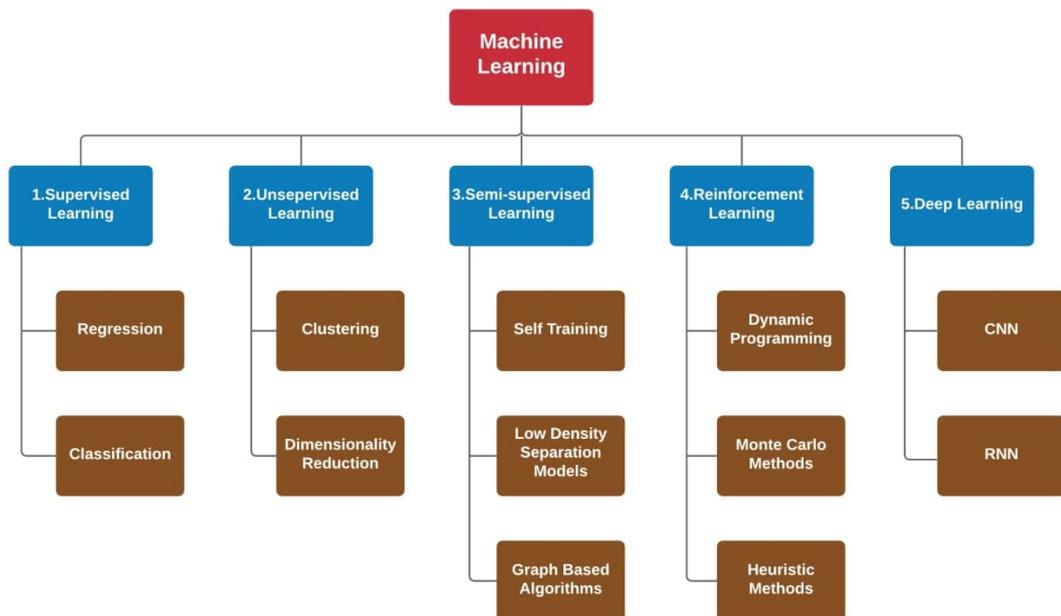
- **Domain:** the problem to be solved [31].
- **Features:** the description of the objects of the domain [31].
- **Task:** the abstract representation of the problem that reflects in the mapping between the input and the output (the automatic classification of data collection samples) [31].

- **Model:** the output of the ML when the training set is fed to the algorithms [31].



**Figure 2.1:** Machine Learning Procedure

The first step towards using ML is to clearly understand the problem, the number of resources, and required efforts in different kinds of ML algorithms that are sub-categorized to different types. Figure 2.2 represents different types of ML.



**Figure 2.2:** Machine Learning Classification

### 2.1.1 Supervised Machine Learning

In any supervised ML problem, there is a training set which is a set of data used to train the algorithm, denoted as  $D$  (stands for Data), and it consists of the two following elements:

- $x_i$  represents input of machine which is a column vector, each row is a  $D$ -dimensional vector which describes the data and each element is called feature.
- $y_i$  the scalar coefficient, the output/answer associated with the  $x_i$  vector.

$$D = \{(x_i, y_i)\}_{i=1}^N \quad (2.1)$$

Formula 2.1 represents data, where  $N$  is the number of training examples. Each training output  $y_i$  can:

- belong to a finite set:  $y_i \in \{1, \dots, C\}$  **CLASSIFICATION PROBLEM**, where our data can belong to 1 out of  $C$  different classes [37].
- be a real number, in this case we call this **REGRESSION PROBLEM** [37].

### 2.1.2 Unsupervised Machine Learning

Unsupervised data learning inducts pattern identification without the requirement of a target attribute. Hence, all the elements used in the analysis are denoted as inputs. The unsupervised learning algorithms are more appropriate for creating the labels in the data, leading to supervised learning tasks. Therefore, unsupervised clustering algorithms associate groupings within the unlabeled data. On the other hand, they identify rules or patterns that accurately demonstrate the relationships among attributes [38]. **Cluster analysis** is one of the main approaches used in unsupervised learning. Cluster Analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense) to each other than to those in other groups (clusters) [39].

#### K-Means Clustering

The **K-means** is an algorithm to implement clustering over data points. It is an iterative approach that tries to partition data into  $K$  pre-defined distinct clusters where each point can only belong to one cluster. The K-means algorithm defines the number of clusters and initializes centroids of each cluster randomly from the data point. It then calculates the distance between each data point and the centroids. Then, data points are assigned to the cluster with the minimum distance. Before

starting the next iterate, the centroids are calculated for each cluster which is simply the arithmetic mean of all the data points that belong to that cluster. Suppose it is different from the previous cluster's centroid, in this case, the centroid moves to the new one, and this procedure continues until there is no more improvement in clustering centroid [40]. The following steps state one iteration of K-means process:

1. Specify K as the number of clusters.
2. Selecting randomly K data points for the centroids.
3. Computing the sum of the squared distance between points and all centroids.
4. Assigning each data point to the closest cluster(centroid).
5. Computing the new centroids for the clusters by taking the average of the all data points that belong to each cluster.

Contrary to supervised learning, where exists the ground truth to evaluate the model's performance, in clustering **Silhouette analysis** can be used, which determines the degree of separation between clusters. The silhouette value is a measure of how similar an object is to its cluster ( $a_i$ ) compared to other clusters ( $b_i$ ). The silhouette ranges from -1 to +1, where a high value indicates that the object is well matched to its cluster and poorly matched to neighboring clusters [41].

$$s(i) = \frac{b_i - a_i}{\max(a_i, b_i)} \quad (2.2)$$

## 2.2 Feature Selection Techniques

Feature selection reduces the number of input variables by removing non-informative or redundant features to reduce the computational cost of modeling and improving the model's performance. Using many features in some models can slow the development and training process and degrade the performance [42]. In implementing ML, it is crucial to select features with the most informative data of the problem .

Generally, feature selection techniques can be divided into two categories, supervised and unsupervised [43]. The difference is related to the selection of features based on the target variable (label of data) or not, it means supervised methods choose the high correlated variables to the target output. One of the other important advantages of feature selection is reducing the probability of **Overfitting** (the model has trained very well but just for the training dataset, and it is not

generalized for other data [44]. Overfitted models learned from all the data, containing the unpreventable noisy data on the training set, instead of studying the hidden rule beyond the data [45]). Different types and methods of feature selection can be listed as follows [43]:

- **Unsupervised:** uses the target variable (removes redundant variables).
  - **Correlation Matrix**
  - **Principal Component Analyses (PCA)**
- **Supervised:** Uses the target variable (removes irrelevant variables).
  - **Wrapper:** searches for well-performing subsets of features (RFE).
  - **Filter:** selects a subset of features based on their relationship with the target.
  - **Intrinsic:** performs an automatic method during training process for feature selection.

### 2.2.1 Unsupervised Feature Selection

Unsupervised learning is a set of statistical tools for scenarios with a set of not labeled features (no target variable).

#### Feature Selection based on Correlation Matrix

One of the most common and valuable techniques of feature selection is based on the Correlation Matrix. The correlation term of two variables refers to the measure level(amount) of their dependency. Features with high correlation are more linearly dependent and have almost the same effect on the data description. Consequently, when two features have a high correlation, one of them can be dropped. The correlation matrix represents the correlation between each pair of features, and the diagonal elements of this matrix are always one [46].

There are various metrics to calculate correlation, such as Pearson R correlation, Kendall rank correlation, Spearman rank correlation. Pearson R correlation measures the degree of the relationship between linearly related variables. Its output is a number between -1 and 1, where 1 shows high correlation, zero means no correlation, and -1 demonstrates negative linear correlation [47]. Assuming  $X$  and  $Y$  are random variables, and the number of samples inside of each one is  $n$ , correlation coefficients can be written as follows:

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{X})(y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{X})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{Y})^2}} \quad (2.3)$$

where  $x_i$  and  $y_i$  are the individual sample points,  $\bar{X} = \frac{1}{n} \sum_{i=1}^n x_i$  is the mean of variable and analogously for  $\bar{Y}$  [47].

### Principal Component Analysis

The **Principal Component Analysis** (PCA) is a particular kind of unsupervised and widespread technique used in literally any ML problem. In ML problems, it is possible to face the high dimensional dataset while only a few features correspond to latent factors (variables that are not directly observed but are rather inferred), and makes processing a tough task. This is the main reason for using PCA [48].

In fact, low-variance components are often not helpful due to noise and consequently, they are almost not informative. PCA method finds the directions of maximum variance of the data and then rotates the dataset based on these directions. Considering each row of the dataset has  $D$  features, the steps to apply PCA can sort as follows:

1. Calculating the sample covariance matrix ( $\hat{\Sigma} \in \mathbb{R}^{D \times D}$ ) of dataset.

$$\hat{\Sigma} = \frac{1}{N} \sum_{i=1}^N x_i x_i^T \quad (2.4)$$

2. Performing eigendecomposition of covariance matrix and obtain eigenvalues and eigenvectors.
3. Ordering the eigenvalues from largest to smallest (eigenvalue essentially states the amount variance in the data along the eigenvector direction)
4. Selecting the best value of  $L$  from  $D$  that allows to keep the most informative vectors.

Selecting the best value of  $L$  (components) needs an iterative approach which is performed by looking at the cumulative explained variance ratio as a function of the number of components starting from 1 latent variable up to the number of features. Typically a good choice for  $L$  could be the minimum number of components that retain a desired percentage of the data variance.

It is imperative to mention that features must be standardized before applying PCA because the PCA calculates a new projection of the dataset, the new axis is based on the standard deviation of variables. So a variable with a high standard deviation will have a higher weight for the calculation of axis than a variable with a low standard deviation. By standardization the dataset, all variables have the

same standard deviation and same weight, and as a result, PCA calculates the relevant axis. In order to standardize the dataset, firstly it is needed to remove the mean value from each feature, then divide it with the standard deviation of that feature which ensures that every column in the dataset has a unit standard deviation [49].

## 2.2.2 Supervised Feature Selection

As mentioned, an important distinction between supervised and unsupervised feature selection is ignoring the outcome during the elimination of features or not. There are different techniques such as Wrapper, Filter, or Intrinsic methods. Wrapper methods use different subsets of input features, create and train a model, and select those subsets of features that result in the best performing model according to a performance metric. These methods are unconcerned with the variable types, therefore, wrapper methods train a new model for each subset, they can be very computationally intensive [50].

The filter methods use statistical techniques to evaluate the relationship(score) between each feature and the target variable. Then, those scores are used to select the features that have more correlation with output [51]. This method consists of three steps as follows [52]:

- **Screening:** This step removes variables and cases that do not provide helpful information for prediction and issues warnings about variables that may not be useful. For instance, variables that have all missing or constant values will be removed. If any row of the dataset has no output value, it will be eliminated from the dataset [51].
- **Ranking:** This step considers one feature at a time and check performance level of its prediction for target variable. Each variable's importance value is calculated as  $(1 - p)$ , where  $p$  is the  $p$ -value. The  $p$ -value is the probability of obtaining test results at least as extreme as the results observed, under the assumption that the null hypothesis is correct [53]. F-statistic is an appropriate statistic test to calculate  $p$ -value for continuous features(continuous variables are numeric variables that have an infinite number of values between any two values)[51].

The F-statistic simply denotes a ratio of two variances. The idea is to perform a one-way Analysis of Variance(ANOVA) for each continuous feature. ANOVA can determine whether the means of three or more groups are different and

uses F-tests to statistically test the equality of means [51]. The  $p - value$  based on the F statistic is calculated by:

$$p - value = Prob\{F(J - 1, N - J) > F\} \quad (2.5)$$

$F = \frac{\sum_{j=1}^J N_j(\bar{x}_j - \bar{\bar{x}})/(J-1)}{\sum_{j=1}^J (N_j-1)s_j^2/(N-J)}$	-
$N_j$	The number of cases with $Y = j$
$\bar{x}_i$	The sample mean of predictor $X$ for target class $Y = j$
$s_j^2 = \sum_{i=1}^{N_j} (x_{ij} - \bar{x}_i)^2 / (N_j - 1)$	The sample variance of predictor $X$ for target class $Y = j$
$\bar{\bar{x}} = \sum_{j=1}^J N_j \bar{x}_i / N$	The grand mean of predictor $X$

**Table 2.1**

- **Selecting:** this step identifies the important subset of features to use in subsequent models, once the predictors are ranked by  $p - value$  in ascending order[51].

## 2.3 Neural Network

A Neural Network(NN) is a network of artificial neurons that simulates how the human brain operates. A real NN in the brain is shaped by many neurons in different layers. Similarly, the artificial network is made up of several nodes distributed in various layers. Each node implements an instruction called algorithms that guides the machine in identifying patterns in the dataset. Such systems learn to do jobs by evaluating samples rather than receiving particular instructions given by programmers. Today, NN is used to solve many problems in different industries. For example, they are essential in computer training for face recognition, driving, demand forecasting, and more [54]. Generally, NN is classified into three categories listed in the following:

- **FeedForward:** FeedForward neural networks are also recognized as multilayer networks of neurons. These model networks are denominated as feedforward because the data is only transmitted forward in the NN toward input nodes, then into hidden layers, and finally into output nodes. There are no feedback communications in the feedforward networks to return the network output on their input.
- **Convolutional:** It is maybe the most widespread one and indeed a simplified form of feedforward NN. One of the simplification advantages is that it ends up with a much lower number of parameters. Training these networks also on extensive data such as images or videos can be done efficiently. This simplification is done using filters that are placed after the convolution operation [55].
- **Recurrent:** This type of network allows some outputs to be backup input for previous layers that creates loops in the network. By using loops, the network becomes like a state machine and learns new things from the memory of the past. Recurrent NN is working very well in time-varying systems, but the training process becomes much more complicated [56].

### 2.3.1 Learning Process of Neural Networks

The learning process in artificial neural network mainly depends on four factors[57]:

1. Number of layers in the network: the possible options for these factors are, single layer or multiple layers.
2. Signal flow direction: it can be Feedforward or Recurrent.

3. The number of nodes in layers: in the input layer, the number of nodes is equivalent to the input dataset features. The number of nodes that appear in the output layer will depend on the possible results. However, the number of hidden layers is selected by the user. The more nodes in the hidden layer, the higher the performance, yet using several nodes in this layer may cause overfitting as well as increase computational costs.
4. Weight of interconnected nodes: Deciding on the amount of weight attached to each connection among each node so that a particular learning problem can be properly solved has been one of the difficult tasks.

In summary, this process includes feeding labeled data to the network and changing the network's parameters such as learning rate, epoch, cost function to bring the results close enough to the desired output. On the other hand, the network's training process is an iterative approach where there is a cost function, and the aim is to minimize its value [57]. Although there are three different types of cost functions including, linear, quadratic, and cubic. The quadratic cost function has several useful features listed below:

- It is convex and smooth, which facilitates the evaluation of derivatives.
- In the case of linearity in the system, the controller design is usually translated as a kind of Ricatti Equation that is easy to study solutions and properties.
- When using nonlinear systems, Lyapunov functions are usually easy to use when using quadratic optimal control functions.

Formula 2.6 is a mathematical representation of cost function equation.

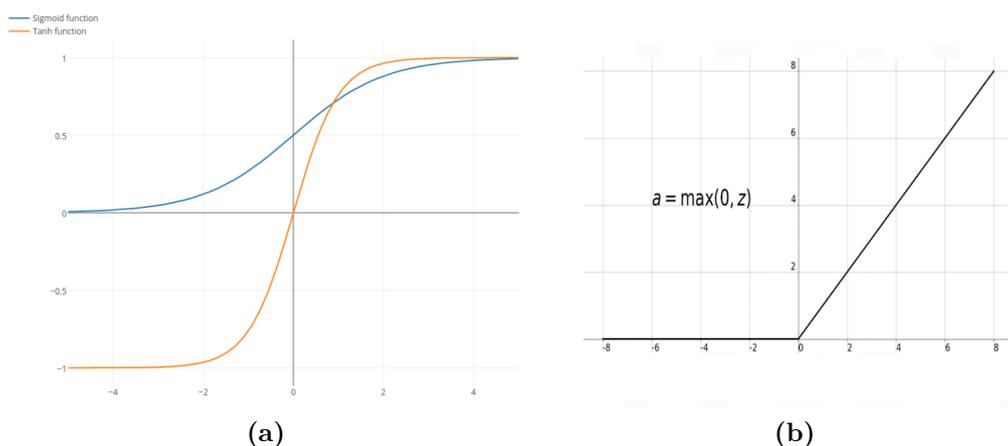
$$C(w, b) = \frac{1}{2n} \sum_x ||y(x) - a(x, w, b)||^2 \quad (2.6)$$

where the function  $a(\cdot)$  is the result of NN, and its parameters are  $x$  as the input of NN, while  $w$  and  $b$  represent weights and biases respectively for all neurons in the network. In particular, the activation function  $a_j^l$  is defined for each neuron( $j$ ) in each layer( $l$ ) of network. Equation 2.7 shows the details of the activation function. Where  $\sigma(\cdot)$  represents the activation function. In this compact equation,  $W^l$  indicates the weight matrix and vector  $b^l$  contains bias value for each node.  $j$  is the index of the neuron in the layer  $l$ .

$$a^l = \sigma(w^l a^{l-1} + b^l) = \sigma(z_l) \quad (2.7)$$

Training starts with initial random weights and biases, and then at each iteration, these parameters will be updated using Back-propagation, which is described in

section 2.3.3. There are many different activation functions with their pros and cons, such as perceptron, Sigmoid, Hyperbolic Tangent. One of the most widespread and popular activation functions for the neural network is called Rectified Linear Units (RELU). It is a non-linear function and constructed from the union of two linear tracks of functions. The positive input value of  $z$  will be returned unchanged, whereas an input value of 0 or a negative value will be returned as the value 0. Unlike the sigmoid function, learning does not slow down with a large weight, and learning stops entirely with a small weight [54].



