IDEA League

Master of Science in Applied Geophysics

Research Thesis

Exploring Gravitational Waves Recordings with Machine Learning Techniques

Hamed Ali Diab Montero

August 9,2019

ETH zürich

Exploration & Environmental Geophysics Group
Seismology & Geodynamics Group
ETH Zurich
Switzerland

Exploring Gravitational Waves Recordings with Machine Learning Techniques

Master of Science Thesis

for the degree of Master of Science in Applied Geophysics at

by

Hamed Ali Diab Montero

August 9,2019

ETH zürich

Printed in Switzerland

IDEA LEAGUE
JOINT MASTER'S IN APPLIED GEOPHYSICS

Delft University of Technology, The Netherlands
ETH Zürich, Switzerland
RWTH Aachen, Germany

Dated: *August 9,2019*

Supervisor(s):

Dr. Cedric Schmelzbach

Dr. Men-Andrin Meier

Dr. Luigi Ferraioli

prof. Dr. Domenico Giardini

Dr. Filippo Broggini

Committee Members:

Dr. Cedric Schmelzbach

Dr. Men-Andrin Meier

Dr. Luigi Ferraioli

Dr. Filippo Broggini

Dr. Norbert Klitzsch

Abstract

The study of Gravitational Waves (GWs) opened a new window of possibilities to improve our understanding of the Universe. GWs provide suitable astronomical messengers for studying events that were not possible before through electromagnetic radiation, or in other cases complementing their observations. Ground-based interferometers like LIGO have been recording multiple GW events since the first detections in 2015. Despite the success of Earth-based observatories, the space limitations and noise sources on Earth point toward the need of building a spaceborne interferometer. The Laser Interferometer Space Antenna (LISA) is a planned project that will provide us with such a detector and will allow gaining access to lower frequency bands and more types of GW sources. To make the most out of LISA's strengths, it is important to identify and develop alternative data analysis tools which are more appropriate for low latency searches of GWs than the current ones in use. Machine Learning techniques are a promising candidate since they can provide high accuracies, higher speeds, and a lower computational cost. Therefore, they can be used for the development of Low Latency Detectors (LLD) of GWs, which will be used to analyze the LISA recordings. I propose to build a prototype LLD by using a Sliding Window Algorithm, which makes use of Convolutional Neural Networks (CNNs) as its classification mechanism. To implement the LLD, I first create datasets composed of synthetic GW recordings of two different GW source types: Galactic Binaries (GBs) and Merging Blackhole Binaries (MBHBs). Then, I transform these recordings originally represented only in the time domain, into the frequency domain, and the time-frequency domain and train two different ML architectures (CNNs and Fully-Connected Neural Networks) using both the original and the transformed data. A performance evaluation is done to select the best combination of ML architecture and domain representation for solving the detection task. The chosen combination is then used as the classifier mechanism of the LLD acting in windows of five days duration. The LLD is tested on one-year-long recordings with different levels of noise. The analysis suggests that the time-frequency domain representations offer the most promising results for detecting both types of sources (GBs and MBHBs) reaching high accuracies in recordings with low to moderate signal-to-noise ratio (SNR).

Acknowledgements

I would like to express my sincere gratitude to my supervisors for their guidance and support throughout my thesis. First, I would like to thank Dr. Cedric Schmelzbach for assisting me through the six-month process of the thesis. Mainly I want to thank him for his help with the drafting of the final document. Second, to Dr. Men-Andrin Meier for his insights into Deep Learning, his advice for the description and the evaluation of the results, and for his constant encouragement throughout the whole project. My thanks also extend to Dr. Luigi Ferraioli for his help with all the doubts regarding the topics of gravitational waves and their synthetic generation. Moreover, to Dr. Filippo Broggini for his general assistance with the software, HPC tools, and access to CSCS services, necessary to build up the Deep Learning models that I used during the thesis.

Finally, I would also like to thank Mr. Heinrich Horstmeyer for his support with all the hardware and software needed for the research and Duncan Edwards for providing the MATLAB code necessary for creating part of the synthetic data used for the project . Last but not least, thanks to my flatmates for hearing me talking countless nights about my research on machine learning and gravitational waves. Thanks, Aneri, Adrian, Jose, and Ossama, your ideas and applications of artificial intelligence in the area of Robotics were very inspiring for this outcome.

ETH Zurich Swiss Federal Institute of Technology — Hamed Ali Diab Montero
August 9,2019

Table of Contents

List of Figures

List of Tables

Acronyms

Signal Processing

CWT Continuous Wavelet Transform

EWS Early Warning System

LLD Low Latency Detector

SNR Signal-to-Noise Ratio

STFT Short Time Fourier Transformation

Gravitational Waves

GB Galactic Binaries

GW Gravitational Waves

LIGO Laser Interferometer Gravitational Wave Observatory

LISA Laser Interferometer Space Antenna

MBHB Merging Black-hole Chirping Binaries

Machine Learning

AI Artificial Intelligence

ANN Artificial Neural Network

CNN Convolutional Neural Network

DL Deep Learning

FCNN Fully-Connected Neural Network

ML Machine Learning

RF Random Forest

SVM Support Vector Machine

Institutions

DUT Delft University of Technology

ETH Swiss Federal Institute of Technology

RWTH Aachen University

Chapter 1

Introduction

The study of waves has been one of the most important driving forces behind human progress. In our daily lives, we make use of electromagnetic and mechanical waves for several purposes that range from energy production to protection against natural hazards. Some basic examples of these wave-related applications are human communication, illumination, GPS, or even the internet, all fundamental tools of society. Waves are either an oscillation of matter or oscillations that propagate through vacuum carrying electromagnetic radiant energy. In contrast, gravitational waves (GWs), although also oscillatory phenomena, are of a very different nature having the fabric of space-time itself as their medium of propagation. This characteristic opens a whole new window for inventions and discoveries since GWs can travel extremely long distances virtually unobstructed by masses [Danzmann, 2017]. Most importantly, they may carry information about two of the most intriguing scientific enigmas. The first is a greater insight into the essence of gravity, one of the four fundamental forces of nature, the weakest, and the less understood [Cornish, 2019]. The second enigma, the gravitational wave background (GWB) left behind after the Big Bang, which offers the possibility of a better understanding of the origins of the Universe [Christensen, 2018].

Any accelerated mass can produce gravitational waves. However, massive objects with extreme accelerations are required for producing measurable ones [Schutz, 1983]. This is one of the main reasons why astrophysics, which studies the evolution, interaction and nature of celestial bodies, has been the branch of knowledge with the most applications and interest in GWs at the moment. This has encouraged the development of some of the highest precision equipment created in human history, in order to detect these particular kinds of waves which have amplitudes in the order of 10^{-21}. These attempts were successful in September of 2015 when the Laser Interferometer Gravitational-Wave Observatory (LIGO) located in The United States made the first direct observation of gravitational waves. This measurement guaranteed the LIGO team the Physics Nobel Prize in 2017 and opened new possibilities for Astrophysics [Duncan Brown, 2004]. In fact, this finding is particularly important for multi-messenger astronomy [Tyson Littenberg, 2019] which looks for the combined analysis of the four messengers we currently have access for studying the Universe: electromagnetic radiation (studied with telescopes and radio telescopes), neutrinos, cosmic rays and now gravitational waves [Branchesi, 2016].