**Figure 2.3:** Sigmoid, Hyperbolic Tangent and Relu activation functions

## 2.3.2 Minimization Problem

As explained in Section 2.3.1, the cost function is defined, and the aim is to find the optimum set of weights and biases that minimize the cost function. There are different algorithms for the minimization problem, and one of the most common and widespread one is Stochastic Gradient Descent (SGD), an updated version of gradient descent that solved the problem of stopping at local minimums. SGD starts with a random vector of weights and biases, and it goes toward the minimum point of function by using an iterative approach. Weights and biases at each iteration update using Formula 2.8 [54].

$$w_{k+1} = w_k - \eta \frac{\partial c}{\partial w_k} \quad b_{l+1} = b_l - \eta \frac{\partial c}{\partial b_l} \quad (2.8)$$

where  $\frac{\partial c}{\partial w_k}$  and  $\frac{\partial c}{\partial b_l}$  are the partial derivative of the cost function with respect to weights and biases.  $\eta$  is a positive coefficient, called **Learning Rate**, which has to be small enough so that the Taylor approximation holds and avoids the algorithm

diverging. However, the small value of the learning rate requires much time for training [54].

To calculate the derivatives and apply changes in the weights, the whole training set must be considered because the average is taken over all training data. In practical cases, the training set is enormous, and it will take much time to run the training algorithm. Unlike gradient descent, SGD calculates the cost function for a small subset of the training set, not for the whole training set. If the subset is small enough, then this would be a good approximation [58].

This algorithm picks  $m$  data vectors randomly and calculates all averaging of the cost function and the derivatives only for this subset known as **mini-batch**. When the mini-batch calculation is finished, other  $m$  samples that have not been included in previous mini-batches are chosen randomly for the next iteration. Once an **epoch** is finished, meaning every sample in the training set has been used only once, then the algorithm goes to the next epoch, and it picks new mini-batches.

Another advantage of SGD is that it estimates noisy derivatives because sometimes it is possible to overshoot a little bit in its estimation of the optimal  $\Delta w$ , and it allows the algorithm to explore different local minima by moving away from the current solution [58].

### 2.3.3 Backpropagation algorithm

Backpropagation is a straightforward algorithm that updates the weights and biases iteratively to minimize a given cost function. In order to do that, the algorithm needs to know some values for partial derivatives of the cost function with respect to any parameter in the network. These derivatives represent the amount and direction of changes in the cost function [59].

Every iteration of the algorithm consists of two parts: The first one is called the feedforward pass, where it simply uses the neural network, and the other is the back-propagation. The steps are as follows [59]:

1. The input of the algorithm is a given training instance. in practice, it will typically be a mini-batch that is the activation at the first layer. Then initial values of the weights and biases for all neurons of all layers are needed. There is no knowledge about the optimal value of weights and biases in the most general cases, so they are chosen randomly.
2. Then, in the feedforward pass, activation functions  $z^l$  are calculated from the first layer to the last layer  $L$ .

3. When all activations are derived, it is possible to calculate the neurons' error in the last layer  $\delta^L$ .

$$\delta_j^L = \frac{\partial c}{\partial a_j^L} \sigma'(z_j^L) \quad (2.9)$$

In the equation , the first term indicates how much the cost is changing as a function of the last layer's activation output. The cost should change as a function of activation because the result of the activation function in the last layer is simply the answer of the neural network, which can be compared with the ground truth and obtain our cost. The second term indicates how fast the activation result is changing at a specific layer.

4. The algorithm backpropagates the error for every layer starting from  $l - 1$  by evaluating  $\delta_{l-1}$  and then all intermediate parameters of the network are calculated.
5. In the last step, the derivatives for all parameters in the network are evaluated. Finally, stochastic gradient descent is applied to update the values of the network's parameters. Once all these five steps are done, and weight and bias values are updated, the algorithm will go to step 2. These iterations continue since the desired performance of the network is achieved.

### 2.3.4 Cost Function

One problem of the quadratic cost is that sometimes it learns slowly and takes too many computational resources to complete the training for a computationally-intensive problem. In general, when the output is very wrong, due to the random choice of the parameters, the cost function decreases very slowly.

When the cost function is very large, neurons are working in saturated regions, and getting out of the saturated region takes much time. One of the possible ways to overcome this problem is to change the cost function. So, in classification problems, the **Cross-Entropy Cost Function** can be used, which is defined in equation where  $n$  is the number of training examples, and it can be proved that the cost function is always positive [60].

$$C = -\frac{1}{n} \sum_x [y \ln(a) + (a - y) \ln(1 - a)] \quad (2.10)$$

### 2.3.5 Regularization and Dropout

A deep neural network is simply a NN with many layers, neurons, and a massive number of parameters. As the number of parameters increases, it is likely that model overfits, and as a consequence, the accuracy drops. In some NN cases, the result of cost function will constantly decrease during the training and plateau at a certain point. By evaluating the trained network with test data, the cost function's result reduces until it reaches the minimum value, then it starts growing. It can be proved that the NN is not learning any more after a specific epoch [61].

There are different approaches to avoid overfitting, such as using a smaller network (smaller number of layers and neurons), early stopping (just monitoring the trend of cost function and choosing the minimum one and then stop), or using more data for training the network (using more information is sometimes impossible, and very expensive.)

Therefore, **regularization** is a common method to reduce the chance of overfitting and improves model performance. This technique uses a linear combination of the old cost functions  $C_0$  (quadratic, cross-entropy) plus a new term  $L^2$  which is the norm of the weights in the NN (the sum of the quadratic value of all weights in the NN) [62].

$$C = C_0 + \frac{\lambda}{2n} \sum_w ||w||^2 \quad (2.11)$$

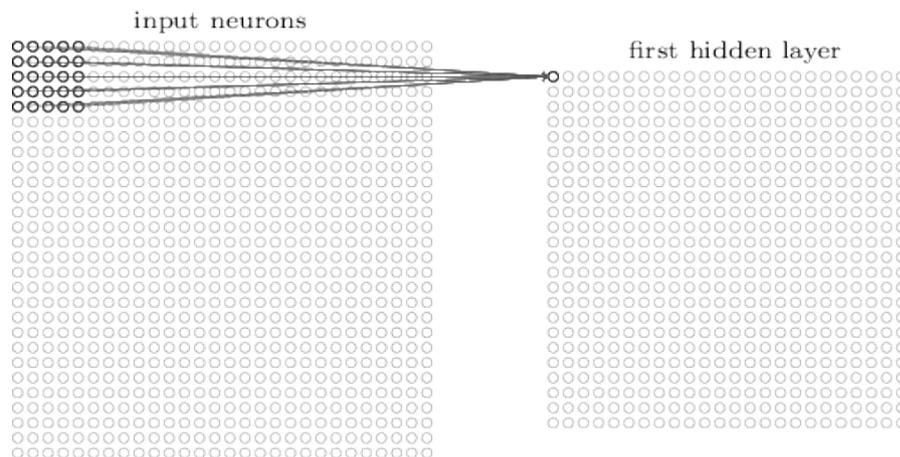
In deep learning, the goal is to minimize the loss function. In L2 regularization, a component that tunes large weights will add to the cost function. The equation 2.11 shows the added component, and lambda is the regularization parameter. By adding the squared norm of the weight matrix and multiplying it by the regularization parameters, large weights will decrease to minimize the cost function.

Another extremely popular solution in training neural networks is a so-called **dropout**. It is an aggressive way of fighting against overfitting: essentially, the NN is divided into two subnetworks, so every neuron for every layer is associated with either NN-A or NN-B. Both networks are trained simultaneously, and once training is done, they are combined in the final network. This technique works well because it is a sort of averaging of all bad things that can happen in training over the network taken singularly, it is similar to averaging the noise [63].

### 2.3.6 Convolutional Neural Networks

What has made neural networks so famous and influential is the Convolution Neural Network(CNN): the name comes from after the convolutional operator from the filtering domain. it can filter the input vector extracted from signals, images and other possible type of data, with a linear and timing-variant filter. The CNN becomes powerful in solving different problems by implementing three important concepts.

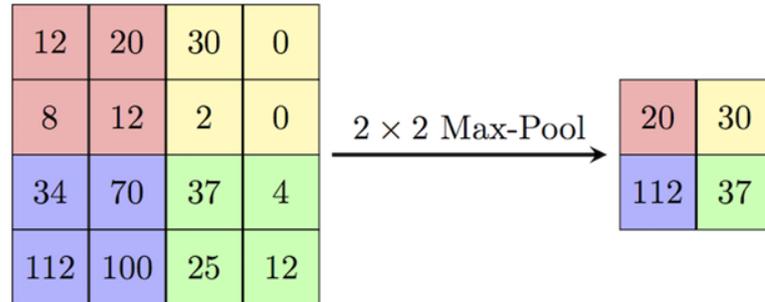
- **Local Receptive Fields:** This concept explains using of a new design in such a way that each neuron in the next layer is connected to a subset of the outputs of the previous layer, instead of using a fully connected neural network, such that each neuron in the next layer connects to all neurons in the previous layer. It is rather evident that the number of parameters decreases tremendously by using this technique [55].



**Figure 2.4:** Local Receptive Fields [36]

- **Shared weights/biases:** Besides the limited number of input samples for each neuron, the concept of shared weights implies that all weights in each window of neurons are the same. The number of parameters is reduced because neurons share the same weights [55]. The connections from the local receptive field to each neuron have all the same weights:
  - All neurons in the first layer detect the same feature at different locations in the image
  - This exploits (possible) shift-invariance of the image content
  - The shared weights/bias define a filter

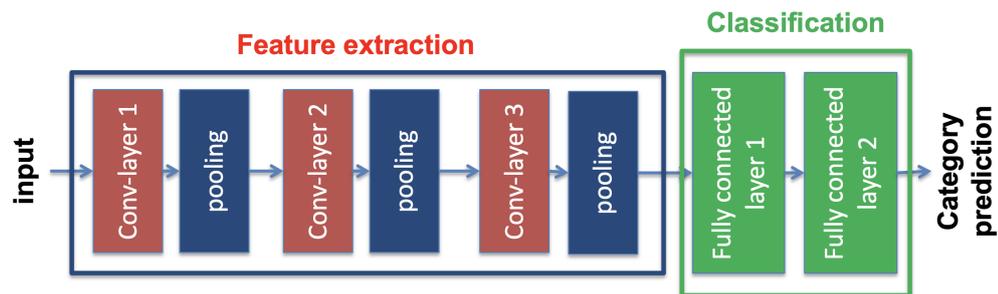
- **Pooling Layers:** A pooling layer is essentially a downsampling layer and reduces the data dimensions by combining the outputs of neuron clusters at one layer into a single neuron in the next layer. Local pooling combines small clusters, typically two by two. Figure 2.5 represents a sample of max pooling layer [54].



**Figure 2.5:** Sample of Max Pooling [36]

The last layer of the CNN (last one or two layers) takes all inputs from convolutional layers, combines them in a fully connected way to classify images, and usually employs a softmax to come up with decisions about the class that the image belongs. It means the network can be split into two distinct parts, the convolutional part of the network which extracts some discriminative features of the image, and the classifier part, in which those features are classified.

Figure 2.6 shows a complete CNN. A sequence of convolutional layers and pooling layers, as many as needed (not more than thousands that are used only in a specific puposes, called ResNet), is followed by one or more fully connected layers.



**Figure 2.6:** Complete Layers of CNN (Number of layers can be rather large)[36]

# Chapter 3

## Simulation

GNSS and chirp signals are simulated by using NFUEL [64], and Matlab applications [65].

### 3.1 GNSS Signal

In order to simulate GNSS signal, N-FUELS (FULL Educational Library of Signals for Navigation) application has been used, which is a signal/disturbances generator, implementing as a set of non-real-time MATLAB scripts able to simulate the samples of a GNSS signal as seen by the receiver after the A/D (Analog to Digital) conversion.

The received signal at the first stage of the receiver(antenna) is a continuous-time signal. However, in order to characterize in digital signal processing, the signal needs to be transformed to discrete-time and is widely known as a sampling signal. The sampling frequency or sampling rate,  $F_s$ , is the average number of samples obtained in one second (samples per second). The sampling frequency in the GNSS signal simulation scenario is 40 MHz. It means each second of signal in continuous-time can be represented with 40 million points in discrete-time. The duration between each sample as follows:

$$T_s = \frac{1}{F_s} \quad (3.1)$$

The simulated GNSS signals belong to the GPS constellation and L1 band (1575.42MHz). Each satellite within the GPS constellation has a unique Pseudo-random Noise Code (PRN code) [66] which allows any receiver to identify precisely the transmitted satellite of the signal. 32 GNSS signals are generated referring to PRN-1 up to PRN-32. Some of the main characteristics of the simulated GNSS

signals are sorted in Table 3.1.

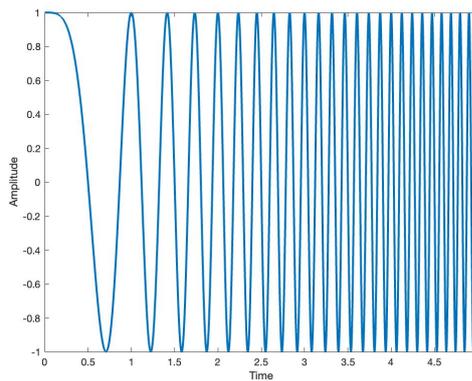
Parameter	Value
Modulation	GPS L1
Signal Length	1 s
Sampling Frequency	40 MHz
Satellite PRN codes	1 - 32
Code Delays*	0.0004 s
Received Signal Power	-157 dBW
Carrier to Noise Ratio $C/N_0$	45 dBHz

**Table 3.1:** Main Characteristics of GNSS Signals (\* Code delay with respect to the ideal start instant of the first PRN code chip)

The output of GNSS simulation is a set of binary files (.bin format) for each signal. Note that the points of sampled GNSS signal are complex numbers and composed of real and imaginary parts.

## 3.2 Chirp Signal

Chirp is a signal in which its frequency varies with respect to the time, thus the frequency increment or decrement known as Up-chirp or Low-chirp, respectively. This type of signal is known as the swept frequency signal. Figure 3.1 shows a simple chirp signal whose frequency changes from 1 up to 10 Hz in a period of 5 milliseconds.



**Figure 3.1:** Simple Chirp Signal

In term of mathematics, assuming chirp signal is defined as  $X(t) = \sin(\phi(t))$  where depending on how the frequency is changing,  $\phi(t)$  can be linear or non-linear. The instantaneous angular frequency is the first derivative of  $\phi(t)$ , and if the derivative is divided by  $2\pi$ , the frequency in hertz can be obtained.

$$\omega(t) = \frac{d\phi(t)}{dt} \quad , \quad f(t) = \frac{\omega(t)}{2\pi} \quad (3.2)$$

### 3.2.1 Chirp Classifications

The swept-frequency jammers are distinguished by their ability to generate overwhelming signals with carrier frequencies that vary over GNSS signal bands, causing disruptive effects on GNSS receiver performance [9]. According to Strike3 project [67], chirp signal characteristics can classify in terms of shape, sweep range, sweep rate, and power. The chirp simulation scenario follows draft standards developed by STRIKE3 to create different types of chirp signals.

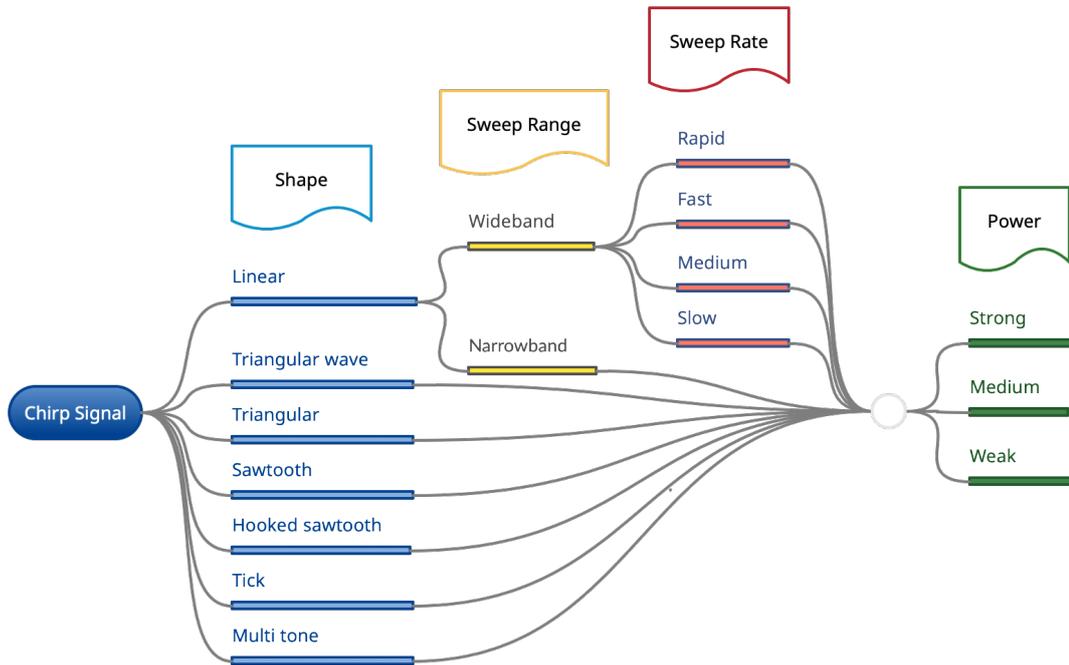
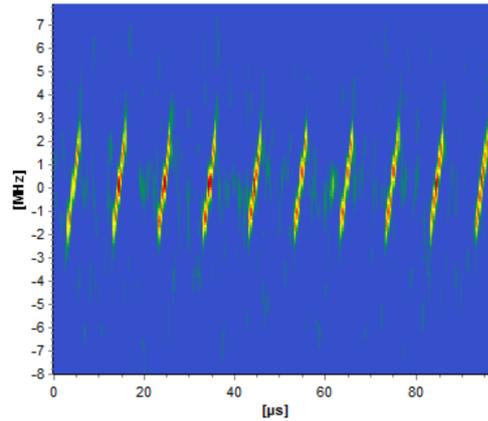


Figure 3.2: Chirp Signal Classifications

## Chirp Shapes

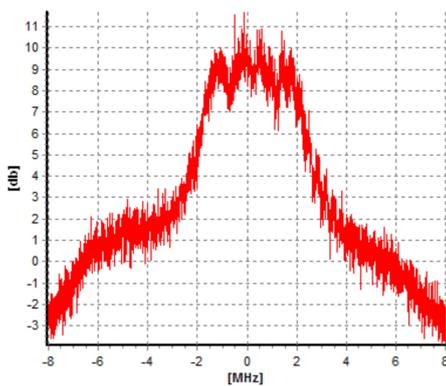
The chirp signal's shape is directly related to the sweep frequency equation mentioned, such as linear, triangular. One way to see the shape of the signal is by plotting the spectrogram of the signal. For instance, figure 3.3 represents the spectrogram of the linear chirp signal. Different shapes of chirp signal and their characteristics in spectrogram and spectrum plot can be found in appendix A.