Although LIGO has opened all the possibilities mentioned above, there are some limitations in the ranges of frequencies it can measure, currently from 10 Hz to 10 kHz, and the level of precision it can achieve. One of the most significant limitations has to deal with the length of the arms of the equipment (4 km in total) that can be constructed on Earth. This issue limits the sensitivity of the waves that can be detected because the distortion caused by GWs is expressed as strains (relative length changes) measurements; the longer the arms of our devices, the more sensitive it becomes. Additionally, environmental noise caused by human activities, seismicity, and simple forces like the one produced by air can interfere with reaching such a high level of precision in these small measurements on Earth. This implies that extreme isolation needs to be guaranteed to protect the devices. Therefore, looking for alternatives to overcome these limitations, the European Space Agency (ESA) in collaboration with the National Aeronautics and Space Administration (NASA) proposed the construction of the Laser Space Interforemeter Antenna (LISA), which would be the first space-borne gravitational wave interferometer built-in human history [Danzmann, 2017].

Out in space, the devices to be built can reach lengths that are enormous when compared to the ones on Earth by having drag-free spacecraft that use active lasers links, traveling through the vacuum,acting as arms. For instance, according to the last approved proposal of LISA in 2017 [Danzmann, 2017], it is planned that the 'arms' of the device will be 2.5 million km in length. This way, LISA can give access to lower frequency bands than LIGO, having a range between 0.1 mHz and 100 mHz. These bring two advantages: The first one is to record more events with very different characteristics and sources than the currently studied. The second one is to be able to record with better accuracy the first of events like the Merging Blackhole Binaries (MBHBs). For instance, according to [Curt Cutler, 2019], the detection of the colliding blackholes of 2015 related to the Nobel prize, could have been detected a few years before during their merging phase by a space-borne interferometer.

Since LISA will be able to detect the first stages of evolution of many GW sources, it would be possible to use it for low latency detections in multimessenger astronomy [Shaun Hooper and Luan, 2010]. For using LISA as a Low Latency Detector (LLD), it is vital to make use of tools of high efficiency and speed in scanning and analyzing the data recorded. The technique currently used by LIGO is called matched filtering [Benjamin J. Owen, 1999], and it has given excellent results in performing the detection of events recorded by ground-based interferometers [Emanuele Berti and Cavaglià, 2007] [Birjoo Vaishnav and Deirdre, 2007]. It uses a template searcher which tries to identify possible events in the recordings by matching, fitting and adapting the data to the patterns found in catalogs of past recorded events or synthetic events built from theoretical models of known sources. Nevertheless, this technique works in detecting only events from which templates are available, and that is very computationally expensive to create [Hunter Gabbard and Messenger, 2018]. Moreover, due to the computational power needed for it to work, it can perform slowly for real-time applications which discourage is us for low latency detections [Hunter Gabbard and Messenger, 2018].

Some efforts have been done to use Machine Learning techniques as tools that are nearly as accurate as Match Filtering for gravitational waves detection [George and Huerta, 2017] [Daniel George and Huerta, 2017], [N. Muund, 2017] [Andrzej Krolak and Vallisneri, 2008b]. The work of [George and Huerta, 2017] proofed Deep Learning(DL) could be used for detecting GWs and estimating some of their source parameters obtaining results which are very close to the ones using Matched Filtering. Their system is constructed us-

ing time-domain represented recordings of LIGO measurements. Other authors like ([Hunter Gabbard and Messenger, 2018]) have constructed similar DL detectors with the use of GW templates, while others have built models fed by spectrograms but only for the detection of glitches in the recordings [Daniel George and Huerta, 2017]. Therefore, I propose to extend these efforts and use DL for GW detection, but in this case using GW data represented in multiple domains for building tools for low latency detection of events occurring in the frequency band that will be accessed with LISA.

In this project, I build a prototype Low Latency Detector via a Sliding Window Algorithm which makes use of Deep Learning architectures as its classification mechanism. I produce the datasets constituted by synthetic data recordings simulated in the time-domain and transformed into the frequency-domain and the time-frequency domain. These representations are done to identify the most promising ones for low latency detection tasks. The gravitational waves I analyze are mathematical simplifications of Galactic Binaries (GBs) and Merging Blackhole Binaries (MBHBs) responses under different levels of signal-to-noise ratio (SNR) ranging from 0.1 to 100. I do an exhaustive hyperparameter optimization analysis to find the most accurate and precise CNNS and FCNNs models to be used for the detector. The best models found during this evaluation process are the ones implemented as the classifier mechanisms of the sliding window of the LLD. The performance and sensitivity of the LLD were evaluated on one-year recordings with the presence of several MBHBs, placed in segments of the data with different SNR. After applying this prototype detector, the analysis suggests that time-frequency domain representations offer the most favorable results for the early detection of MBHBs. It also allows reaching the highest accuracy when dealing with signals with the lowest SNR.

1-1 Motivation and Objectives

The main objectives of this thesis are:

- Creating a Low Latency Detector(LLD) for LISA, that can act as an Early Warning System (EWS) for multi-messenger astronomy purposes.
- Evaluating the usefulness of machine learning techniques, specifically deep learning ones, for studying gravitational waves.

The specific objectives of this thesis are:

- Finding the most appropriate representation of gravitational waves recordings, in terms of time and frequency domain that allows their detection under high levels of noise or the presence of numerous overlapping sources' responses.
- Finding the most promising neural network architecture for the tasks of detection and analysis of gravitational waves recordings.

1-2 Outline

This thesis consists of four main parts and is structured as follows. In the first part, consisting of chapters 2 and 3, I give a theoretical framework about the theory of Gravitational Waves (GWs) and of Deep Learning (DL) algorithms. In chapter 2, I explain what GWs are, their characteristics, the type of sources that generates them, the measuring devices that have been developed for their detection, and the specific characteristics of LISA. In Chapter 3, I explain the general workflow for developing a machine learning project, the specific tasks that need to be addressed for building a DL model and introduce the metrics that are used in the project to evaluate their performance.

In the second part, in Chapter 4, I explain the methodology I follow for the construction of the Low Latency Detector (LLD). Additionally, I explain the generation of the synthetic datasets for training the DL models and how the Sliding Window Algorithm which act as classifier works. Afterward, in the third part which consists of chapter 5, I present a hyperparameter analysis needed for building the Artificial Neural Networks (ANNs) and finding the best DL models to be applied as part of the LLD. Moreover, I show some results of running the LLD algorithm in a one-year recording with presence of MBHBs at different levels of noise.

Finally, in Chapter 6, I do a discussion of the technical difficulties, limitations of the method, a comparison with other authors work, present the conclusions of my work, and I make an outlook of future work to be done.

Chapter 2

Gravitational Waves

2-1 Theory

Gravitational waves are ripples in space-time caused by massive objects moving with extreme accelerations [Sathyaprakash and Schutz, 2009]. Albert Einstein predicted their existence as a consequence of his theory of General Relativity. He proposed in his work that these types of waves should share a series of similarities with light: they both oscillate sinusoidally, travel at the same speed, and they are both transverse waves with an oscillatory polarized metric that operates in a plane perpendicular to them. Even though there are also some crucial differences between them, being the first and most fundamental one that GWs travel through the Universe virtually unimpeded by masses [Schutz, 1983]. This property is one of the principal focus of interest for their study since they provide a better tool (messenger) for analyzing phenomena like Blackholes and Dark Matter when compared to Electromagnetic Radiation. The second one is that the polarization of GWs are at 45° angle from each other instead of 90° [Sathyaprakash and Schutz, 2009], these are named h_{plus} and h_{cross}, and are illustrated in Figure 2-1. This transverse nature of GWs and their polarizations provoke that once they pass through an object, they stretched it and squeezed in various directions at the speed of light. This instantaneous stretching and squeezing is a fundamental aspect to consider for developing devices for measuring GWs, especially laser interferometers that are the most extended, precise, and popular devices for measuring GWs in our days.

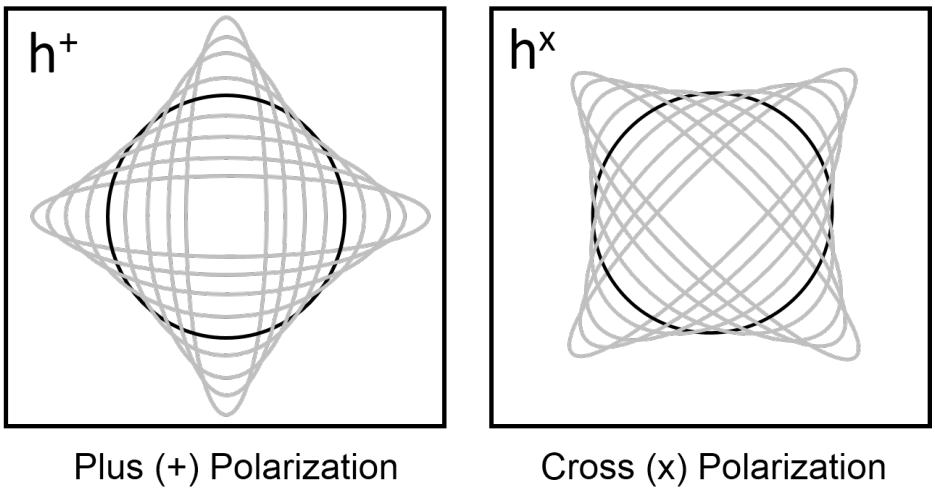

Figure 2-1: Plus-polarization and cross-polarization of a gravitational wave. The gray lines illustrate the stretching and swelling that an object, represented by the black circle, experiments by the effect of each of the polarizations of the passing GW. Modified from [Sathyaprakash and Schutz, 2009].

Although the physics of gravitational waves is explained by the General Relativity and the Special Relativity of Einstein, it can also be described using Newtonian physics less rigorously. A very intuitive approach is explained in [Schutz, 1983], where a GW is described as a consequence of one of the main postulates of Special Relativity which is that no information should be able to propagate faster than the speed of light. The transference of information that comes from instantaneous changes in mass density from a source field needs to satisfy this principle too. For this, a retardation factor is added the total gravitational Newtonian field, which translates in a far-field component that is the only one noticeable given a considerable distance from the source. These are then the GWs, which are never more than a small part of the overall external gravitational field of the emitter. They carry time-dependent tidal forces while the stationary part adds to the overall sources of the Universe [Schutz, 1983].

2-2 Sources

As it was described before in the introduction, any accelerated mass can produce gravitational waves. Nevertheless, some aspects need to be considered to measure them. Firstly, since the stationary part of the gravitational field is very strong at close distances, for measuring gravitational waves, it is required for the detector to be at least one wavelength away from the source [Schutz, 1983]. At that distance, the tidal effect of the GWs starts to be more significant than the stationary field. This means that to produce human-made detectable gravitational waves we would need to produce extreme high frequencies sources, which in any case with the masses available on Earth, are in order of magnitudes that are far away from being detectable [Sathyaprakash and Schutz, 2009]. Therefore, it is more logical to look for natural sources in the Universe which are very massive so that they produce enough gravitational radiation and that is moving at high acceleration.

Secondly, it is required for the motion of the accelerated massive bodies to be non-spherically symmetric [Schutz, 1983]. This requisite comes from the fact of having mass gravitational quadrupole tensors [Sathyaprakash and Schutz, 2009]. This condition is commonly satisfied in the Universe by rotation of masses and tidal interaction between binary systems [Schutz, 1983].

Fortunately, there are several phenomena in the Universe that satisfy the requirements mentioned above. Among the most common sources, we can list Galactic Binaries (GBs), Merging Blackhole Binaries (MBHBs), Extreme Mass Ration Inspirals (EMRI), or the Space Radiation Background. There is a general classification of the GWs into four main types of sources that I list below:

- **Galactic Binaries**: These consist of a pair of spinning massive objects. They have the characteristic of emitting GWs continuously with the same frequency and amplitude.

- **Compact Binary Inspiral**: These sources are orbiting massive and dense pairs of celestial bodies. The most common examples of these bodies are white dwarf stars, black holes, and neutron stars. Due to the large masses of the components of these binaries, and their mutual gravitational attraction, these events tend to start a merging process. During the merging, the GW emission peaks making them one of the most common and promising observable events by GW detectors. The detections of 2015 was of one of these type of sources, specifically of a MBHB [Sathyaprakash and Schutz, 2009].

- **Burst from Unknown Sources**: They are short-duration GWS that are unknown and produced by unanticipated sources. They can be generated, for instance, by supernovae or gamma-ray bursts.

- **Stochastic Background Radiation**: These are the superposition of a large number of independent sources that may include the earliest events in the Universe, including remanent gravitational waves from the Big Bang. [Cornish, 2019] [Cornish, 2017] [Christensen, 2018].

In this project, I work with data from two out of these four different types of sources. The first ones are Galactic Binaries (GBs), which are monochromatic in terms of frequency. The second ones are Merging Blackhole Binaries. A graphical representation comparing the expected sensitivity of the Laser Space Interferometer Antenna (LISA) to different types of sources is shown in Figure 2-2. There the Galactic Binaries are represented as purple points which tend to overlap, forming a 'point cloud' that can sometimes hide other sources' responses of greater interest in the recording. The MBHBs are represented by curves which color changes from yellow to black, as does their frequency content as their system evolve. This characteristic change in frequency and broad frequency band $10^{-5}\,Hz$ to $10^{-1}\,Hz$ for LISA is one of the main reason for considering them good candidates as detection goals of prototypes Low Latency Detectors (LLDs).

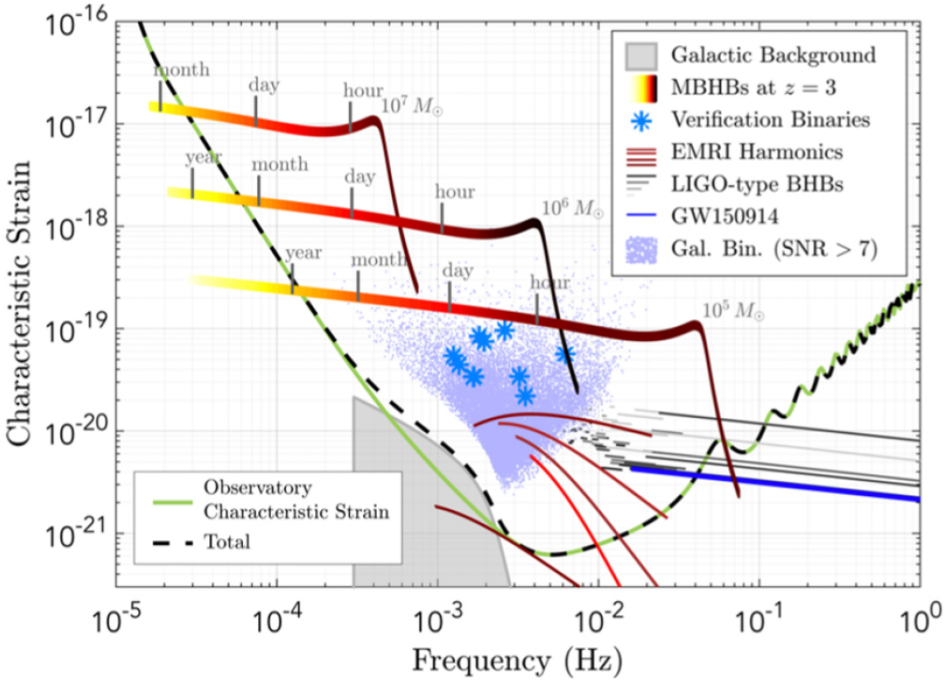

Figure 2-2: Expected sensitivity and potential sources signals in the frequency range of LISA. Multiple GW sources are displayed. The most representative are the Galactic Binaries (GBs) that form a cloud point of monochromatic events, and the Merging Blackhole Binaries (MBCBs) that range from $10^{-5}\,Hz$ to $10^{-1}\,Hz$. Taken from [Cornish, 2019]