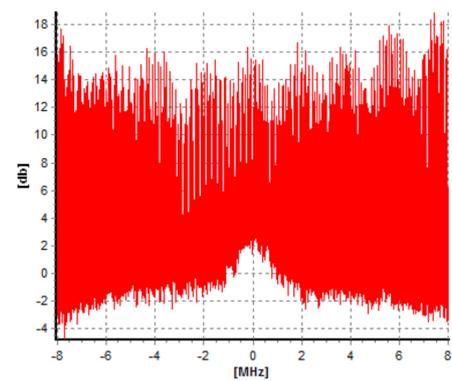
**Figure 3.3:** Narrow-band Chirp Signal [68]

## Chirp Sweep Range

The linear chirp signal can be classified in terms of the bandwidth into narrow-band and wide-band with respect to the GNSS signal bandwidth. In simulation scenario bandwidth of narrow and wide chirp signals are 5 and 16 MHz, respectively.



**Figure 3.4:** Narrow-band Chirp



**Figure 3.5:** Wide-band Chirp

### Chirp Sweep Rate

The repetition number of one cycle of chirp signal (varying from initial frequency up or down to target frequency) in 100  $\mu s$  time window of a wideband chirp signal is known as sweep rate. They are categorized into four classes as follows:

- Rapid: 13 - 16 chips in 100  $\mu s$
- Fast: 8 - 12 chips in 100  $\mu s$
- Medium: 4 - 6 chips in 100  $\mu s$
- Slow: 2 - 3 chips in 100  $\mu s$

### Chirp Power

The power of chirp signals can be categorized with respect to the received signal's power in GNSS receivers as follows:

- Strong: -127 up to -107 dBW.
- Medium: -147 up to -127 dBW.
- Weak: -157 up to -147 dBW.

## 3.3 Combining with CHIRP

According to chirp classification, section 3.2.1, different types of chirp signal is created by using Matlab script as follow:

- Wide linear chirp (sweep rate: 10 in 100  $\mu s$ )
- Narrow linear chirp
- Triangular chirp
- Triangular wave chirp
- Sawtooth chirp
- Hooked sawtooth chirp
- Tick chirp
- Multi-tone chirp

The sweep rate can vary from 2 up to 16 chirp repetitions in 100 microseconds in the linear wide-band chirp signal. In the simulation, a linear wide-band chirp signal with ten chirp repetitions in 100 microseconds is only considered to have a normal distribution in terms of the number of signals in each class. Totally eight different types of chirp signals are generated which their sampling frequency and duration are the same as the simulated GNSS signal. Despite the different chirp

signal shapes, chirp signals' power is amplified from -157 dBW up to -107 dBW incrementing by one (49 cases), which represent weak, medium, and strong power.

- The 32 GNSS signals are simulated with duration of one second.
- Each GNSS simulated signals is divided into ten equal time-based portions, thus it is expected to derive 10 new signals with 100 ms duration from each signal of satellites.
- Then only one of ten portions of divided signals is picked in order to combine with the chirp signals.
- The picked GNSS signals combine with the chirp signals (totally eight cases) in an additive way as follows:

$$Combined_{signal} = chirp_{signal} + original_{signal} \quad (3.3)$$

Assuming power of chirp signals is amplified from -157 dBW up to -107 dBW incrementing by one (49 cases).

- The total constructed signals based on each GNSS signal would be 402, which consists of 10 signals result in the dividing of every GNSS signal into ten portions without combining to chirp signal and 392 (8x49=392) combined signals.
- The final dataset consists of 12864 different signals.

Figure 3.6 shows the pseudocode or algorithm behind combining signals.

```

Data: GNSS-Signals and Chirp-Signals
Result: Combined-Dataset
for Original-Signal ∈ GNSS do
  for signal ∈ ChirpSignals do
    for Power ∈ (-157 : -107) do
      chirp = chirp × power
      final-signal = Original-Signal + chirp;
      add: final-signal to Combined-Dataset
    end
  end
end

```

**Figure 3.6:** Pseudocode of Combining GNSS signal with Chirp Signals

# Chapter 4

## Feature Extraction

Feature extraction involves reducing the number of resources required to describe a large set of data or extracting an event's information in a readable way for machines. The forming of signals' dataset is described in Chapter 3.3, each signal of the dataset needs to be analyzed using Digital Signal Processing (DSP) in time and frequency domains separately. Also applying Wavelet or Wigner-Ville transformations represents the signal in the time-frequency domain.

Matlab has been used to extract the features. An object-oriented program is developed where its input is the simulated signals, and output is a series of features describing the signal. In the following subsections, different DSP techniques of feature extraction are discussed.

### 4.1 Features in Time Domain

Time-domain signal processing involves analyzing the variation of the amplitude(or peak) of signal with respect to time. Considering the GNSS signal's complex sinusoidal shape, features related to the amplitude are extracted individually for real and imaginary parts by using statistical parameters. The features with a short description are sorted as follows:

- **Mean Value:** is adding up all the observed value and dividing by the number of them or often simply described as the "average".
- **Median Value:** is the middle of a sorted list of values, or it can also be mentioned that it is a threshold value which separates the higher half from the lower half of values. Comparing to mean value, median is not skewed so much by a small proportion of extremely large or small values.

- **Standard Deviation:** shows the amount of variation or dispersion of values with respect to the mean value. For instance, the low value of standard deviation indicates the value is tended near to the mean value.
- **Mean Absolute Deviation:(MAD)** is the average distance between each value and the mean in observed data set. Mean absolute deviation is a way to define variants in a data set.
- **Skewness:** describes how a data distribution leans. considering the normal data distribution(Gaussian distribution) which the mean is in the middle of data, if skewness is negative, the mean will be shifted to left and if it is positive, the peak tends to the right and the tail goes to left [69].
- **Kurtosis:** Like skewness, it describes the distribution. Whereas skewness differentiates extreme values in one versus the other tail, kurtosis measures extreme values in either tail. Distributions with large kurtosis exhibit tail data exceeding the tails of the normal distribution [70].

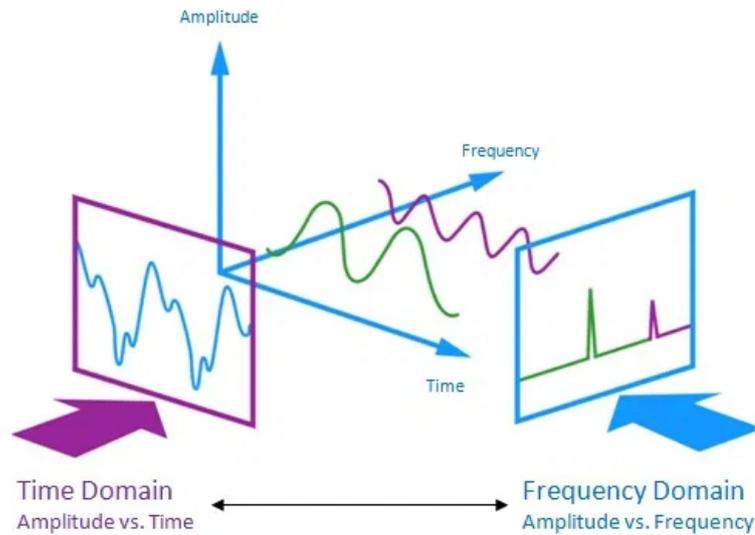
Domain	Feature Name	Name Meaning
Time (14 Features)	<i>Time_mean_real</i>	Mean of amplitude of Real part
	<i>Time_med_real</i>	Median of amplitude of real part
	<i>Time_std_real</i>	Standard Deviation of amplitude of real part
	<i>Time_mad_real</i>	Mean Absolute Deviation of amplitude of real part
	<i>Time_var_real</i>	Variance of amplitude of real part
	<i>Time_skw_real</i>	Skewness of amplitude of real part
	<i>Time_krt_real</i>	Kurtosis of amplitude of real part
	<i>Time_mean_img</i>	Mean of amplitude of Imaginary part
	<i>Time_med_img</i>	Mean of amplitude of Imaginary part
	<i>Time_std_img</i>	Median of amplitude of Imaginary part
	<i>Time_mad_img</i>	Standard Deviation of amplitude of Imaginary part
	<i>Time_var_img</i>	Variance of amplitude of Imaginary part
	<i>Time_skw_img</i>	Skewness of amplitude of Imaginary part
	<i>Time_krt_img</i>	kurtosis of amplitude of Imaginary part

**Table 4.1:** Feature Extracted in Time Domain for GNSS Signals

## 4.2 Features in Frequency Domain

In the time domain, extracting the features is limited to just statistical parameters, whereas in the frequency domain can be obtained more informative characteristics of the signal. However, the frequency domain of a signal means each signal consists of at least one component at a given frequency, representing the signal's repetition

in 1 second. Therefore, in a complex signal, more than one component creates the signal, which indicates the signal is combination of several signals with different frequencies known as the Harmonic of the primary signal. Figure 4.1 illustrates a signal with two harmonics while every single component spreads out in the time domain as distinct impulses in the frequency domain.



**Figure 4.1:** Time Domain vs Frequency Domain [71]

As depicted in Figure 4.1, a time-domain graph shows how a signal changes over time, whereas a frequency-domain graph displays how much of the signal lies within each given frequency band over a range of frequencies. Moreover, looking at a signal from the point of view of frequency can often give an intuitive description of the signal’s qualitative behavior. Besides, in frequency domain representations, the individual components are often much more closely related to the properties we care about.

***How to convert Time-Domain to Frequency-Domain?*** A given function or signal can be converted between the time and frequency domains by mathematical operations, known as transforms. The Laplace and the Fourier transform are the most famous transformation where Fourier Transform is a particular case of the Laplace Transform, so the properties of Laplace transforms are inherited by Fourier transforms. Joseph Fourier proved that any periodic signal could be decomposed into a summation of sinusoid waves. In other words, Fourier transforms convert a time function into a sum or integral of sine waves of different frequencies. This representation of the signal is known as the spectrum of the signal.

Fast Fourier Transform(FFT) is an approach to compute Discrete Fourier transform (DFT). This approach can be converted to a reverse procedure with respect to going from the frequency domain to the time domain by using Inverse DFT. In term of mathematics descriptions, considering  $x_0, x_1, \dots, x_{N-1}$  be complex numbers, DFT is defined as follows:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-2\pi i k n / N} \quad k = 0, 1, \dots, N - 1 \quad (4.1)$$

The output  $X_K$  is a complex number which encodes the amplitude and phase of a sinusoidal wave with frequency  $k/N$  cycles per unit of time where it comes from Euler's formula as follows:

$$e^{2\pi i k n / N} = \cos(2\pi i k n / N) + i \sin(2\pi i k n / N) \quad (4.2)$$

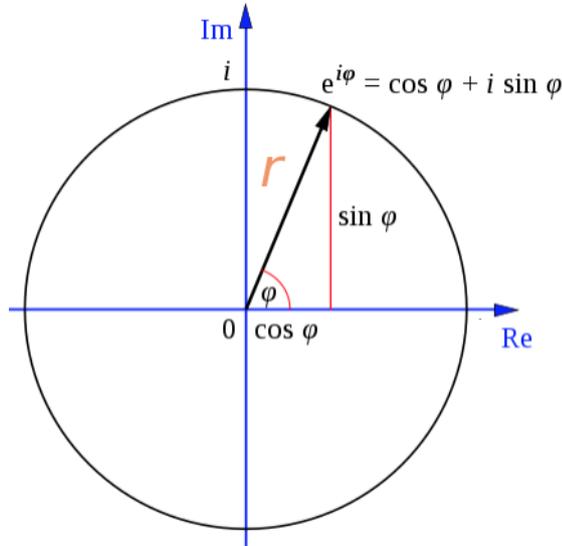
After transformation, since the DFT output is a sequence of complex numbers, the mean and variance of the magnitude and phase of points are calculated and inserted into the features vector. Then the real and imaginary parts will be separated, and statistical parameters (mean, variance, standard deviation, maximum and minimum) of them are derived as features.

Domain	Feature Name	Name Meaning
<b>Frequency</b> (16 Features)	<i>Freq_mean_real</i>	Mean of Magnitude of Real part
	<i>Freq_var_real</i>	Variance of Magnitude of Real part
	<i>Freq_std_real</i>	Standard Deviation of Magnitude of Real part
	<i>Freq_max_real</i>	Maximum of Magnitude of Real part
	<i>Freq_min_real</i>	Minimum of Magnitude of Real part
	<i>Freq_mean_img</i>	Mean of Magnitude of Imaginary part
	<i>Freq_var_img</i>	Variance of Magnitude of Imaginary part
	<i>Freq_std_img</i>	Standard Deviation of Magnitude of Imaginary part
	<i>Freq_max_img</i>	Maximum of Magnitude of Imaginary part
	<i>Freq_min_img</i>	Minimum of Magnitude of Imaginary part
	<i>Freq_mean_cpx</i>	Mean of Magnitude of Complex Signal
	<i>Freq_var_cpx</i>	Variance of Magnitude of Complex Signal
	<i>Freq_max_cpx</i>	Maximum of Magnitude of Complex Signal
	<i>Freq_min_cpx</i>	Minimum of Magnitude of Complex Signal
	<i>Freq_mean_ph_cpx</i>	Mean of Phase of Complex Signal
	<i>Freq_var_ph_cpx</i>	Variance of Phase of Complex Signal

**Table 4.2:** Feature Extracted in Frequency Domain for GNSS Signals

Magnitude is the distance of the complex point from the origin, and phase is the angle between the line that passes through the point in the complex coordinate and the positive real axis. Assume  $a = x + iy$  is a complex number, magnitude( $r$ ) and phase( $\phi$ ) are shown in Figure 4.2 and can be calculated from Formula 4.3.

$$r = \sqrt{x^2 + y^2} \quad \phi = \arctan\left(\frac{y}{x}\right) \quad (4.3)$$



**Figure 4.2:** Magnitude and Phase of Complex Number

In the signal processing domain, the digitalized signal’s magnitude sequence is known as **Power Spectral Density**, which describes the distribution of power into frequency components composing the original signal.

### 4.3 Features by Wavelet Transform

The previous section explained that the Fourier transform provides frequency information of signal while it does not provide information about frequencies existence in the time domain, so it is ideal for a stationary signal (a signal without frequency changes with respect to time). Since the frequency of chirp signals change over time, it needs other techniques to describe the signals. The Wavelet Transform(WT) translates the time-amplitude representation of a signal to a time-frequency representation encapsulated as a set of wavelet coefficients [72]. The concept of WT comes from **Short Time Fourier Transform** (STFT).

STFT comes to overcome the poor time resolution of the Fourier transform and convert signals from the time domain into a time–frequency representation. According to STFT, considering some portion of the non-stationary signal as stationary makes it possible to take window function of fixed length and move it along the signal [73]. Comparing to Fourier transform in terms of mathematic, formulation looks as follow:

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-i\omega t} dt \quad (4.4a)$$

$$F(\tau, \omega) = \int_{-\infty}^{\infty} f(t)w(t - \tau)e^{-i\omega t} dt \quad (4.4b)$$

where  $\tau$  is translation parameter and  $w(t - \tau)$  represents window function. The only difference with respect to Fourier Transform is the window function where the signal is smoothed. There are two main limitations of using STFT:

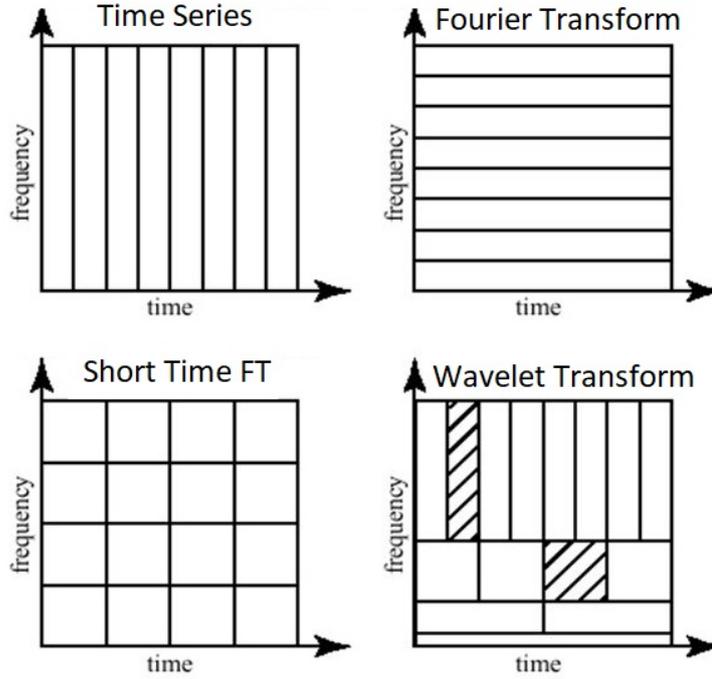
- The window function is finite, so frequency resolution decreases.
- The windows' fixed-length means time and frequency resolution are fixed for the entire length of the signal. If a narrow window is chosen, it is suitable for time resolution but poor in frequency resolution, and in contrast, choose a wide window is proper for frequency and poor in time resolution. In other words, low-frequency components last for an extended period, so a high-frequency resolution is needed and vice versa [74].

WT offers properties to overcome these two limitations, where it analyzes signal into different frequencies at the different resolutions known as Multi-Resolution analysis. Figure 4.3 shows the time and frequency resolution of the different signal transformations.

By definition, the WT of a signal  $f(t)$  is the correlation between the signal and a set of basic wavelets  $\psi(t)$  known as Mother Wavelet. The following Formula expresses WT:

$$F(\tau, s) = \frac{1}{\sqrt{|s|}} \int_{-\infty}^{\infty} f(t)\psi^*\left(\frac{t - \tau}{s}\right) dt \quad (4.5)$$

where  $s$  is scale parameter ( $1/\text{frequency}$ ) and  $\psi^*$  is conjugate of the wavelet function. Generally, different types of wavelets can be chosen with respect to the original signal's shape. These wavelets are new basis functions compared to Fourier transform, where *sin* and *cosine* are basic functions. Wavelets act as a window function where their width and central frequency can change as it moves across the signal. All the windows are compressed versions of mother wavelet by  $s$  and shifted (by  $\tau$ ) version of mother Wavelet. However, the output of Formula 4.5



**Figure 4.3:** Comparison of Different Transformations [75]

increases when the original signal is similar to the wavelet [76]. The same concept is valid also for discrete signal known as Discrete Wavelet Transform(DWT), and its formulation is as follows:

$$D[\tau, s] = \frac{1}{\sqrt{s}} \sum_{m=0}^{p-1} f[t_m] \psi\left[\frac{t_m - \tau}{s}\right] \quad (4.6)$$

where  $\tau = k2^{-j}$  and  $s = 2^{-j}$ ;  $j$  is scale index and  $k$  is wavelet transform signal index. If  $s$  and  $\tau$  chooses based on the power of two(dyadic), then analyses become much more efficient and accurate.

DWT is a multi-level decomposition where the signal passes through high-pass and low-pass filters. The approximation and detail coefficients of wavelet are calculated after the low-pass and high-pass filter, respectively. After each low-pass filter, the procedure is repeated literately at each level. Figure 4.4 shows a DWT procedure [76].

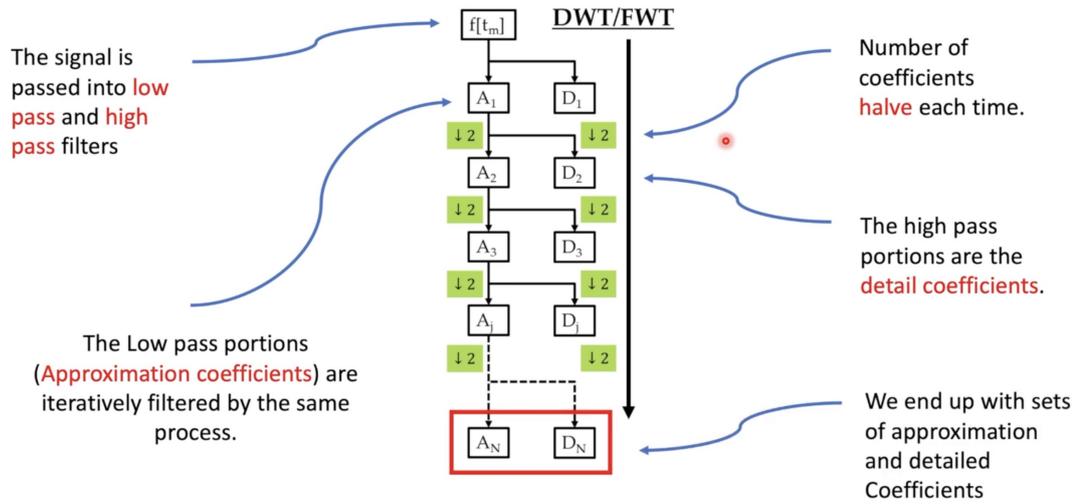


Figure 4.4: Multi-Level Decomposition [76]

Since the GNSS signal is complex sinusoidal, Complex WT is used, a simple extension to the DWT. **Dual-Tree Complex Wavelet Transform** is an approach to handle complex data that analyze the real and imaginary components as two separate signals and perform a DWT on each component separately. It is called "dual-tree" because there are two DWT decompositions in parallel, one for the real components and one for the imaginary [77].

As clarified before, there are different families of wavelets(to see different families refer to [78]), where each one has special properties for different types of signals. The Symlet (sym4) is chosen for this case, and its properties are: nearly symmetrical, similar to the DB family, orthogonal, biorthogonal.

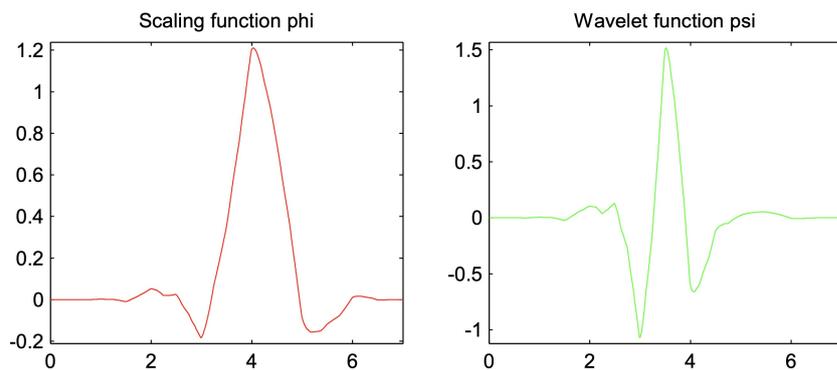
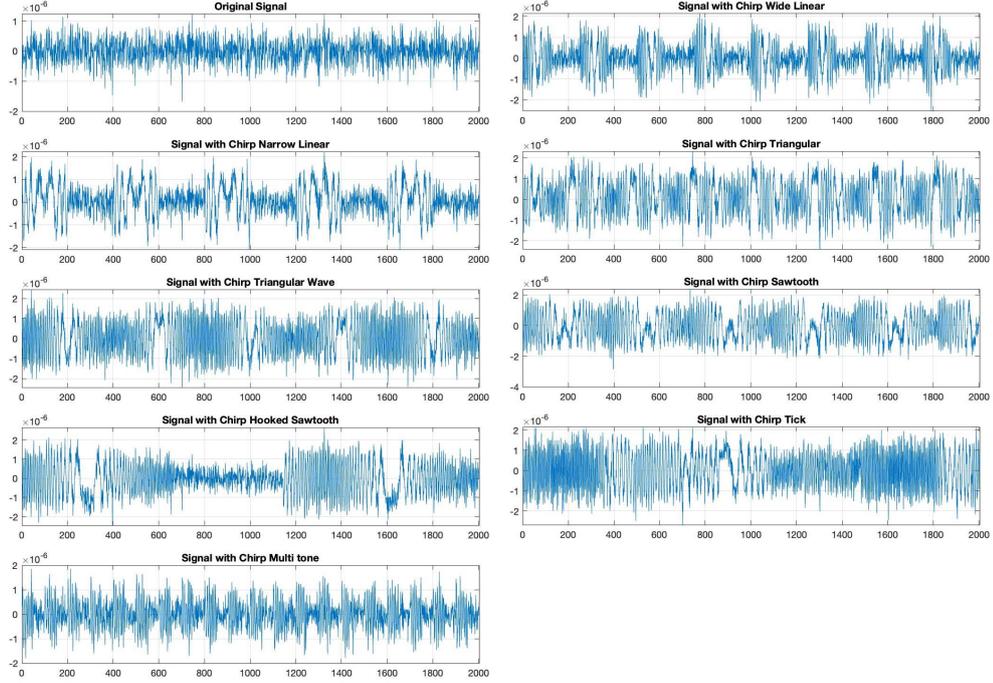


Figure 4.5: Sym4 Wavelet

Once the signal is transformed, the mean and variance of approximation and details coefficients are extracted for each real and imaginary part separately to characterize the output as shown in Table 4.3. Figure 4.6 illustrates approximation coefficients of wavelet analyses for the real part of the original and combined signal with different chirp types.



**Figure 4.6:** Wavelet analyses over original and combined signal with different types of chirp

Domain	Feature Name	Name Meaning
Wavelet (Time-Frequency) (8 Features)	$WL\_cA\_mean\_real$	Mean of Approximation Coefficients of Real Part
	$WL\_cA\_var\_real$	Variance of Approximation Coefficients of Real Part
	$WL\_cD\_mean\_real$	Mean of Details Coefficients of Real Part
	$WL\_cD\_var\_real$	Variance of Details Coefficients of Real Part
	$WL\_cA\_mean\_img$	Mean of Approximation Coefficients of Imaginary Part
	$WL\_cA\_var\_img$	Variance of Approximation Coefficients of Imaginary Part
	$WL\_cD\_mean\_img$	Mean of Details Coefficients of Imaginary Part
	$WL\_cD\_var\_img$	Variance of Details Coefficients of Imaginary Part

**Table 4.3:** Feature Extracted by Wavelet Transform for GNSS Signals

## 4.4 Features by Wigner-Ville Transform

Basically, there are two kinds of time-frequency analyses:

- Linear approaches, including the Wavelet, Gabor, Zak transform.
- Quadratic methods covering the time-frequency distributions such as Wigner distribution, the ambiguity function, smoothed versions of the Wigner distribution.

This section is concentrated on the Wigner-Ville Transform (WVT), a time-frequency representation of the signal and a helpful tool to analyze non-stationary signals. The basic idea of this method is to develop a joint function of time and frequency like the wavelet transform.

From theoretical and application points of view, the WVT or the Wigner-Ville Distribution (WVD) plays a major role in the time-frequency signal analysis for the following reasons. First, it provides a high-resolution representation in both time and frequency for non-stationary signals. Second, it has the special properties of satisfying the time and frequency marginals in terms of the instantaneous power in time, energy spectrum in frequency, and total energy of the signal in the time and frequency plane [79].

WVT essentially computes the Fourier transform of the so-called Auto-correlation function (AF) [ $AF(\tau) = x(t + \tau/2)x^*(t - \tau/2)$ ] which is a general representation of the signal's autocorrelation function. The WVT does not suffer from leakage effects as the STFT does and hence, gives the best spectral resolution. Generally, for a single time series (zero-mean), the Wigner Ville function is simply given by [80]:

$$WVC(t, f) = \int_{-\infty}^{\infty} x(t + \frac{\tau}{2})x^*(t - \frac{\tau}{2})e^{-2\pi i\tau f} d\tau \quad (4.7)$$

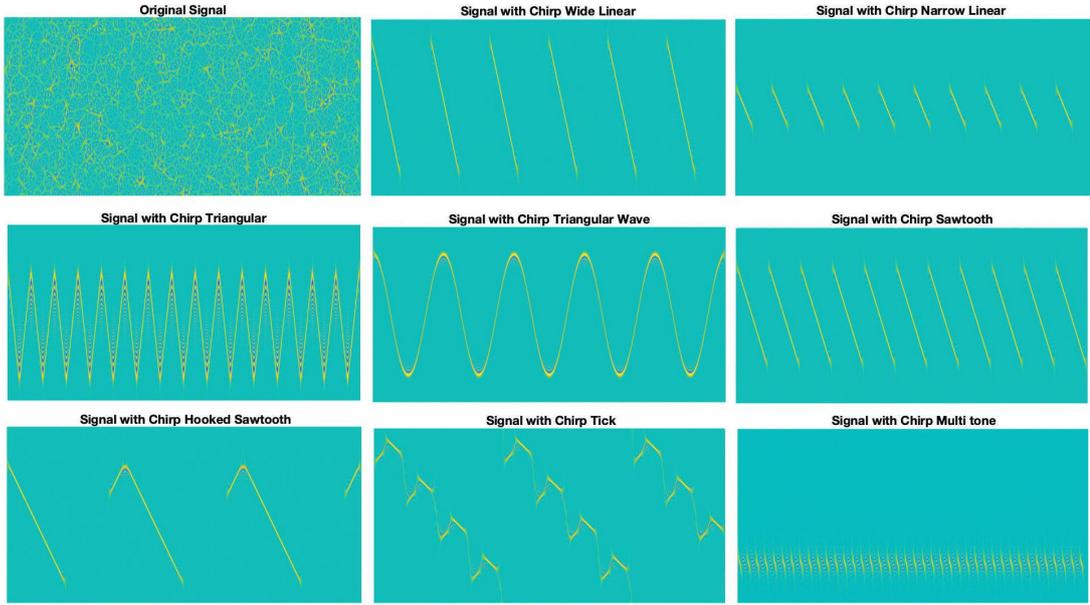
where  $x^*$  denotes the conjugate of  $x$ . There are several presentations in the literature for the Discrete Wigner-Ville Transform (DWVT), the general form can be written as Formula 4.8 [81] for a signal  $x[t]$  where it is sampled by  $N$  number.

$$WV_x[n, m] = \sum_{k=-N}^N x[n + \frac{k}{2}]x^*[n - \frac{k}{2}]e^{-2\pi i km/N} \quad (4.8)$$

where  $n$  and  $m$  denote the index numbers for the time and frequency vectors, respectively. To see more detail of Wigner Ville Distribution refers to [82] [83] [84].

In order to avoid the cross-terms effect in quadratic form, smoothed pseudo Wigner distribution [85] [86] is used to analyse the signal which uses independent windows to smooth in time  $g(n)$  and frequency  $h(m)$  and its formulation is as follow:

$$WV_x[n, m] = \sum_{k=-N}^N g(n)h(m)x[n + \frac{k}{2}]x^*[n - \frac{k}{2}]e^{-2\pi i km/N} \quad (4.9)$$



**Figure 4.7:** Wigner Ville analyses over Original and combined signal with different types of chirp

In terms of computing power, in order to calculate the WVD of a signal with 80 thousand samples, Matlab needs about 10GB of RAM to store the convolution matrix(640 million elements). Besides that, it needs high computational power. For these reasons and due to the signal's periodic characteristic, WVD is calculated for 100  $\mu s$  of the signal where the result is depicted in Figure 4.7. As it is shown in figure 4.7, Wigner Ville is a good tool to detect the different shapes of chirp. But in order to deal with traditional machine learning, some statistical features of the images are extracted, such as mean, variance, minimum, and maximum over all elements in the output of WVD.

Domain	Feature Name	Name Meaning
<b>Wigner-Ville (Time-Frequency)</b> (4 Features)	<i>WV_mean</i>	Mean of Wigner-Ville's matrix
	<i>WV_var</i>	Variance of Wigner-Ville's matrix
	<i>WV_min</i>	Minimum of Wigner-Ville's matrix
	<i>WV_max</i>	Maximum of Wigner-Ville's matrix

**Table 4.4:** Feature Extracted by Wigner-Ville Transform for GNSS Signals

## 4.5 Final Dataset

Totally 42 features are obtained by different techniques for each signal of the simulated dataset. All the outputs export to a CSV file for the further steps.

# Chapter 5

## Methodology and Result

In recent years the GNSS has a significant role not only for positioning but also for many technologies and applications, including aviation. Intentional interference by jammers makes GNSS-based systems unavailable because strong interference may make the GNSS receivers inoperable and lead to loss of quality of the GNSS satellite signals. Consequently, GNSS receivers cannot provide accurate positioning. Therefore, identifying the GNSS interference signal is considered as a high priority task. Many researches have been developed to achieve the automatic detection for various types of interferences in GNSS signals by investigating different ML techniques [33] [31] [35] [20]. This thesis project proposes a method for automatic and accurate detection of chirp signals (known as one of the most common and disruptive interfering signals) based on machine learning algorithms. The considered ML algorithms to get this goal are unsupervised machine learning(k-means clustering) and Convolutional Neural Network (CNN).

The scenario of interfering GNSS signals with different types of chirp signals is explained in chapter 3. Afterward, those signals are analyzed through DSP techniques to derive features representing signal behavior in the time and frequency domain(feature extraction).



**Figure 5.1:** Methodology Diagram

## 5.1 K-means Clustering

In general, selecting the appropriate clustering method mainly depends on some factors such as size and type of datasets and the noise in data. The clustering algorithms could be categorized into Partition-based, Hierarchical-based, Density-based, Grid-based, and Model-based [87]. It has to be noted that no clustering algorithm performs well for all the evaluation criteria and even all the clustering algorithm suffers from stability problem [87].

Some researches were performed on several clustering methods such as Hierarchical or Self-Organizing Map (SOM) clustering algorithms to evaluate their performances and provide a comparison. However, the chosen clustering algorithms for the assessment were based on some factors such as popularity, flexibility, applicability, and handling high dimensionality, and the comparison of their results was made according to the size of the dataset, number of clusters, and type of dataset. The evaluation results indicate that the K-means is better than most other algorithms in terms of performance. Indeed, the K-means algorithm is mostly nominated for a huge amount of data, despite its sensitivity to noise which will affect the outcome of the algorithm [88] [89].

Although clustering high dimensional data is challenging [87], many unsupervised clustering algorithms with specific requirements and abilities exist to detect high dimensional data clusters [90]. There is a fact that with increment the attributes or dimensions in a dataset, measuring the distance will become progressively meaningless. There are several recent approaches to assist in clustering high-dimensional data. These approaches have been successfully performed in many fields. It is not truthful to expect a particular method of clustering approach to be suitable for all types of data or even for all high dimensional data. With increasing dimensionality, some algorithms become computationally complex, making them inconvenient in many real applications [87].

The principal components are the continuous solutions to the discrete cluster membership indicators for K-means clustering. It is possible to implement K-means on PCA results which is one of the feature selection algorithms in this thesis, and in fact, there is a deep connection between K-means clustering with PCA [91].

As a consequence, the K-means clustering is very convenient in exploratory data analysis. Compared to other clustering techniques, the ease of implementation and low computational complexity and memory consumption at the same time with high proficiency make K-means one of the most popular clustering algorithms. The K-means clustering can also be applied as an initialization step for other algorithms

that suffer from complexity in computation, thus approximating the distribution of data is a starting point and reducing the noise present in the whole dataset [92].

### 5.1.1 Data Preparation

Data preparation or data preprocessing is the process of gathering, converting, transforming, cleaning, or other function to make the dataset readable for the machine. One of the important purposes of data preparation is to ensure correction and consistency of the prepared information for analysis since the dataset often includes missing values, inaccuracies, or other errors [93].

The first step is normalization, which means changing the features' scale to have a more informative range. There are different functions to do it, such as Inverse, LogN, Log10, Exponential, Square Root. The fundamental reason for normalization in machine learning is that the features with widespread standard deviation describe data behavior more efficiently. Based on analysis of all features and the shape of data distribution, normalizing with Log10 is applied to the dataset.

Therefore, all the features extracted are numerical, and it is no need for transformation. However, there is the possibility to face some infinity or NaN values that are replaced with Zero. If the number of zero values for each row of data is over five, that row is eliminated from the dataset.

Since K-Means uses distance-based measurements to determine the closeness between data points, the datasets are standardized to have a zero mean and standard deviation of 1 because features in any dataset might have different units of measurements. Also, to prevent misleading training process of K-mean resulting in longer training time, less accurate model, and poor result, outliers are removed.



Selected features are presented in Table 5.1. As it can be seen in Figure 5.2, for instance the correlation between *Time\_kurtosis\_real* and *Time\_kurtosis\_img* is 0.99 and imply that these two features provide the same behavior of the data, as a consequence *Time\_kurtosis\_img* has been removed from the final list of features.