2-2-1 Source Parameters

Every source which generates GWs hold a number of particular characteristics which allows us to define them either spatially or physically. Researchers translated these characteristics into a series of mathematical and physical parameters that can allow them to understand and reproduce the behavior observed in such sources. These are called 'source parameters' and are information that can be extracted from the gravitational waves recorded in our measuring devices. They are important because they allow us to produce more realistic simulations of phenomena occurring in the Universe, corroborate theories like Einstein's Special Relativity, and have the potential of getting a more in-depth insight in events like the Big Bang [Tyson Littenberg, 2019] [Christensen, 2018] [Sathyaprakash and Schutz, 2009] [George and Huerta, 2017] [Schutz, 1983].Some of these parameters can be regarded as common to any gravitational wave source and are summarized in Table 2-1.

Table 2-1: Common source parameters for any type of GW source. Taken from [Arkadiusz Blaut, 2010]

Parameter	Symbol	Unit
Ecliptic latitude	B	[rad]
Ecliptic longitude	γ	[rad]
Polarization angle	ψ	[rad]
Inclination	t	[rad]
Luminosity distance	D	[parsec]

2-2-2 Galactic Binaries

According to [Arkadiusz Blaut, 2010], a Galactic Binary (GB) is a type of source whose main characteristic is to generate a monochromatic wave. In Figure 2-3 I show a graphical representation of how a GB would look. The two components of the system spinning around each other follow a non-symmetrical centrifuge movement that allows the production of the residual field which translates into a measurable GW. Galactic Binaries are the most numerous expected astrophysical sources for LISA, in particular, double white dwarf binaries [Nelemans, 2013]. In Table 2-2 I show the source parameters that correspond to the Galactic binaries.

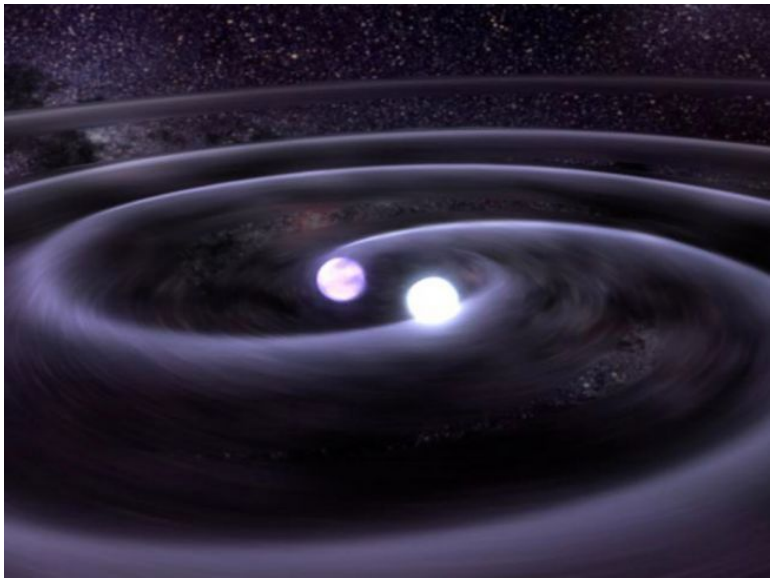

Figure 2-3: Artist representation of the RX J0806.3+1527 White Dwarf Star Spiral, an example of a Galactic Binary. Credit Illustration: D. Berry - Goddard Space Flight Center (GSFC)

Table 2-2: Specific Parameters of Galactic Binary Sources. Taken from [Arkadiusz Blaut, 2010]

Parameter	Symbol	Unit
Amplitude	A	1 [GW strain]
Frequency	f	[Hz]
Initial GW Phase	ϕ_0	[rad]

Although the waves generated by the GBs can described by simple sinusoidal, LISA's measurements of these events will be more complex. Factors like LISA's motion, its orbit around the Sun, and its orientation to the coming GW introduce modulations in the phase and the amplitude in the recorded wave [Andrzej Krolak and Vallisneri, 2008a]. An example of an expected GB recording is shown in Figure 2-4. The modulations make the wave look like "the superposition of multiple sinusoids of smaller amplitude" [Andrzej Krolak and Vallisneri, 2008a].

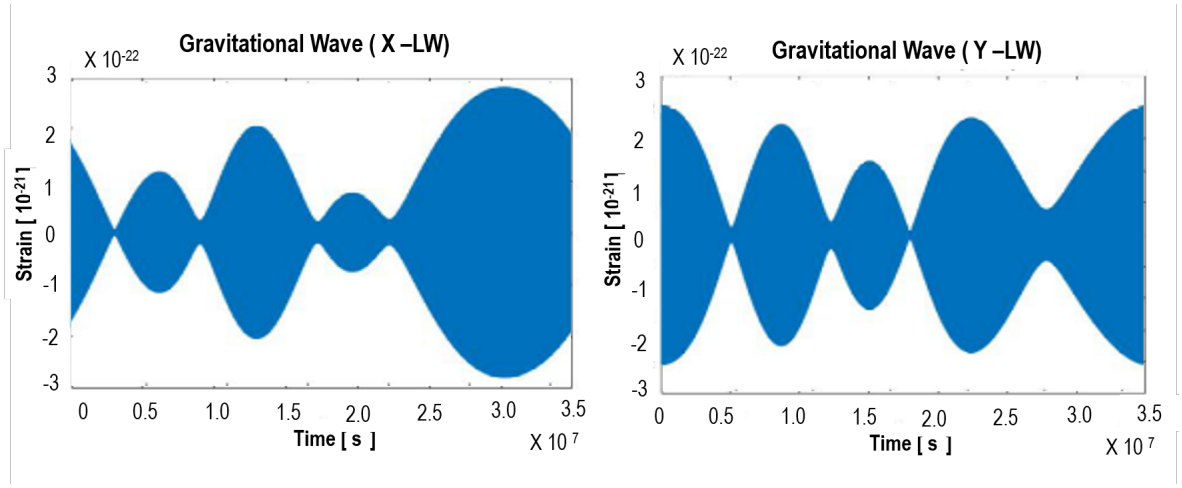

Figure 2-4: Time-domain representation of a typical Galactic Binary. Two (X,Y) out of the Three channels for one-year TDI 1.0 simulation of a GB recording are shown. It is visible in both graphs the modulation present in LISA for GB recordings

2-2-3 Merging Blackhole Binaries

Blackholes are a phenomenon in the Universe that is commonly found as binary systems [H. Sana, 2012].In Figure 2-5 I show the graphical representation of a simulation of a Merging Blackhole Binary (MBHB). These types of astronomical bodies have a tremendous gravitational attraction among them, and they tend to reduce their distance between them progressively with time until eventually merging. There are multiple theories about how these binary systems formed. Some of them point towards a dynamic formation in dense stellar clusters [Banerjee, 2010], other ones indicate that massive stars in their evolution come close together as they evolve into black holes until forming the binary system [Banerjee, 2010], or that they have been from the very beginning close together [S. E. de Mink, 2016].

The specific source parameters for this GW source are shown in Table 2-3. Among these parameters, the mass is the one that dominates the behavior of black holes and their binaries. It is the criteria to classify them into Stellar-mass (five to several tens of solar masses), Intermediate-mass(hundred to several thousand of solar masses) and Supermassive (ranging from a million to a billion solar masses) [Sathyaprakash and Schutz, 2009]. Supermassive black holes exhibit much lower frequencies than the other categories, but much higher amplitudes which make them a focus of interest for detectors like LISA. In MBHBs, is the ratio of the masses what dictates how the merging of the binary system evolves. This change is represented in the signal they emit in their GWs and can be seen in Figure 2-6. As the orbiting black holes give off these waves, the orbit decays, and the orbital period decreases. This stage is called a binary black hole inspiral. The black holes merge once they are close enough. Once merged, the single hole settles down to a stable form, via a stage called ringdown, where any distortion in shape is dissipated as more gravitational waves are emitted In the very last stage of the merger, the black holes can reach incredibly high velocity, and the gravitational wave amplitude reaches its peak.

Table 2-3: Source Parameters Merging Black-hole Binaries. Taken from [Arkadiusz Blaut, 2010]

Parameter	Symbol	Unit
Parameter	Symbol	Unit
Mass of first BH	m_1	[Solar Mass]
Mass of second BH	m_2	[Solar Mass]
Time of Coalescence	t_c	[s]
Angular Orbital Phase at time t=0	ϕ_0	[rad]
Tapering radius	R	[Total Mass]

Figure 2-5: Simulation of the event GW150914, a Merging Blackhole Binary (MBHB) detected by LIGO. Credit Illustration: SXS (Simulating eXtreme Spacetimes) project

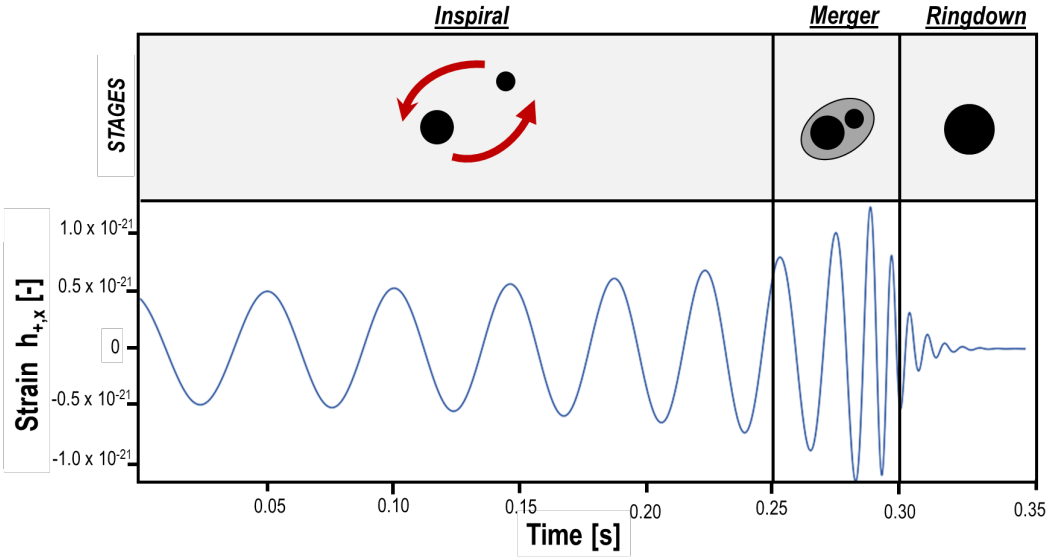

Figure 2-6: Time domain representation of a Merging Black-hole Binary (MBCB) showing the changes in frequency and amplitude during the merging process. Modified from [Abbott, 2016]

2-3 Gravitational Wave Detection

There are two principal classes of GW detectors. The first ones are the *resonant mass detectors* [Sathyaprakash and Schutz, 2009], which are the oldest ones and rely on the oscillations produced by the transference of the energy of the GWs into a massive body. One example of these is the bar detector pioneered by Weber in 1960s. It consisted of a cryogenically cooled massive cylinder that stretches due to the elastic energy gained by the vibration caused by passing GWs [Schutz, 1983]. These detectors are narrow band which range from frequencies of $500\,Hz$ to $1500\,Hz$ [Schutz, 1983]. The second one is the *beam detectors*, which are the most popular and precise in our days and that make use of beams of electromagnetic radiation (lasers) to monitor the passage of a GW [Sathyaprakash and Schutz, 2009].

The principle behind the function of beam GW detectors is called interferometry. It consists of the use of the superposition of combined waves in such a way that a pattern is created during constructive interference and nothing is detected during destructive interference [Sathyaprakash and Schutz, 2009]. For GW astronomy, the Michelson interferometer configuration is commonly used, and it is shown in Figure 2-7. Two laser devices point perpendicularly. One of the lasers is pointed towards a beam splitter that produces two rays. These laser rays bounce back, acting as a ruler measuring the distance that has been traveled by them. If the distance that the rays traveled is the same in both arms, then they cancel each other, having no response in the light detector. Even though, an interference pattern appears if there is a change in the length of one of the arms, even the slightest, producing a detection. When the gravitational wave passes by it stretches in one direction and contracts in the other one due to its polarizations, which produces these interference patterns.

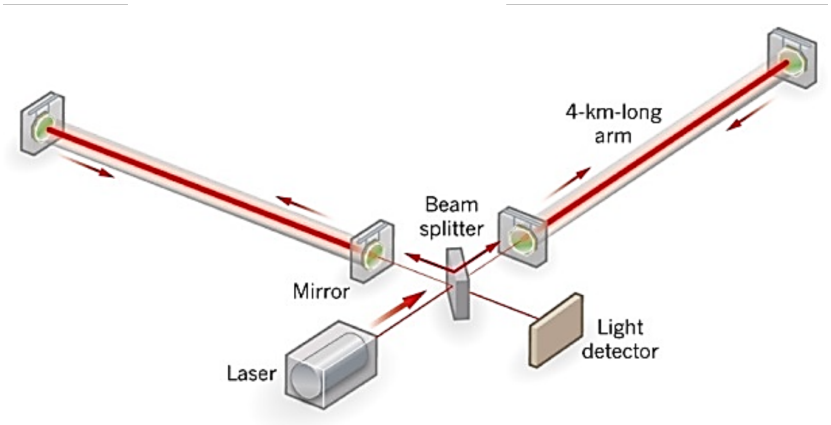

Figure 2-7: Basic Michelson Interferometer with Fabry Perot cavities and Power Recycling mirror. Reproduced from LIGO Caltech (2019)

2-3-1 Ground-Based Interferometry

Multiple Interferometer Observatories have been built all around the world, using more advanced and highly precise modifications of the Michelson Interferometer to detect and analyze GWs. The Laser Interferometer Gravitational-Wave Observatory (LIGO) with facilities in Hanford, Washington, and Livingston, Louisiana provided the first direct observation of grav-

itational waves in 2015. In Figure 2-8 [A] and [D] I show both of these facilities of LIGO which have arms with a length of 4 km.