Time_mean_real	Freq_max_real	Freq_mean_ph_cpx
Time_std_real	Freq_min_real	Freq_var_ph_cpx
Time_skewness_real	Freq_mean_img	WL_cA_mean_real
Time_kurtosis_real	Freq_var_img	WL_cD_mean_real
Time_mean_img	Freq_min_img	WL_cD_var_real
Time_median_img	Freq_var_cpx	WL_cA_mean_img
Time_skewness_img	Freq_min_cpx	WL_cD_mean_img

**Table 5.1:** Selected features by Correlation Matrix

### Supervised Feature Selection

The filter method is implemented in the supervised feature selection, which uses statistical techniques to evaluate the relationship(score) between each feature and the target variable (label) (Section 5.1.2). Those scores are used to select the features that have more correlation with output, and eventually, 14 features are selected from 42. Table 5.2 indicates the selected features.

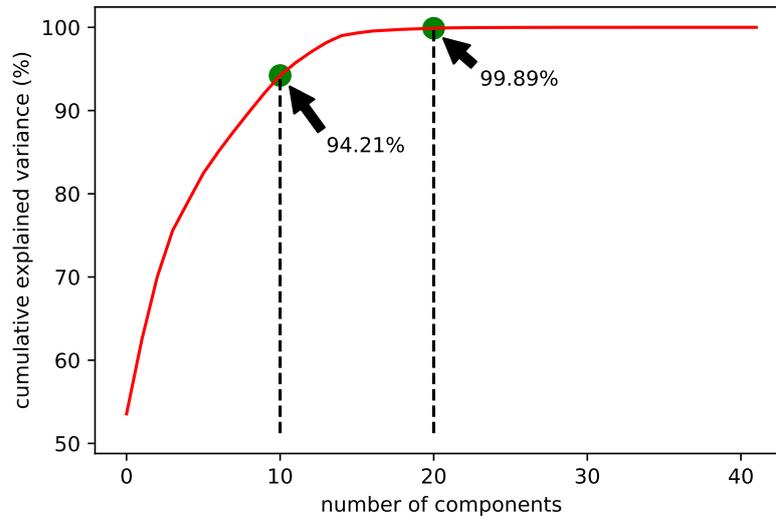
Time_mean_real	Freq_max_real	Freq_min_cpx
Time_median_real	Freq_max_img	Freq_mean_ph_cpx
Time_skewness_real	Freq_mean_cpx	Freq_mean_ph_cpx
Time_mean_img	Freq_var_cpx	WL_cA_mean_real
Time_skewness_img	Freq_max_cpx	

**Table 5.2:** Selected features by Correlation Matrix

### Feature Selection based on PCA

The initial number of features extracted from the input data is 44, and the PCA method is applied to reduce the number of features and decrease the complexity of the model. The processes include calculating the covariance matrix and performing eigendecomposition to map data to a new space representing data by the maximum variance. Then all derived components will get sorted in terms of their variances

from high value to low value. In order to select the number of latent variables, an iterative approach is adopted to calculate cumulative variance with the different number of components that vary from 1 up to 44. Figure 5.3 shows the percentage of data that can be achieved by selecting a particular number of components. For instance, by selecting the first 20 components, 99.8% of data points in the dataset can be covered, while selecting more components does not significantly increase data coverage. Finally, a dataset is created by selecting the first 20 components which have the maximum of variances.



**Figure 5.3:** Cumulative explained variance of different number of components

### 5.1.3 Modelling

Apart from the original dataset with 42 attributes, three different datasets are obtained using different attribute selection techniques. These datasets include:

- Dataset with feature selection based on correlation matrix.
- Dataset with supervised feature selection.
- Dataset based on PCA with 20 components.
- Dataset with all features.

Each dataset comprises 12,800 rows of data. Each row represents an analyzed signal that can be original in the absence of the chirp signal or combined with a

specific chirp signal type (totally eight types) at different power. Consequently, the number of clusters in K-means is nine. The dataset is divided into two parts in order to evaluate the model's performance. The training part consists of 80% of data (10370 rows), and the test set contains 20% of the dataset. K-means algorithm is implemented on the training datasets and starts by randomly selecting a centroid value for each cluster. The algorithm then performs three steps iteratively:

1. Finding the Euclidean distance between each data sample and the centroid of all clusters.
2. Assigning data points with the closest distance to the centroid cluster.
3. Calculating the new centroids on the average values of the coordinates of all data points from the respective cluster.

Note that if each cluster's centroids are determined clearly, it is simple to calculate the distance between new data and the centroids to identify which cluster belongs to the new data points.

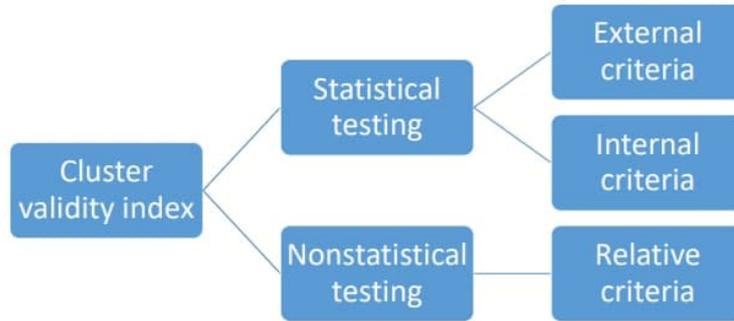
#### **5.1.4 Evaluation and Result**

A research done by Steinbach et al. explains that in supervised learning, the evaluation of the resulting classification model is an integral part of developing a classification model, and there are well-accepted evaluation measures and procedures. In contrast, due to the nature of unsupervised learning, cluster validation does not grow much [94]. It is important to consider several aspects to validate the model when analyzing the clustering results:

1. Determining whether a non-random structure exists.
2. Discovering the exact number of clusters.
3. Evaluating the feature of the clustering results without external information.
4. Analyzing the results achieved with external information.
5. Examining two sets of clusters to determine which one is better.

Since the dataset is generated manually, each signal has a predefined label that is not presented and used in the training phase so that these labels can be used as external information. According to the above categories and the structure of the database, the adopted solution for evaluation of the model lies in the fourth category that can be resolved by external informations or supervised validation methods [94].

Gan et al. [95] introduce a taxonomy of evaluation methods covering internal and external validation approaches. External accuracy validation extends by combining additional information into the clustering validation process, for instance, external class labels for training examples.



**Figure 5.4:** Taxonomy of Evaluation Methods Covering Internal and External Validation Approaches

"Matching Sets" is one of the methods that can be utilized in the k-means clustering technique in order to evaluate the accuracy of the model [96]. The steps of this method is as follows:

- The model can be trained on the first portion of data without considering external information (labels).
- A small subset (20%) of labeled data points in the training dataset is used. These labeled data are injected into the clustering model, and based on the label and number of points that each cluster received, a class label is assigned to each cluster.
- The accuracy can be evaluated by using the test dataset in the presence of labels as ground truth.

Formula 5.1 evaluates the percentage of elements that are properly included in the same cluster.

$$Pr = \frac{TP}{TP + FP} \tag{5.1}$$

where  $TP$  represents True positive, how many data points are correctly classified

in the same cluster. *FP* indicates False positive, how many examples are classified incorrectly. The model’s accuracy with different datasets is shown in Table 5.3.

	Accuracy	No. of Features
Feature Selection based on Correlation Matrix	37.28%	21
Supervised Feature Selection	35.06%	14
Feature Selection based on PCA	41.03%	20
All Features	27.39%	42

**Table 5.3:** Accuracy of Data Injection using K-Mean Algorithm

Remarks on K-means results:

- The obtained accuracy of the clustering models implies that although the development of the K-means clustering algorithm had a considerable role as an initial step in the analysis, it could not satisfy the desired objective completely.
- Conforming to section 5.1.2, feature selection has been used to remove redundant features from the dataset and improve the model performance. Furthermore, it is claimed that removing unneeded data(non-informative) leads to better decision-making due to less noise dependency. According to all aforementioned, the accuracy results of K-means (Table 5.3) indicate that the lower accuracy is related to the dataset considering all the features, which proves obtained assertion.
- According to our study achievements represented in section 4.1, PCA as a dimensionality reduction method is well-matched (has a deep connection) with the K-means clustering algorithm. On the other hand, referring to our results of applying K-means clustering on the attained dataset of several feature selection techniques (Table 5.3), it is evident that among different feature selection approaches, the PCA technique achieved a better result than others regarding accuracy.
- Due to the content of Section 5.1.2 along with observing the final result (Table 5.3), it is determined that although supervised learning as a feature selection reduces the data dimensionality and complexity significantly in contrast to the correlation matrix method, it provides less accuracy due to considering a small subset of important features. However, the small difference in the obtained value of accuracy in applying the K-means algorithm for both mentioned techniques states the proximity of their performances in our model.
- A more detailed evaluation of both techniques, feature selection based on correlation matrix, and supervised feature selection according to the tables’

results (Tables 5.1 and 5.2) specify Wigner-Ville analysis is not utilized in making their models, which implies that the selected statistical parameters of this transformation are not meaningful and well distributed. Whereas the images of WV analysis (Figure 4.7) illustrate a good view of the chirp signal shapes, the extraction of selected statistical parameters from those images will cause the loss of information. Thus, as the next step, Convolutional Neural Network(CNN) is introduced as the second approach to work directly with WV analysis images and improve the gained results.

## 5.2 Convolutional Neural Network

There are many techniques to perform image classifications. Some of the most common methods are studied to find the best-matched technique according to the desired target, which state as follows:

- A convolutional neural network (CNN) is a multilayer neural network proposed to identify visual patterns from images represented by pixels with the least preprocessing [97] [98].
- Transfer Learning (TF) as the second algorithm, first trained according to a problem and then uses the acquired knowledge during the training process to solve a different related problem [99].
- As the next machine learning technique, the Support Vector Machine (SVM) could generate a flexible and powerful model to solve ML problems. SVM represents the various classes in a hyperplane in multidimensional space. As an important component of this technique, the Kernel function has a strong impact on the performance of the model [100].
- The next popular algorithm is K-Nearest Neighbor (K-NN), which is mostly used for classification and regression problems. This method only relies on the distance among the feature vectors and classifies unknown data points by obtaining the most common class between the k-nearest instances [101] [102].

Table 5.4 represents the advantages and disadvantages of the aforementioned methods. A CNN-based method is considered to achieve a more accurate result in image classification by referring to a study on different introduced techniques [103] [104]. This network's input data is a set of images containing different shapes of interference chirp signals, and the output represents nine different values indicating distinct classes of chirp types. In the following sections, the model structure and the results are described.

Method	Advantage	Disadvantage
CNN	<ul style="list-style-type: none"> <li>• The most accurate and reliable method for image classification</li> <li>• Pre-trained weights can be shared among different methods</li> <li>• Simplification of computation without losing important data</li> <li>• Does not require feature extraction</li> </ul>	<ul style="list-style-type: none"> <li>• In order to obtain high accuracy, a big number of samples are needed</li> <li>• The training process is slow</li> </ul>
TL	<ul style="list-style-type: none"> <li>• Does not need huge dataset to do training</li> <li>• Reduce the training time</li> </ul>	<ul style="list-style-type: none"> <li>• It is no flexible in term of reducing parameters</li> <li>• There is the probability of negative transfer as it works based on similarity of training and test set</li> </ul>
SVM	<ul style="list-style-type: none"> <li>• Can classify continues and categorical data</li> <li>• Can be used for linear and non-linear problems</li> <li>• The accuracy remains high even with noisy images</li> <li>• Handling overfitting is not a hard task</li> </ul>	<ul style="list-style-type: none"> <li>• The training is very time consuming</li> <li>• Due to using complicated kernels, understanding the algorithm is hard</li> <li>• SVM are applied on binary classification</li> </ul>
K-NN	<ul style="list-style-type: none"> <li>• The easiest algorithm to implement</li> <li>• For small datasets there is no need to do training</li> </ul>	<ul style="list-style-type: none"> <li>• Uses a large amount of memory to keep training data</li> <li>• The training process is slow</li> </ul>

**Table 5.4:** Advantages and Disadvantages of Various Techniques for Image Classification [104]

### 5.2.1 Images Dataset

Different types of chirp signals described in Section 3.2.1 affect the GNSS signals in different ways. As the chirp signals are non-stationary and their frequency varies with respect to time, Wigner-Ville analysis can be a possible transform for time-frequency modeling of the signal. As explained in Section 3.3, different types of chirp signals totally (8 cases) are combined with GNSS signals at strong, medium, and weak power, but since the weak power of the chirp signal is under the noise floor and Wigner-Ville analysis does not recognize its components, weak power of

the chirp signal are not considered in the simulation scenario.

Implementing Wigner-Ville analysis is described in Section 4.4; however, instead of extracting an image’s statistical parameters as features, the spectrogram’s image of the signals is stored in the dataset. All images are produced in grayscale and rescaled from 1784x892 (280 Kb) to 446x223 (14 Kb) to reduce the size of images. Figure 5.5 represents one sample from each class in the dataset.

As explained in Section 3.1, signals of 32 different satellites are generated, where 25 of them have been used as the training dataset, And the other seven signals were used for testing the algorithm. In the end, in each class of input, there exist almost 350 images. Finally, this image dataset is prepared to feed the CNN algorithm as input.

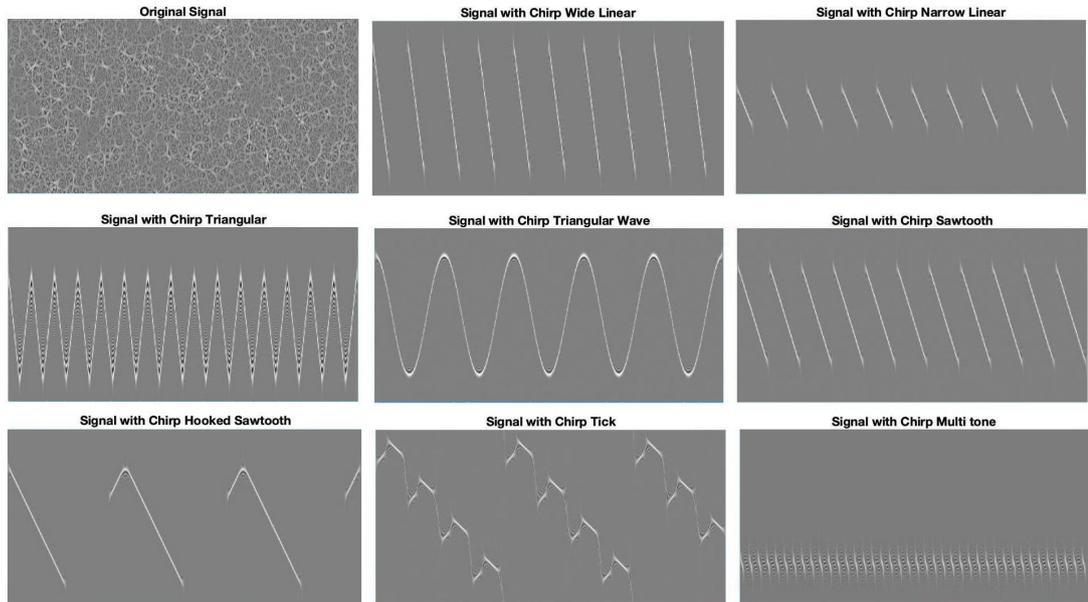


Figure 5.5: Sample of images in dataset

## 5.2.2 Model Structure

There are many models to do the image classification including, AlexNet, GoogleNet, SqueezeNet. The performance and properties of these three networks are evaluated in terms of speed for training samples and the accuracy of the classification model. The obtained results for the same datasets representing that although

the accuracy is almost the same for all three of them, AlexNet is faster as compared to the other introduced pre-trained architectures in respect of the speed of the training process. The reason is that AlexNet has eight layers deep with respect to SqueeZNet and GoogleNet, which have 18 and 22 layers deep, respectively.

Alex Krizhevsky primarily designed AlexNet CNN. The input to AlexNet can be an RGB image of size  $227 \times 227$  pixel that implies all images in the train and test set need to be of size  $227 \times 227$  pixel. All the images used in this experience are resized and crop to be in the desired size. In addition, by using a filter, all the input images changed to grayscale mode. The details of AlexNet architecture are described as follows:

- AlexNet comprises 5 convolutional layers and 3 fully connected layers, 3 max-pooling layers, 2 normalization layers, and 1 softmax layer.
- ReLU nonlinearity is applied after all the convolution and fully connected layers. The ReLU nonlinearity of the first and second convolutional layers are followed by a local normalization step before doing pooling.
- The first two convolutional layers are followed by the overlapping max pooling layers.
- The third, fourth and fifth convolutional layers are connected directly and The fifth convolutional layer is followed by an Overlapping Max Pooling layer.
- the output of fifth convolutional layer after pooling goes into a series of two fully connected layers. After each of these two layers, there are dropout layers to avoid overfitting.
- The output of the second fully connected layer goes directly to the third fully connected layer and then it feeds into a softmax classifier with 9 class labels.
- AlexNet has totally about 60 millions of parameters.

The entire architecture of AlexNet is depicted in Figure 4.4. Using the softmax layer in this model provides some level of reliability and probability in the output layer that the data belongs to a specific class instead of a rough answer.