In Figure 2-8 [B] and 2-8 [C] I show a comparison of the detection of the event GW150914 of 2015 represented in the time-frequency-domain and time-domain respectively. It is an MBCBs event identified using the technique called Matched Filtering. This method consist in the comparison of a wave recording with a template produced by simulations or past events for detecting a similar signal in it[Benjamin J. Owen, 1999] [Emanuele Berti and Cavaglià, 2007] [Birjoo Vaishnav and Deirdre, 2007]. There are two curves in the time-domain representation for every observatory. These curves correspond to the simulation of an MBCB and the recorded data. There is a very good matching between the expected gravitational wave and the one recorded. This measurement conferred the LIGO team, the Physics Nobel Prize in 2017. Nevertheless, this method has been found to have some drawbacks like been blind to new sources events, high consumption of computational resources, and a limited number of dimensions to be considered in the search [George and Huerta, 2017]. This implies the need of developing new tools and methodologies that can overcome these difficulties.

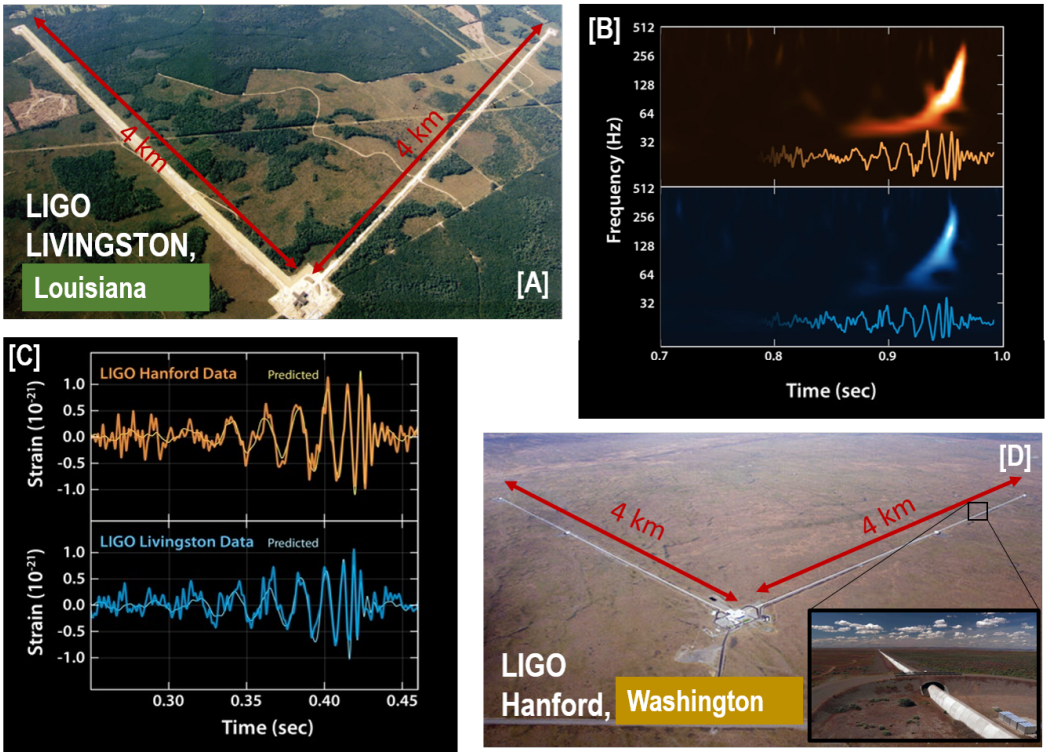

Figure 2-8: Representation of the system used by the ground-based interferometers like LIGO with Data Examples Aerial view of the Laser Interferometer Gravitational wave Observatory (LIGO) facilities in [A] Livingston, Louisiana and [D] Hanford, Washington. [B] Time-Frequency-domain and [C] Time-Domain representations of the MBHB event GW150914 detected in 14 September 2015. It was the first direct observation of gravitational waves and conferred the LIGO team the Nobel Prize in 2017. In [C] a comparison is made between the matching of the predicted curve of a model simulation (thin linewidth curves) and the recorded events (thick linewidth curves). Modified from LIGO Caltech (2019)

2-3-2 Spaceborne Interferometry

..

The Laser Interferometer Space Antenna (LISA) is an ESA mission with support from NASA expected to launch at the early 2030s. It consists in a constellation of three spacecrafts, distanced 2.5 million km from each other, forming the corners of an equilateral triangle that follows a heliocentric orbit. In Figure 2-9 [A] I show a drawing that illustrates how LISA would look like and in Figure 2-10 graphical representation of its orbit. The constellation will use laser interferometry to precisely measure the distances between spacecrafts by pointing between widely separated freely falling test masses housed in each one of them [Danzmann, 2017]. This technology is being tested in missions like LISA Pathfinder, shown in Figure 2-9 [B], which was launched in December 2015 and performed inspection measurements for two test masses in a near-perfect gravitational free-fall with a very high accuracy [Michele Armano, 2019]. LISA should observe low-frequency GWs ranging from 0.01 mHz to 10 mHz, allowing to analyze the radiation from long cosmological distances, and complementing the ground-based experiments already being operated.

This initiative poses a lot of challenges. The translation, rotation, and orientation of the constellation will be changing during the whole year, moving all the time with respect to the source. Additionally, the positioning system between spacecrafts needs to be extremely precise to have reliable measurements, especially to guarantee equal arms lengths since there are no rigid arms and that there is a relative speed between spacecrafts. This non-stationarity is a problem for processing the data and raises the need for developing advanced tools for making corrections to it. A series of generations of corrections known as Time Delay interferometers (TDI) have been developed [Andrzej Krolak and Vallisneri, 2008a] to take into consideration all of these effects, and are summarized in Table 2-4.

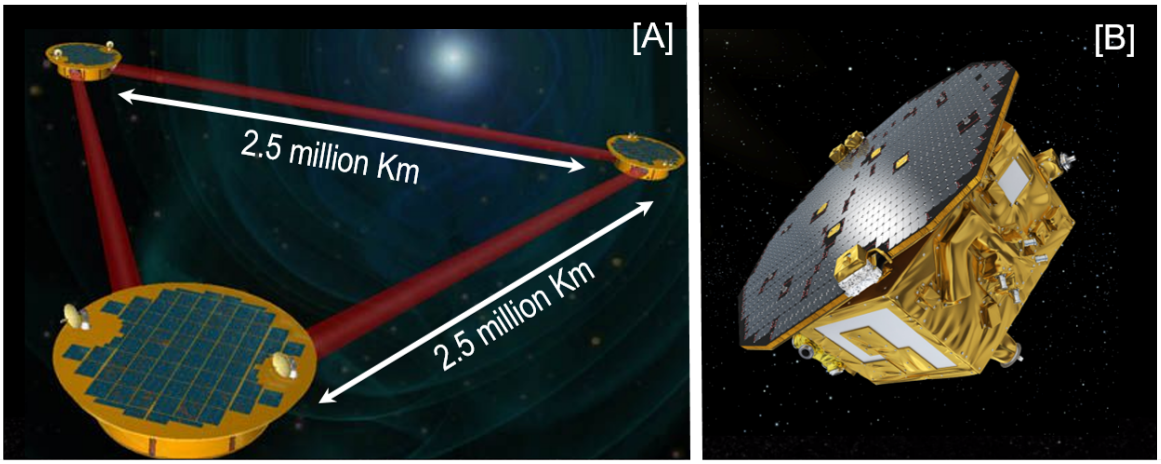

Figure 2-9: [A] Schematic representation of LISA showing the satellite constellation of three spacecrafts localizing. Credits ESA/NASA (2019) [B] Schematic representation of the LISA Pathfinder mission.(Not to scale) [Michele Armano, 2019]