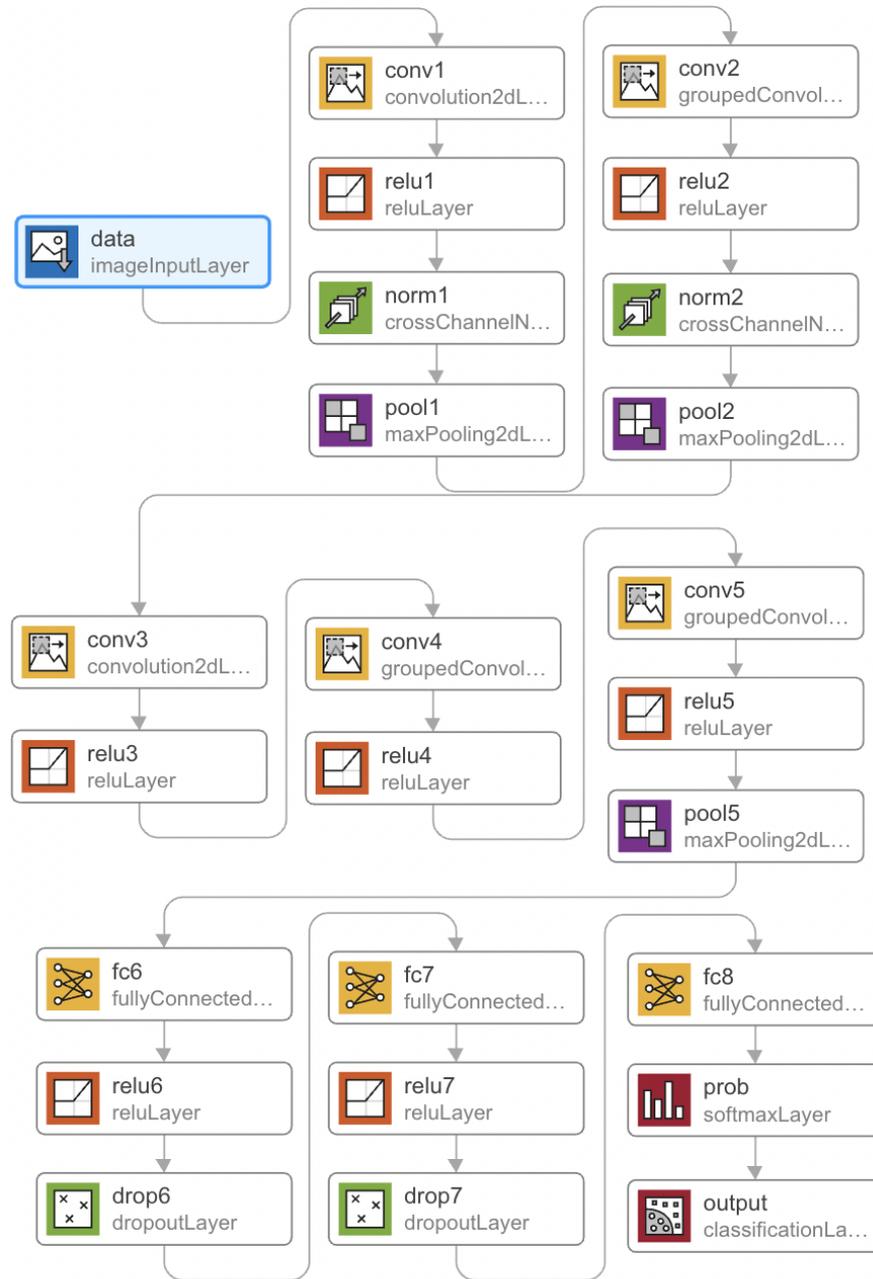


Figure 5.6: AlexNet Architecture

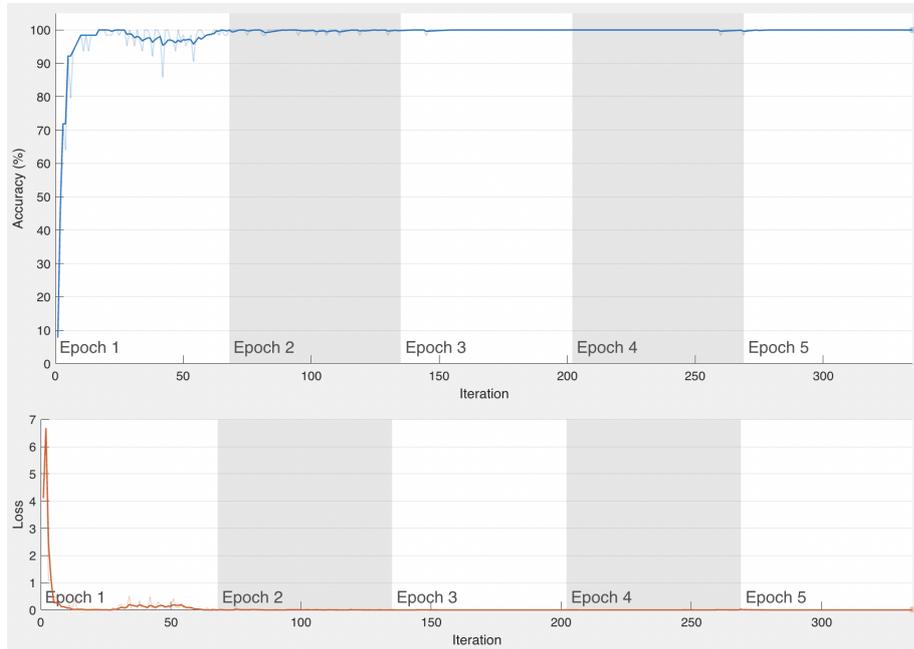
### 5.2.3 Configuration and Result

The initial configuration of the CNN is described in table 5.5.

Parameter	Value
Minimization Algorithm	Stochastic Gradient Descent Momentum
Initial Learning Rate ( $\lambda$ )	0.001
Batch size	64
Input Image Size	227 x 227
Weight Learn Rate Factor	20
Bias Learn Rate Factor	20
Maximum Number of Epoch	5

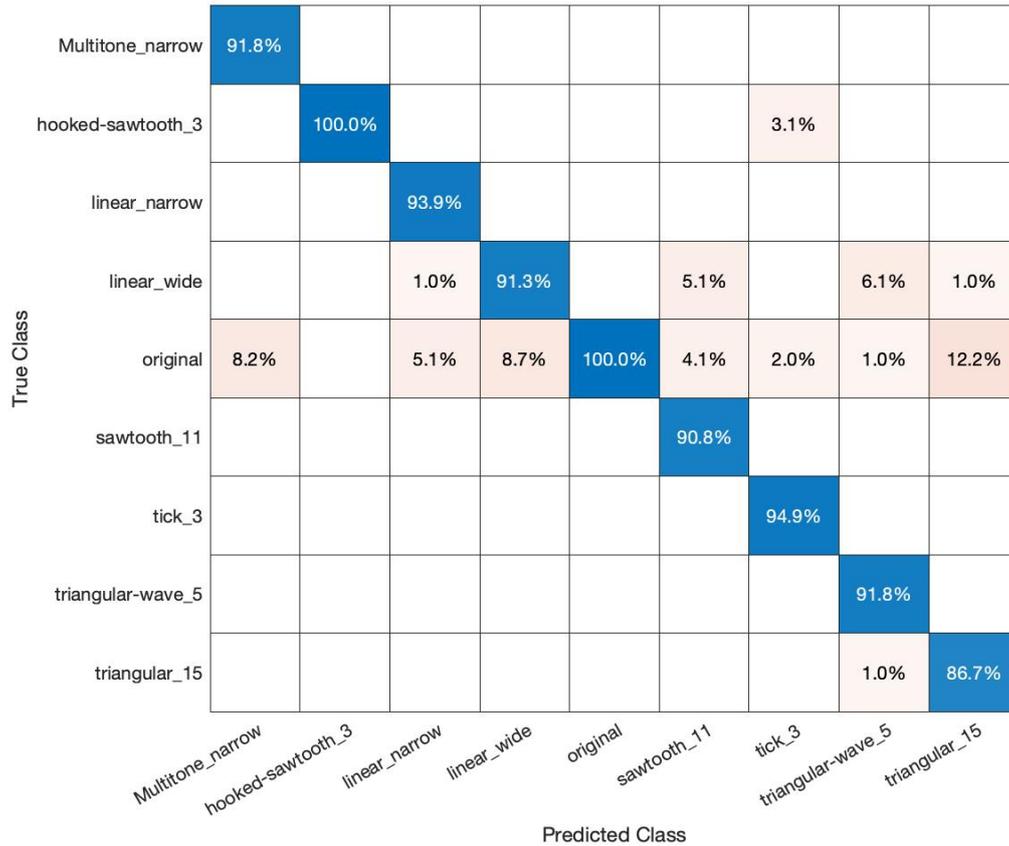
**Table 5.5:** Initial Configuration of AlexNet

The CNN is trained during five epochs using the SGDM algorithm, which means the all images in training dataset is passed through the neural network five times both in forward and back-propagation to find the optimal values of the network's parameters. Figures 5.7 shows the process of training using two metrics, training loss and training accuracy. As it is depicted, the accuracy of CNN is reached up to 99.1% in the training dataset.



**Figure 5.7:** Accuracy and Lost in the Training Process of AlexNet

The last step is to assess the trained network, and the evaluation phase provides a performance metric of the trained model on unobserved data. For the evaluation phase of the model, test data is utilized in order to create the confusion matrix. The confusion matrix determines the accuracy of a classification model in the way of how well it predicts the correct and incorrect class for the test data. Figure 5.8 illustrates the confusion matrix of the test dataset.



**Figure 5.8:** Confusion Matrix of CNN Algorithm

Figure 5.8 represents that the CNN method for classification has an overall accuracy of 93.46%. The CNN is also able to predict the existence and detect shape of chirp signal in GNSS signal with an accuracy of 92.65%.

Remarks on CNN result:

- As reported in section 5.2.1, the CNN technique is applied on the grayscale images, which includes the spectrum of the signal, as an input of the algorithm

to apply the classification of chirp signals. As expected the obtained result is faced higher accuracy in contrast to statistical analysis.

- Due to the fact that there are many convolutional layers, including lots of parameters such as weights and biases, therefore the increment of computational time and complexity in the analysis of data will be expected. As the next point, the CNN technique has a high accuracy in image classification which is our target.
- The main privilege of using CNN over the statistical approach is feature selection done by convolutional layers. Due to the redundant attributes and a large amount of input data in original data sets, feature selection is an important technique for improving neural network performance which uses several convolutional layers to detect the most significant features.

# Chapter 6

## Conclusion

GNSS signals are really at risk of being interfered with by various intentional and unintentional interferences, which will cause degrading accuracy or complete lack of positioning. Hence, in order to acquire high accuracy in positioning, it is critical to detect and recognize different types of interference in GNSS signals.

This dissertation ended a full-field analysis method, which focuses on chirp jammers detection based on the two different machine-learning algorithms. The selected machine learning algorithms to perform classification on datasets were based on literature studies and other research results.

Firstly, the K-means clustering technique belongs to the unsupervised machine learning types. Since it is an unsupervised algorithm, it can overcome one of the significant limitation aspects of supervised techniques: the necessity of labeled datasets representing interference and interference-free conditions. Our achievements related to the study on four different datasets (section 5.1.3) indicate that the research models based on K-means failed to produce interpretable solutions. Although K-means has simplicity in implementation (referring to section 5.1), it yields poor outcomes in terms of performance accuracy. In fact, the obtained result of this approach states that the final output has a high dependency on the chosen statistical parameters and the number of features.

As the second implemented algorithm, we developed a methodology in order to identify and classify the chirp interference based on the analysis of the spectrum of the signal. Hence, in this scenario, the datasets included spectrum images. Based on the different studies, the CNN method reportedly has a good performance in image classification problems. This seems to be matched with our result of applying this algorithm to the dataset.

It has to be noted that two distinct algorithms based on two separate approaches and datasets are investigated to achieve the desired purpose (presence of chirp and classify it). A brief consequence obtained from a study on their implementation and results demonstrates although it does not make sense to compare two different approaches, it is possible to achieve a better outcome with respect to choosing a more proper algorithm. In fact, the main conclusion represents that regardless of the specific capability of different ML algorithms, there is no preferred detection method. Detection reliability depends on many factors, mainly the characteristics and number of observed data in the dataset and the desired objective.

# Appendix A

## Chirp Shapes

Many different types of signals are characterized as a chirp. The following sections describe the definitions of the different categories used in the thesis.[68]

### A.1 Wide sweep

<p>Spectrum plot:</p> <ul style="list-style-type: none"><li>• show wide variation in power levels at all frequencies</li><li>• Often see shape of reference spectrum defining bottom edge of power levels</li></ul>	<p>Spectrogram:</p> <ul style="list-style-type: none"><li>• Clearly defined and separated linear (or slightly curved) diagonal lines across wide frequency range</li><li>• Most commonly show frequency increasing with time</li></ul>
---	--

Table A.1

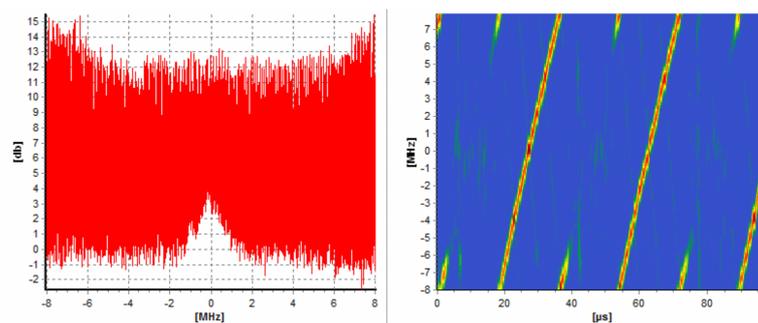


Figure A.1: Wide sweep - Slow(2 to 3 chirps per 100 microseconds) [68]

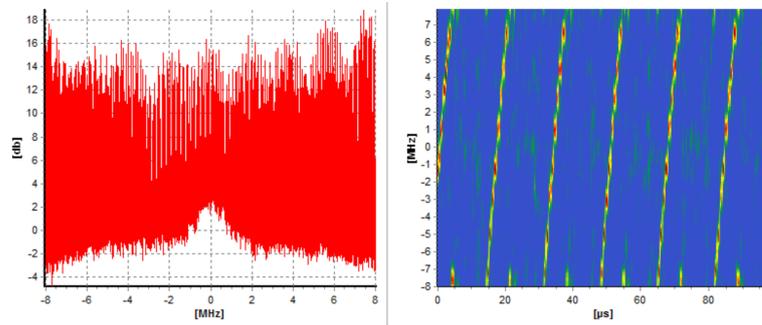


Figure A.2: Wide sweep - Medium(4 to 6 chirps per 100 microseconds) [68]

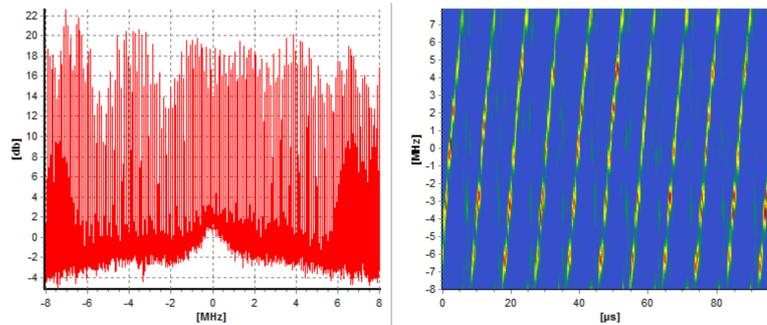


Figure A.3: Wide sweep - Fast(8 to 12 chirps per 100 microseconds) [68]

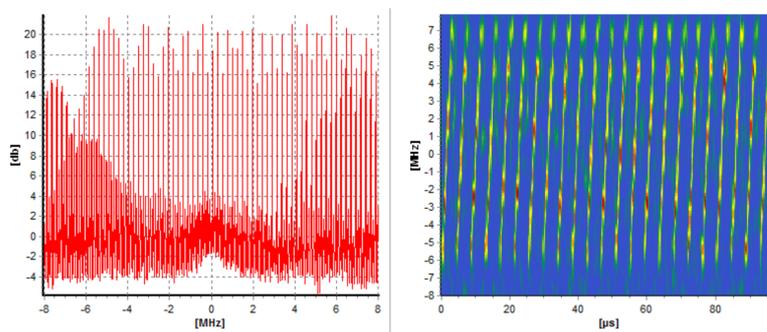


Figure A.4: Wide sweep - Rapid(more than 12 chirps per 100 microseconds) [68]

## A.2 Narrow sweep

<p>Spectrum plot:</p> <ul style="list-style-type: none"><li>• show increase in power levels across narrow frequency range</li></ul>	<p>Spectrogram:</p> <ul style="list-style-type: none"><li>• Clearly defined and separated linear (or slightly curved) diagonal lines covering small frequency range</li><li>• Most commonly show frequency increasing with time</li></ul>
---	---

Table A.2

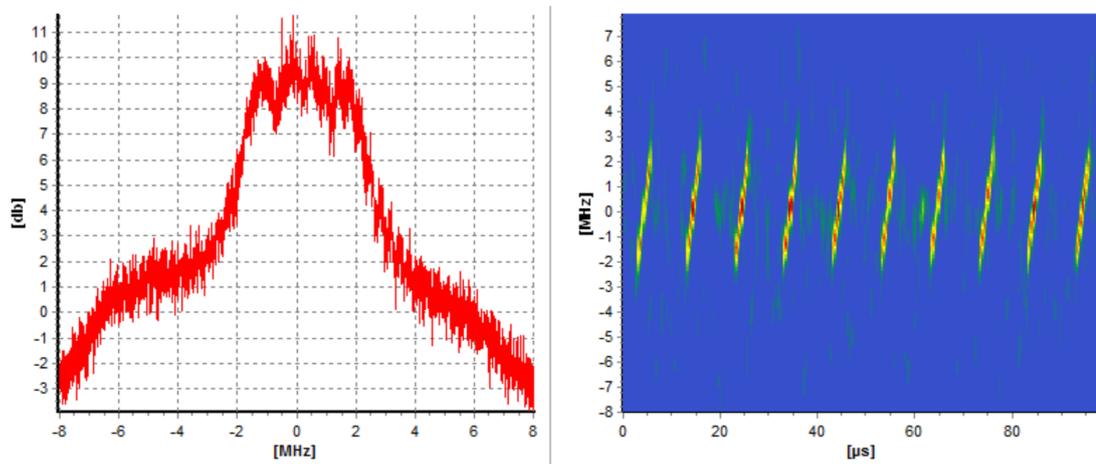


Figure A.5: Narrow sweep [68]

## A.3 Triangular

<p>Spectrum plot:</p> <ul style="list-style-type: none"><li>• more likely to see raised power over affected frequency range</li></ul>	<p>Spectrogram:</p> <ul style="list-style-type: none"><li>• Clearly see decrease and increase in frequency with time</li><li>• Gradient and power level of downward and upward slopes are more equal than in sawtooth case</li></ul>
---	--