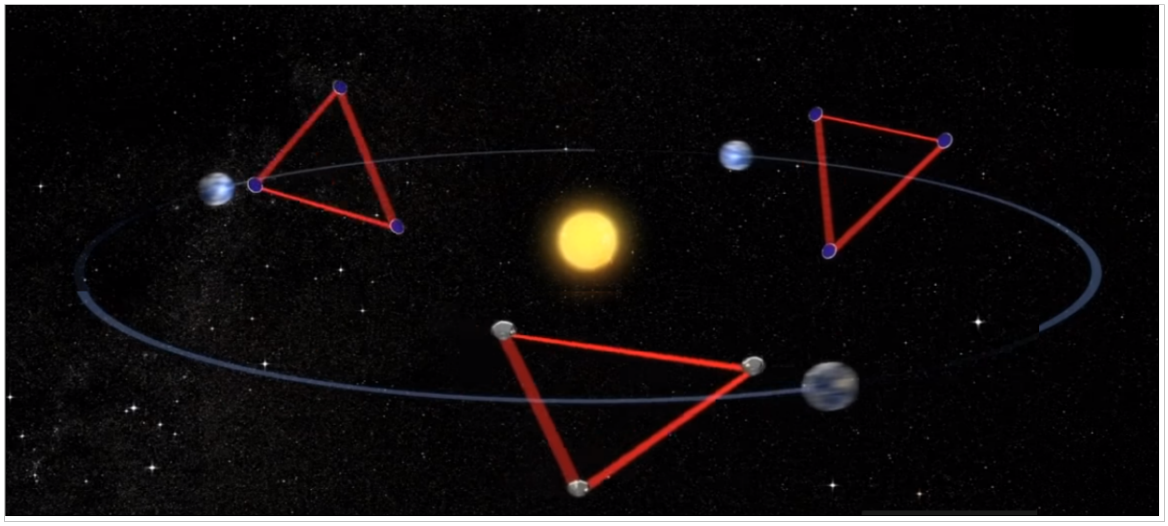

Figure 2-10: Representation of LISA and its orbit with respect to Earth and the Sun (Not to scale) Credit: ESA/NASA (2017)

Table 2-4: Time Delay Interferometry Generations. [Andrzej Krolak and Vallisneri, 2008a]

Compensations	TDI 1.0	TDI 1.5	TDI 2.0
For different arm lengths	X	X	X
For non-stationary position and orbiting		X	X
Flexible (non-rigid) constellation			X
Adding constant relative velocity between spacecrafts			X
Adding accelerations between spacecrafts			

Data Challenges

A task force called "The Mock LISA Data Challenge (MLDC) " was formed in 2006 with the purpose of encouraging the development of data analysis tools and models for the future LISA mission [Keith A. Arnaud, 2006]. They have collectively formulated several challenge problems of maximum efficacy, to establish criteria for the evaluation of the analyses and development of standard models for LISA. The tasks addressed in these challenges include the detection of specific signals in very noisy environments or the need for estimating the specific source parameters from a given data. In Figure 2-11 I show an example of two of those data challenges. These are of the special interest for this project since they are related to the two types of sources that I use for building a Low Latency Detector (LLD).

The first challenge can be seen in Figure 2-11 [A] where a single galactic binary has been burden into a high level of noise that makes it very difficult to identify it. In Figure 2-11 [B] I show the noise-free counterpart of that same galactic binary recording. From the representation, it is possible to conclude that the frequency of the GB is approximately $0.01\,Hz$. Finally, in 2-11 [C], the task is to find a single MBCB also in a very noisy recording. The noise-free data is shown in green.

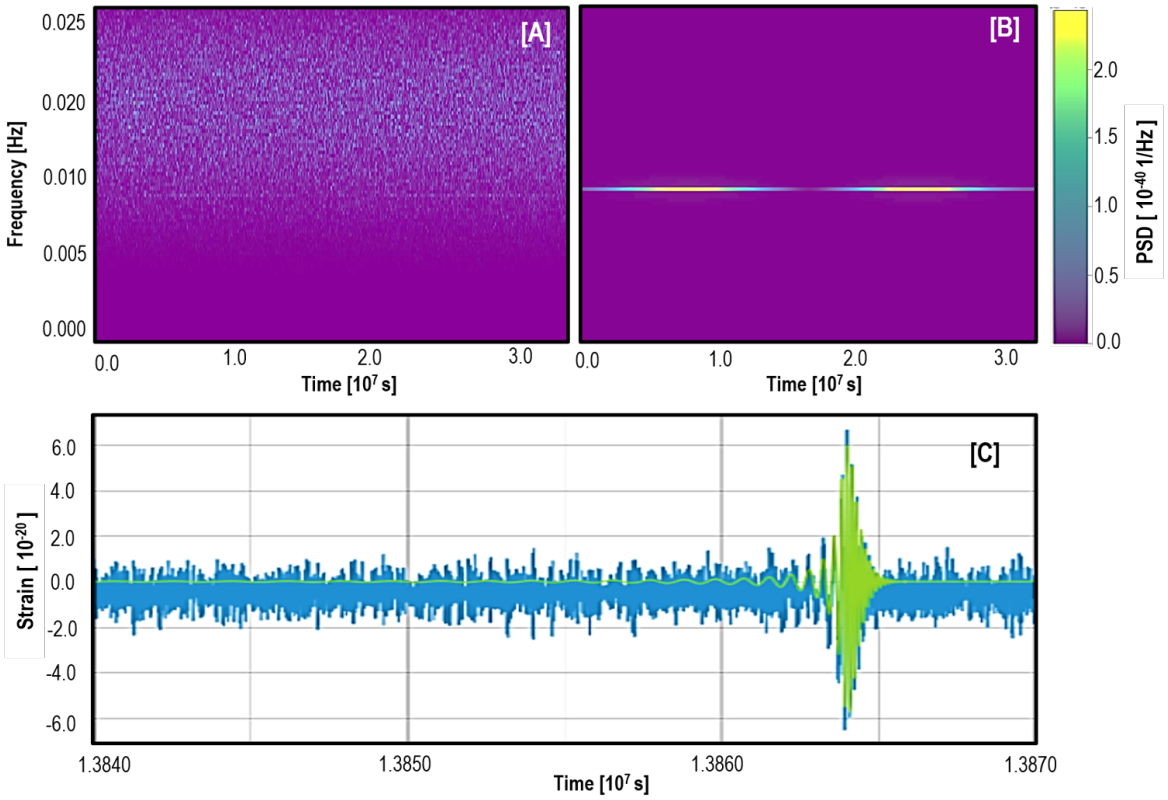

Figure 2-11: Data challenges examples to train for the mission and that were the problems addressed during this project [A] Data Challenge 1.1.1.C. With Noise [B] Data Challenge 1.1.1.C. Noise-free. Modified from NASA Astrogravs (2006) [C] Data Challenge LDC1-1. Modified from LISA Data Challenge 1: Radler (2019)

Chapter 3

Deep Learning Techniques

Machine learning (ML) is a branch of the computational sciences that has gained special attention during the last years. I define it as the implementation of algorithms developed in the last century, and other computational techniques, with the purpose that machines can use inductive thinking to its maximum exponent to extract general pattern and knowledge from particular observations. ML is one of the main driving forces behind the Artificial Intelligence (AI) revolution of the last decade, making it one of the most promising inventions for industry and research at the moment. It is so relevant today in our society that it is initiating the Fourth Industrial Revolution in history due to its great benefits for the automation of tasks.