Table A.3

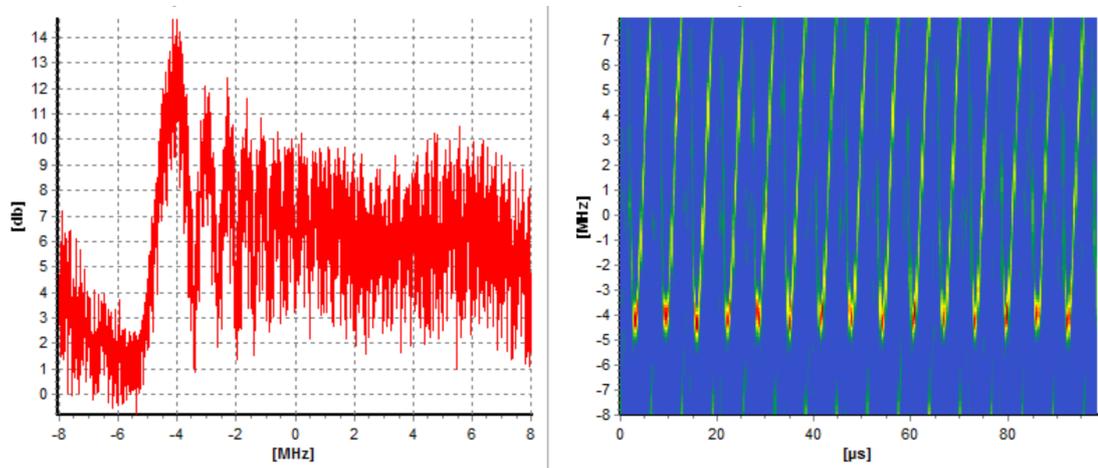


Figure A.6: Triangular [68]

## A.4 Triangular wave

Spectrum plot: <ul style="list-style-type: none"><li>• wide range of powers over affected frequency range</li></ul>	Spectrogram: <ul style="list-style-type: none"><li>• Wave pattern showing clear continuous increase and decrease in frequency with time</li></ul>
---	---

Table A.4

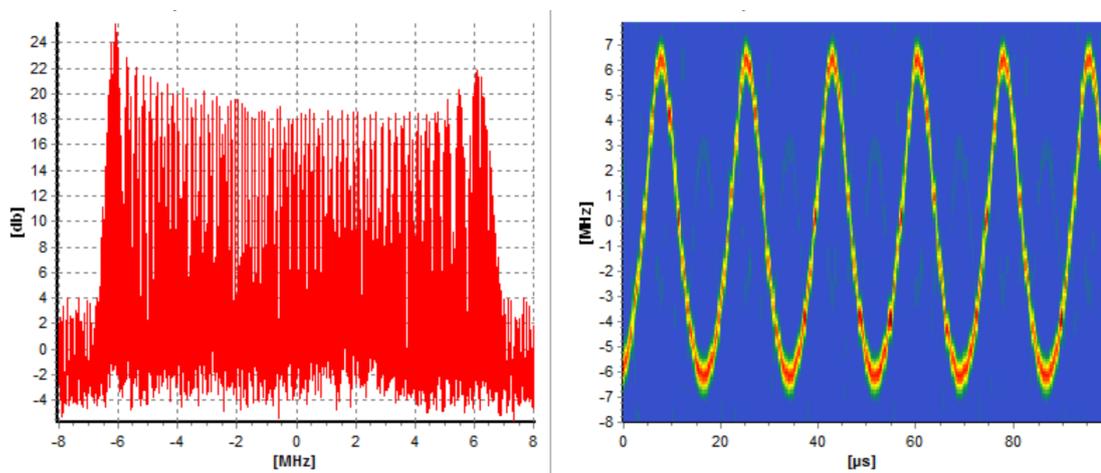


Figure A.7: Triangular Wave [68]

## A.5 Sawtooth

<p>Spectrum plot:</p> <ul style="list-style-type: none"><li>• Raised power over affected frequency range</li></ul>	<p>Spectrogram:</p> <ul style="list-style-type: none"><li>• Linear sweeps in frequency across wide range</li><li>• See decrease in frequency with time as well as the increase</li><li>• Gradient of downward slope is much sharper than main upward slope, and less well defined</li></ul>
--	---

Table A.5

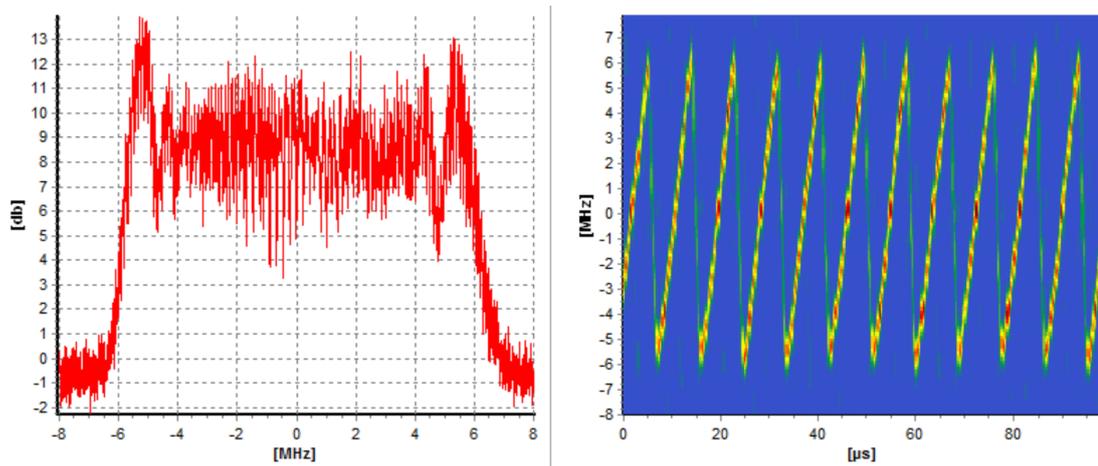


Figure A.8: Sawtooth [68]

## A.6 Hooked sawtooth

<p>Spectrum plot:</p> <ul style="list-style-type: none"><li>• Raised power over affected frequency range</li></ul>	<p>Spectrogram:</p> <ul style="list-style-type: none"><li>• Linear sweeps in frequency across wide range</li><li>• See decrease in frequency with time as well as the increase</li><li>• Gradient of downward slope is much sharper than main upward slope, and less well defined</li></ul>
--	---

Table A.6

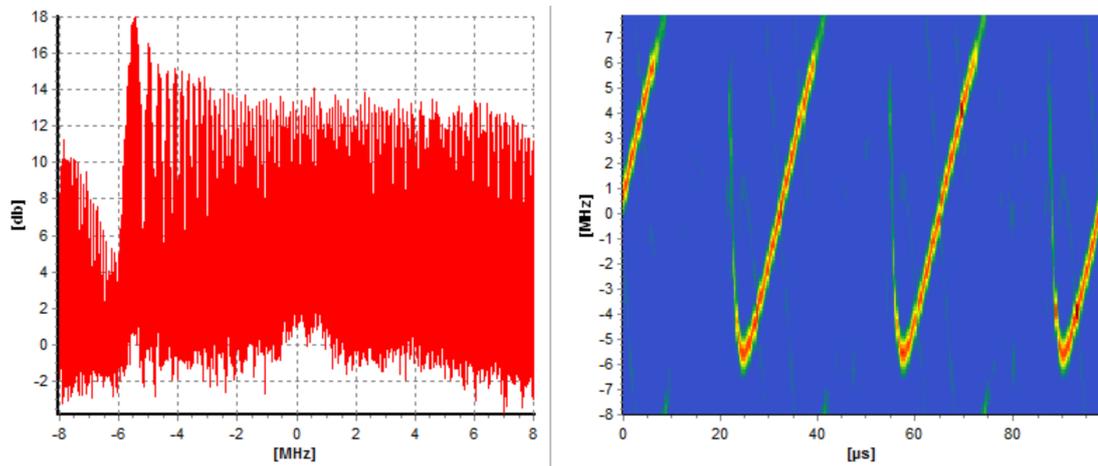


Figure A.9: Hooked Sawtooth [68]

## A.7 Tick

<p>Spectrum plot:</p> <ul style="list-style-type: none"><li>• general increased power across the spectrum</li></ul>	<p>Spectrogram:</p> <ul style="list-style-type: none"><li>• underlying slow wide sweep (2-3 sweeps per 100 microseconds)</li><li>• Additional structure and variation (taking form of a tick) overlaying the underlying slow sweep</li></ul>
---	--

Table A.7

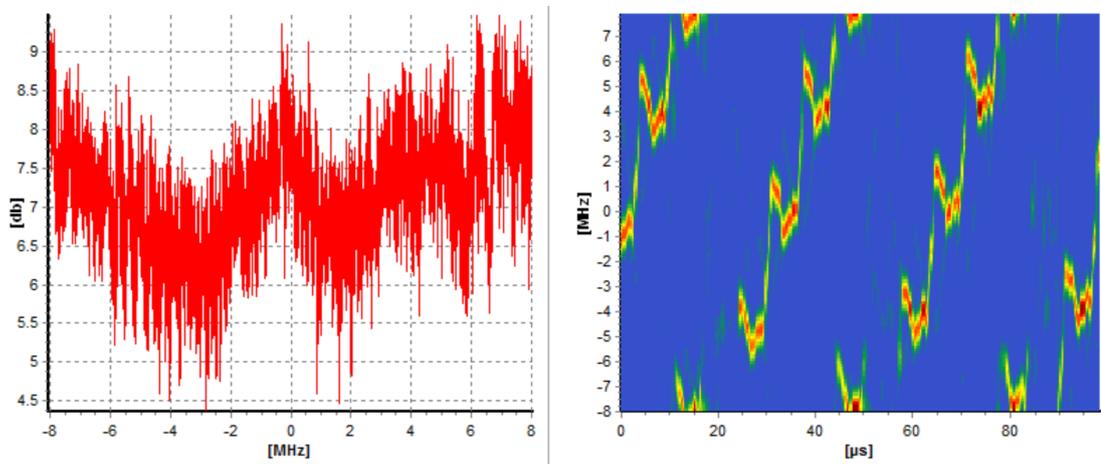


Figure A.10: Tick [68]

## A.8 Multi tone

Spectrum plot: <ul style="list-style-type: none"><li>• Multiple distinct tones with high power at different frequencies</li></ul>	Spectrogram: <ul style="list-style-type: none"><li>• multiple closely spaced near vertical lines in the region of affected frequency</li></ul>
---	--

Table A.8

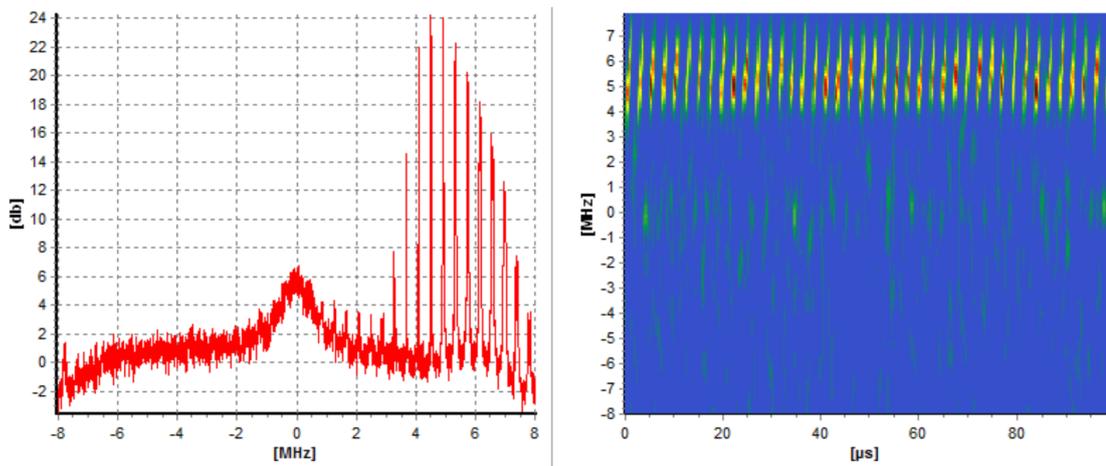


Figure A.11: Multi Tone [68]

# Bibliography

- [1] Christopher J. Hegarty Elliot D. Kaplan. *Understanding GPS: principles and applications*. 2006 (cit. on pp. 1, 8).
- [2] François Peyret, Pierre-Yves Gilliéron, Laura Ruotsalainen, and Jesper EN-GDAHL. *SaPPART White paper : Better use of Global Navigation Satellite Systems for safer and greener transport*. 2015 (cit. on p. 1).
- [3] Borrás Jorge Querol. *Radio Frequency Interference Detection and Mitigation Techniques for Navigation and Earth Observation*. 2018 (cit. on pp. 2, 6).
- [4] *L frequency Band*. European Space Agency. URL: [https://gssc.esa.int/navipedia/index.php/GNSS\\_signal](https://gssc.esa.int/navipedia/index.php/GNSS_signal) (cit. on pp. 3, 4).
- [5] *Wave interference definition*. Physics and Radio-Electronics. URL: <http://www.physics-and-radio-electronics.com/physics/waveinterference.html> (cit. on p. 4).
- [6] International Telecommunication Union. *Radio Regulations Articles*. 2012 (cit. on p. 4).
- [7] Fabio Dovis. *GNSS Interference threats and countermeasures*. 2015 (cit. on pp. 5–12).
- [8] Jun Yang Zhi Qu and Jianyun Chen. «Continuous Wave Interference Effects on Ranging Performance of Spread Spectrum Receivers». In: (Dec. 2014) (cit. on p. 5).
- [9] Wenjian Qin, Micaela Troglia Gamba, Emanuela Falletti, and Fabio Dovis. «Effects of Optimized Mitigation Techniques for Swept-frequency Jammers on Tracking Loops». In: Oct. 2019, pp. 3275–3284. DOI: 10.33012/2019.17067 (cit. on pp. 5, 32).
- [10] Kung Chie Yeh and Chao-Han Liu. «Radio wave scintillations in the ionosphere». In: *Proceedings of the IEEE* 70.4 (1982), pp. 324–360. DOI: 10.1109/PROC.1982.12313 (cit. on p. 6).
- [11] *Radio Frequency Bands*. tera hertz imaging. URL: <https://terasense.com/terahertz-technology/radio-frequency-bands/> (cit. on p. 7).

- [12] B.M. Titus. «Intersystem and Intrasystem Interference Analysis Methodology». In: (Sept. 2003) (cit. on p. 7).
- [13] James T. Curran, Michele Bavaro, Pau Closas, and M. Navarro. «On the Threat of Systematic Jamming of GNSS». In: Sept. 2016. DOI: 10.33012/2016.14672 (cit. on p. 7).
- [14] Daniel Marnach. «Detecting Meaconing Attacks by Analysing the Clock Bias of Gnss Receivers». In: (Jan. 2013) (cit. on p. 8).
- [15] Daniele Borio, Fabio Dovis, Heidi Kuusniemi, and Letizia Lo Presti. «Impact and Detection of GNSS Jammers on Consumer Grade Satellite Navigation Receivers». In: *Proceedings of the IEEE* (May 2016), pp. 1–13. DOI: 10.1109/JPROC.2016.2543266 (cit. on p. 9).
- [16] Steven P. Powell Ryan H. Mitch Mark L. Psiaki and Jahshan A. Bhatti. «Civilian GPS Jammer Signal Tracking and Geolocation». In: (2012) (cit. on pp. 9, 12).
- [17] M. Tamazin, M. Karaim, H. Elghamrawy, and A. Noureldin. «A Comprehensive Study of the Effects of Linear Chirp Jamming on GNSS Receivers under High-Dynamic Scenarios». In: (2018), pp. 9–14. DOI: 10.1109/ICCES.2018.8639399 (cit. on p. 9).
- [18] Matej Bažec, Franc Dimc, and Polona Prešeren. «Evaluating the Vulnerability of Several Geodetic GNSS Receivers under Chirp Signal L1/E1 Jamming». In: *Sensors* 20 (Feb. 2020), p. 814. DOI: 10.3390/s20030814 (cit. on p. 9).
- [19] Jorge Querol Borrás, Adrian Perez, Joan Francesc Munoz-Martin, and Adriano Camps. «Real-time Radio Frequency Interference Detection and Mitigation with the Front-End GNSS Interference eXcisor (FENIX)». In: (2019) (cit. on p. 10).
- [20] F. Dovis, R. Imam, W. Qin, C. Savas, and H. Visser. «Opportunistic use of GNSS Signals to Characterize the Environment by Means of Machine Learning Based Processing». In: (2020), pp. 9190–9194. DOI: 10.1109/ICASSP40776.2020.9052924 (cit. on pp. 10, 13, 48).
- [21] Li H. Xu J Ying S. «GPS Interference Signal Recognition Based on Machine Learning». eng. In: (2020) (cit. on pp. 10, 12).
- [22] Andrzej Felski and Marta Gortad. «The significance of an antenna for jamming resistance of a GPS receiver». In: (June 2016) (cit. on p. 10).
- [23] Beatrice Motella and Letizia Lo Presti. «Methods of Goodness of Fit for GNSS Interference Detection». In: (July 2014) (cit. on p. 10).
- [24] H. Borowski, O. Isoz, F.M. Eklof, Suhardy lo, and Dennis Akos. «Detecting false signals: With automatic gain control». In: 23 (Apr. 2012), pp. 38–43 (cit. on p. 11).

- [25] Frédéric Bastide, Dennis Akos, Christophe Macabiau, and Benoit Roturier. «Automatic Gain Control (AGC) as an Interference Assessment Tool». In: (Sept. 2003) (cit. on p. 11).
- [26] Lukas Marti and Frank van Graas. «Interference Detection by Means of the Software Defined Radio». In: (Sept. 2004) (cit. on p. 11).
- [27] Kewen Sun and Dongkai Yang. «An Improved Time-Frequency Analysis Method in Interference Detection for GNSS Receivers». In: *Sensors (Basel, Switzerland)* 15 (Apr. 2015), pp. 9404–26. DOI: 10.3390/s150409404 (cit. on p. 11).
- [28] E. Falletti, M. Pini, and L. L. Presti. «Low Complexity Carrier-to-Noise Ratio Estimators for GNSS Digital Receivers». In: *IEEE Transactions on Aerospace and Electronic Systems* (2011), pp. 420–437 (cit. on p. 11).
- [29] E. Chatre Bastide F. and C. Macabiau. «GPS Interference Detection and Identification Using Multicorrelator Receivers». In: (Sept. 2001) (cit. on p. 12).
- [30] D. Manolakis and S.kogon. «Statistical and Adaptive Signal Processing». In: (2005) (cit. on p. 12).
- [31] N. Linty, A. Farasin, A. Favenza, and F. Dosis. «Detection of GNSS Ionospheric Scintillations Based on Machine Learning Decision Tree». In: *IEEE Transactions on Aerospace and Electronic Systems* 55.1 (2019), pp. 303–317. DOI: 10.1109/TAES.2018.2850385 (cit. on pp. 12–15, 48).
- [32] Peter Flach. *Machine Learning: The Art and Science of Algorithms That Make Sense of Data*. Jan. 2012. ISBN: 9780511973000. DOI: 10.1017/CB09780511973000 (cit. on pp. 12, 14).
- [33] Olorato Mosiane. «Radio Frequency Interference Detection using Machine Learning.» In: (2017) (cit. on pp. 12, 48).
- [34] Samuel Foster Kevin Vinsen and Richard Dodson. «Using Machine Learning for the detection of Radio Frequency Interference». In: (Mar. 2019) (cit. on p. 12).
- [35] Alberto de la Fuente Ruben Morales Ferre and Elena Simona Lohan1. «Jammer Classification in GNSS Bands Via Machine Learning Algorithms». In: (Nov. 2019) (cit. on pp. 13, 48).
- [36] Magli Enrico. «Statistical learning and neural networks». In: (cit. on pp. 14, 28, 29).
- [37] Richard Berk. «Classification and Regression Trees (CART)». In: June 2020, pp. 157–211. ISBN: 978-3-030-40188-7. DOI: 10.1007/978-3-030-40189-4\_3 (cit. on p. 16).

- [38] Mohamed Alloghani, Dhiya Al-Jumeily Obe, Jamila Mustafina, Abir Hussain, and Ahmed Aljaaf. «A Systematic Review on Supervised and Unsupervised Machine Learning Algorithms for Data Science». In: Jan. 2020, pp. 3–21. ISBN: 978-3-030-22474-5. DOI: 10.1007/978-3-030-22475-2\_1 (cit. on p. 16).
- [39] S. Sinharay. «An Overview of Statistics in Education». In: *International Encyclopedia of Education (Third Edition)*. Ed. by Penelope Peterson, Eva Baker, and Barry McGaw. Third Edition. Oxford: Elsevier, 2010, pp. 1–11. ISBN: 978-0-08-044894-7 (cit. on p. 16).
- [40] David MacKay. «An Example Inference Task: Clustering». In: *Information Theory, Inference, and Learning Algorithms*. Cambridge University: Hardback, 2003, pp. 284–292. ISBN: 978-0-521-64298-9 (cit. on p. 17).
- [41] Peter J. Rousseeuw. «Silhouettes: A graphical aid to the interpretation and validation of cluster analysis». In: *Journal of Computational and Applied Mathematics* 20 (1987), pp. 53–65. ISSN: 0377-0427. DOI: [https://doi.org/10.1016/0377-0427\(87\)90125-7](https://doi.org/10.1016/0377-0427(87)90125-7) (cit. on p. 17).
- [42] Isabelle Guyon and André Elisseeff. «An Introduction of Variable and Feature Selection». In: *J. Machine Learning Research Special Issue on Variable and Feature Selection* 3 (Jan. 2003), pp. 1157–1182. DOI: 10.1162/153244303322753616 (cit. on p. 17).
- [43] Max Kuhn and Kjell Johnson. «An Introduction to Feature Selection». In: *Applied Predictive Modeling*. Jan. 2013, pp. 487–518. ISBN: 978-1-4614-6848-6. DOI: 10.1007/978-1-4614-6849-3\_3 (cit. on pp. 17, 18).
- [44] Juha Reunanen. «Overfitting in Making Comparisons between Variable Selection Methods». In: 3.null (Mar. 2003), pp. 1371–1382. ISSN: 1532-4435 (cit. on p. 18).
- [45] Xue Ying. «An Overview of Overfitting and its Solutions». In: *Journal of Physics: Conference Series* 1168 (Feb. 2019), p. 022022. DOI: 10.1088/1742-6596/1168/2/022022 (cit. on p. 18).
- [46] Mark Hall. «Correlation-Based Feature Selection for Machine Learning». In: *Department of Computer Science* 19 (June 2000) (cit. on p. 18).
- [47] Jules J. Berman. «11 - Indispensable Tips for Fast and Simple Big Data Analysis». In: *Principles and Practice of Big Data (Second Edition)*. Ed. by Jules J. Berman. Second Edition. Academic Press, 2018, pp. 231–257. ISBN: 978-0-12-815609-4. DOI: <https://doi.org/10.1016/B978-0-12-815609-4.00011-X> (cit. on pp. 18, 19).

- [48] Tom Howley, Michael G. Madden, Marie-Louise O’Connell, and Alan G. Ryder. «The Effect of Principal Component Analysis on Machine Learning Accuracy with High Dimensional Spectral Data». In: *Applications and Innovations in Intelligent Systems XIII*. Ed. by Ann Macintosh, Richard Ellis, and Tony Allen. London: Springer London, 2006, pp. 209–222. ISBN: 978-1-84628-224-9 (cit. on p. 19).
- [49] Johannes Forkman, Julie Josse, and Hans-Peter Piepho. «Hypothesis Tests for Principal Component Analysis When Variables are Standardized». In: *Journal of Agricultural, Biological and Environmental Statistics* 24 (Feb. 2019). DOI: 10.1007/s13253-019-00355-5 (cit. on p. 20).
- [50] Samuel Huang. «Supervised feature selection: A tutorial». In: *Artificial Intelligence Research* 4 (Mar. 2015). DOI: 10.5430/air.v4n2p22 (cit. on p. 20).
- [51] IBM Corporation. *IBM SPSS Modeler 18.2 Algorithms Guide*. 2018 (cit. on pp. 20, 21).
- [52] S. M. Holland. «PRINCIPAL COMPONENTS ANALYSIS (PCA )». In: 2008 (cit. on p. 20).
- [53] Ronald L. Wasserstein and Nicole A. Lazar. «The ASA Statement on p-Values: Context, Process, and Purpose». In: *The American Statistician* 70.2 (2016), pp. 129–133. DOI: 10.1080/00031305.2016.1154108 (cit. on p. 20).
- [54] Aurelien Geron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition*. 2019. ISBN: 9781492032649 (cit. on pp. 22, 24, 25, 29).
- [55] Rikiya Yamashita, Mizuho Nishio, Richard Do, and Kaori Togashi. «Convolutional neural networks: an overview and application in radiology». In: *Insights into Imaging* 9 (June 2018). DOI: 10.1007/s13244-018-0639-9 (cit. on pp. 22, 28).
- [56] Alex Sherstinsky. «Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network». In: *Physica D: Nonlinear Phenomena* 404 (Mar. 2020), p. 132306. DOI: 10.1016/j.physd.2019.132306 (cit. on p. 22).
- [57] Tom Heskes and Bert Kappen. «Learning processes in neural networks». In: *Physical review. A* 44 (Sept. 1991), pp. 2718–2726. DOI: 10.1103/PhysRevA.44.2718 (cit. on pp. 22, 23).
- [58] Z. You and B. Xu. «Improving training time of deep neural network with asynchronous averaged stochastic gradient descent». In: *The 9th International Symposium on Chinese Spoken Language Processing*. 2014, pp. 446–449. DOI: 10.1109/ISCSLP.2014.6936596 (cit. on p. 25).

- [59] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. <http://www.deeplearningbook.org>. MIT Press, 2016, pp. 164–223 (cit. on p. 25).
- [60] Krzysztof Gajowniczek, Leszek Chmielewski, Arkadiusz Orłowski, and Tomasz Ząbkowski. «Generalized Entropy Cost Function in Neural Networks». In: Oct. 2017, pp. 128–136. ISBN: 978-3-319-68611-0. DOI: 10.1007/978-3-319-68612-7\_15 (cit. on p. 26).
- [61] Rich Caruana, Steve Lawrence, and C. Giles. «Overfitting in Neural Nets: Backpropagation, Conjugate Gradient, and Early Stopping.» In: vol. 13. Jan. 2000, pp. 402–408 (cit. on p. 27).
- [62] Nusrat Ismoilov and Sung-Bong Jang. «A Comparison of Regularization Techniques in Deep Neural Networks». In: *Symmetry* 10 (Nov. 2018), p. 648. DOI: 10.3390/sym10110648 (cit. on p. 27).
- [63] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. «Dropout: A Simple Way to Prevent Neural Networks from Overfitting». In: *Journal of Machine Learning Research* 15 (June 2014), pp. 1929–1958 (cit. on p. 27).
- [64] E. Falletti, D. Margaria, M. Nicola, G. Povero, and M. T. Gamba. «N-FUELS and SOPRANO: Educational tools for simulation, analysis and processing of satellite navigation signals». In: *2013 IEEE Frontiers in Education Conference (FIE)*. 2013, pp. 303–308. DOI: 10.1109/FIE.2013.6684836 (cit. on p. 30).
- [65] *What is Matlab?* Matlab. URL: <https://www.mathworks.com/discovery/what-is-matlab.html> (cit. on p. 30).
- [66] A. E. Süzer and H. Oktal. «PRN code correlation in GPS receiver». In: *2017 8th International Conference on Recent Advances in Space Technologies (RAST)*. 2017, pp. 189–193. DOI: 10.1109/RAST.2017.8002960 (cit. on p. 30).
- [67] *STRIKE3 Project*. European GNSS Agency - STRIKE3. URL: <http://gnss-strike3.eu> (cit. on p. 32).
- [68] Patrik Eliardsson Oliver Towlson David Payne and Venkatesh Manikundalam. «Standardisation OF GNSS Threat Reporting and Receiver Testing through International Knowledge Exchange, Experimentation and Exploitation». In: (Jan. 2019) (cit. on pp. 33, 68–76).

- [69] Karl Pearson. «IX. Mathematical contributions to the theory of evolution. Second supplement to a memoir on skew variation». In: *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* 216.538-548 (1916), pp. 429–457. DOI: 10.1098/rsta.1916.0009 (cit. on p. 37).
- [70] D. N. Joanes and C. A. Gill. «Comparing measures of sample skewness and kurtosis». In: *Journal of the Royal Statistical Society: Series D (The Statistician)* 47.1 (1998), pp. 183–189. DOI: <https://doi.org/10.1111/1467-9884.00122> (cit. on p. 37).
- [71] Jian Sun. «Research on vocal sounding based on spectrum image analysis». In: *EURASIP Journal on Image and Video Processing* 2019 (Jan. 2019). DOI: 10.1186/s13640-018-0397-0 (cit. on p. 38).
- [72] Arthur Asuncion. «Signal Processing Applications of Wavelets». In: () (cit. on p. 40).
- [73] Carlos Mateo and Juan Talavera. «Short-Time Fourier Transform with the Window Size Fixed in the Frequency Domain». In: *Digital Signal Processing* 77 (Nov. 2017). DOI: 10.1016/j.dsp.2017.11.003 (cit. on p. 41).
- [74] Yufeng Zhang, Zhenyu Guo, Weilian Wang, Side He, Ting Lee, and Murray Loew. «A comparison of the wavelet and short-time fourier transforms for Doppler spectral analysis». eng. In: (2003) (cit. on p. 41).
- [75] Ahmet Taspinar. «Wavelet Transform in Machine Learning». In: (2018) (cit. on p. 42).
- [76] Babak Azmoudeh and Dean Cvetkovic. «Wavelets in Biomedical Signal Processing and Analysis». In: *Encyclopedia of Biomedical Engineering*. Ed. by Roger Narayan. Oxford: Elsevier, 2019, pp. 193–212. ISBN: 978-0-12-805144-3. DOI: <https://doi.org/10.1016/B978-0-12-801238-3.99972-0> (cit. on pp. 42, 43).
- [77] I. W. Selesnick, R. G. Baraniuk, and N. C. Kingsbury. «The dual-tree complex wavelet transform». In: *IEEE Signal Processing Magazine* 22.6 (2005), pp. 123–151. DOI: 10.1109/MSP.2005.1550194 (cit. on p. 43).
- [78] Chaichan Pothisarn, Jittiphong Klomjit, Atthapol Ngaopitakkul, Chaiyan Jettanasen, Dimas Anton Asfani, and I Made Yulistya Negara. «Comparison of Various Mother Wavelets for Fault Classification in Electrical Systems». In: *Applied Sciences* 10.4 (2020). ISSN: 2076-3417. DOI: 10.3390/app10041203. URL: <https://www.mdpi.com/2076-3417/10/4/1203> (cit. on p. 43).
- [79] Lokenath Debnath. «The Wigner-Ville Distribution and Time-Frequency Signal Analysis». In: (2002) (cit. on p. 45).

- [80] John O' Toole, Mina Mesbah, and Boualem Boashash. «A Discrete Time and Frequency Wigner–Ville Distribution: properties and Implementation». In: (Jan. 2005) (cit. on p. 45).
- [81] Leon Cohen and Patrick Loughlin. «leID1. Time-frequency analysis: Theory and applications». In: *The Journal of the Acoustical Society of America* 134.5 (2013), pp. 4002–4002. DOI: 10.1121/1.4830599 (cit. on p. 45).
- [82] Boualem Boashash and PJ BLACK. «An efficient real-time implementation of the Wigner-Ville distribution». In: 35 (Dec. 1987), pp. 1611–1618 (cit. on p. 46).
- [83] Eric Chassande-Mottin and Archana Pai. «Discrete Time and Frequency Wigner–Ville Distribution: Moyal’s Formula and Aliasing». In: (July 2005) (cit. on p. 46).
- [84] Monika Pinchas and Yosef Pinhasi. «A New Approach for the Characterization of Nonstationary Oscillators Using the Wigner-Ville Distribution». In: (June 2018) (cit. on p. 46).
- [85] L. Stankovic. «A method for time-frequency analysis». In: *IEEE Transactions on Signal Processing* 42.1 (1994), pp. 225–229. DOI: 10.1109/78.258146 (cit. on p. 46).
- [86] Zoukaneri Ibrahim M. and Milton J. Porsani. «Instantaneous frequency and Wigner-Ville distribution using the maximum entropy method: Application for gas and hydrates identification». In: (2013) (cit. on p. 46).
- [87] Mudamala Pavithra and R.M.S. Parvathi. «A survey on clustering high dimensional data techniques». In: *International Journal of Applied Engineering Research* 12 (Jan. 2017), pp. 2893–2899 (cit. on p. 49).
- [88] Carlos Sorzano, Javier Vargas, and A. Montano. «A survey of dimensionality reduction techniques». In: (Mar. 2014) (cit. on p. 49).
- [89] Osama Abu Abbas. «Comparisons Between Data Clustering Algorithms». In: *Int. Arab J. Inf. Technol.* 5 (July 2008), pp. 320–325 (cit. on p. 49).
- [90] Priya Kochar Barkha Narang Poonam Verma. «Application based, advantageous K-means Clustering Algorithm in Data Mining». In: (2016) (cit. on p. 49).
- [91] Jianhong Xue, Cindy Lee, Stuart G. Wakeham, and Robert A. Armstrong. «Using principal components analysis (PCA) with cluster analysis to study the organic geochemistry of sinking particles in the ocean». In: *Organic Geochemistry* 42.4 (2011), pp. 356–367. ISSN: 0146-6380. DOI: <https://doi.org/10.1016/j.orggeochem.2011.01.012>. URL: <https://www.sciencedirect.com/science/article/pii/S0146638011000301> (cit. on p. 49).

- [92] Laurence Morissette and Sylvain Chartier. «The k-means clustering technique: General considerations and implementation in Mathematica». eng. In: (2013) (cit. on p. 50).
- [93] Jiawei Han, Micheline Kamber, and Jian Pei. *Data Mining: Concepts and Techniques*. Jan. 2012. DOI: 10.1016/C2009-0-61819-5 (cit. on p. 50).
- [94] Pang-Ning Tan, Michael Steinback, and Vipin Kumar. *Introduction to Data Mining*. Jan. 2006 (cit. on p. 54).
- [95] Chaoqun Ma and Jianhong Wu. *Data Clustering: Theory, Algorithms, and Applications*. Vol. 20. Jan. 2007. DOI: 10.1137/1.9780898718348 (cit. on p. 55).
- [96] Ricardo Maronna. «Charu C. Aggarwal and Chandan K. Reddy (eds.): Data clustering: algorithms and applications». In: *Statistical Papers* 57 (Jan. 2015). DOI: 10.1007/s00362-015-0661-7 (cit. on p. 55).
- [97] D. Rathi, S. Jain, and S. Indu. «Underwater Fish Species Classification using Convolutional Neural Network and Deep Learning». In: *2017 Ninth International Conference on Advances in Pattern Recognition (ICAPR)*. 2017, pp. 1–6. DOI: 10.1109/ICAPR.2017.8593044 (cit. on p. 58).
- [98] Kavish Sanghvi, Adwait Aralkar, Saurabh Sanghvi, and Ishani Saha. «Fauna Image Classification using Convolutional Neural Network». In: vol. 13. May 2020, pp. 8–16 (cit. on p. 58).
- [99] M. Shaha and M. Pawar. «Transfer Learning for Image Classification». In: *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*. 2018, pp. 656–660. DOI: 10.1109/ICECA.2018.8474802 (cit. on p. 58).
- [100] Xiaowu Sun, Lizhen Liu, Hanshi Wang, Wei Song, and Jingli Lu. «Image classification via support vector machine». In: *2015 4th International Conference on Computer Science and Network Technology (ICCSNT)*. Vol. 01. 2015, pp. 485–489. DOI: 10.1109/ICCSNT.2015.7490795 (cit. on p. 58).
- [101] R. A. Nurtanto Diaz, N. Nyoman Tria Swandewi, and K. D. Pradnyani Novianti. «Malignancy Determination Breast Cancer Based on Mammogram Image With K-Nearest Neighbor». In: *2019 1st International Conference on Cybernetics and Intelligent System (ICORIS)*. Vol. 1. 2019, pp. 233–237. DOI: 10.1109/ICORIS.2019.8874873 (cit. on p. 58).
- [102] Dong-feng Cai Yun-lei Cai Duo Ji. «A KNN Research Paper Classification Method Based on Shared Nearest Neighbor». In: (2010) (cit. on p. 58).

- [103] Tomoko Maruyama, Norio Hayashi, Yusuke Sato, Shingo Hyuga, Yuta Wakayama, Haruyuki Watanabe, Akio Ogura, and Toshihiro Ogura. «Comparison of medical image classification accuracy among three machine learning methods». In: *Journal of X-Ray Science and Technology* 26 (Sept. 2018), pp. 1–9. DOI: 10.3233/XST-18386 (cit. on p. 58).
- [104] Saurabh Sanghvi Kavish Sanghvi Adwait Aralkar. «A Survey on Image Classification Techniques». In: (Jan. 2021) (cit. on pp. 58, 59).