There are several algorithms commonly used for attending Machine Learning tasks. Among them, the most commonly known are Support Vector Machine (SVM), Clustering, Random Forests (RF), Logistic Regressions, Artificial Neural Networks (ANNs), among others [Kotsiantis, 2007]. There are a set of steps that need to be followed when implementing any of the algorithms mentioned earlier that go from the basics of defining the type of problem that one wants to solve to the analysis of results obtained after applying the designed methodology. In Figure 3-1, I show a diagram that summarizes this common workflow of activities where I identify five main stages for building up an ML model. These stages are the problem definition, the dataset constitution, the model creation, the model training, and finally the model testing. This whole process is cyclic and iterative to optimize the ML model as much as possible. It is also important to mention, as the arrows try to indicate, that the process is not just sequential but that decisions made in one of the stages will affect the other ones. Some general rules have been derived heuristically for designing the optimal architecture for each application of ML, but their design has grown to be more a sort of an art than a predictable logical process.

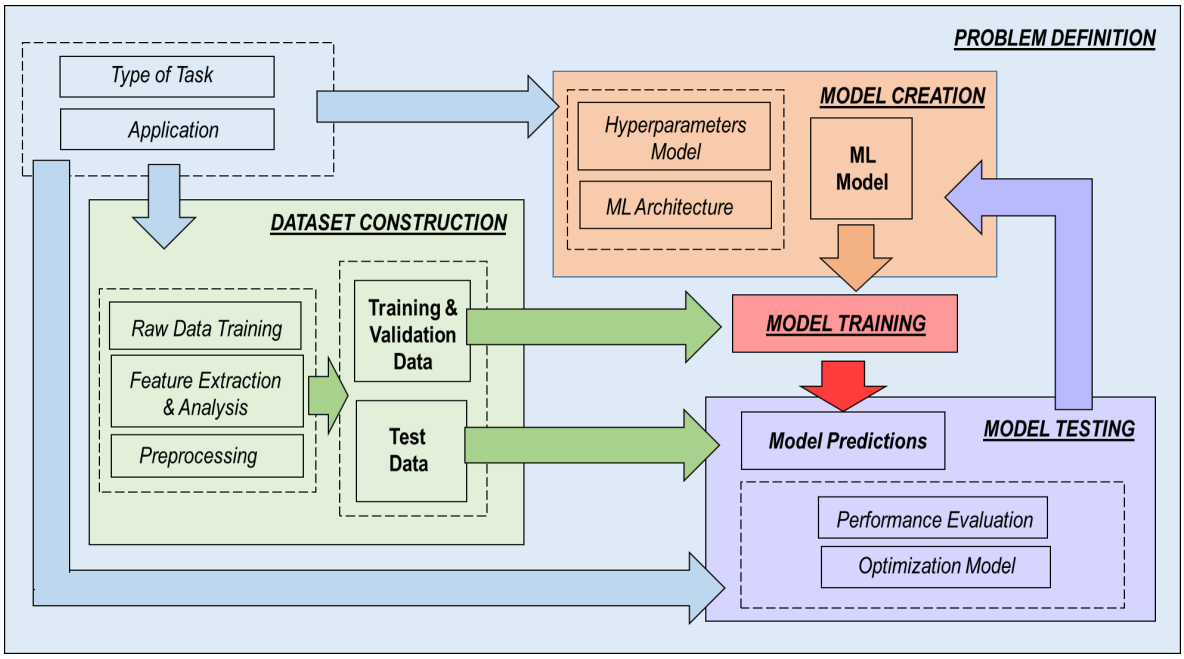

Figure 3-1: A workflow that describes graphically the common steps needed to follow for addressing any machine learning project. Five main tasks are identified. They are in order: the problem definition, the dataset construction, the model creation, the model training, and the model testing. The optimization of the produced ML model makes of this exercise an iterative process, and the decisions taken during any one of its stages end up affecting the outcome of the other steps.

Problem Definition

The first step of the workflow is to define the type of problem, question, or application that one wants to address. The nature of such a problem indicates what kinds of ML tasks are the most appropriate to solve, and it is one of the core factors to decide the path to follow in all the other four stages of the working flow. There are different classifications of these types of tasks.

The first classification has to deal mainly on whether the data we produce is labeled or not. In the case it is labeled, then the task is called supervised and is more suitable to use ANNs or SVMs to solve them. If it is not labeled, then it is an unsupervised problem where clustering is more appropriate to gather the data points based on metrics of their features or characteristics. Additionally, there is a third category where data can be labeled or not without defining the type of task. This category is called reinforcement learning, and its focus is decision making, specifically looking for maximizing the rewards after reaching a goal or simply to achieve it in an optimal way. Together the supervised, the unsupervised and the reinforcement learning constitute the three basic ML paradigms.

The second subdivision of tasks attains mainly to supervised problems and is whether one pursues to perform a classification of the data or calculate a regression using it. If the objective is to identify as best as possible, which is the label that belongs to a data point, then it is a classification task. If on the contrary, what is required is to produce a numerical output given a certain collected data, then it is a regression task.

Finally, ML has been growing into specialized subfields that tend to be defined by the type of data that is handled and the final application of the ML model. The main types of data are time series, text, images, and categorical data, and an example of an application can be to count the number of attendees to an event from a picture taken from it. This task is called 'crowd counting' and can be solved using object recognition techniques that are commonly used in the field of ML called *computer vision*. Since in its essence, this problem is just to count objects in an image, the architecture developed for solving it can be easily transferable to other branches of knowledge. In the case of this project, I use them for the analysis of gravitational wave recordings.

Dataset Definition

The accuracy that an ML model can reach relies heavily on the quantity of data that it is used to train it. For some applications, this is not a difficulty since it is possible to construct large datasets from publicly available data sources or they can be acquired through measurements, surveys, or experiments. Nevertheless, in other cases, there is a need for producing synthetic data to compensate for the scarcity of it due to limitations like confidentiality or large expenses behind acquiring every data point. Naturally, the advantages of ML are better exploited when using real data since it is less prone to have inner biases that can hinder the performance of the ML model, and because it allows the machine itself to find more interesting patterns that may not have been considered previously by theoretical models. This disadvantage of synthetic models raises the need for making them as close to reality as possible.

After the decision of whether using synthetic generated or real datasets, a selection of the features to be considered to constitute them needs to be done. These features are the variables or characteristics that are the most significant to discriminate between labeled classes in a classification task or to produce a numerical output for a regression one. This selection normally is a very time-consuming process and of a substantial impact on the efficiency of the tool built from the ML model. Typically, a first raw training is done to evaluate what is the impact of each of the features that are initially considered to be crucial for addressing the task. Then a selection of them is made manually by evaluating the existence of interdependencies or ranking their significance in the final solution. This process allows decreasing the dimensionality of the problem, increasing the efficiency and allowing to make a better assignment of the computational resources. The substantial effort in this feature selection is what has increased the popularity of Deep Learning approaches in the last years. These are a group of ML methods that are based on ANNs that have the advantage of performing the feature extraction automatically and internally in their architectures. The Deep Learning models achieve this thanks to the system of layers of the neural networks where the first layers act as feature extraction systems and the deeper ones as connectors of multipliers of the extraction made. For this reason, commonly, deeper models tend to perform better when compared to shallower ones.

Finally, the data that has been organized by features is allocated or split into three different packages or datasets that serve different purposes. The first one is the training dataset, which is used for finding the optimal parameters that constitute the ML model. The second one is the validation dataset that allows evaluating whether the built ML model present problems of overffiting. The third and final one is the testing dataset, which is a fraction of the data unseen during the training phase that helps to do a performance evaluation of the ML model and assess how well it will probably solve the task when used on new data.

Model Creation and Training

Once the problem to be addressed has been clearly defined, the next step is to determine the ML architectures (SVMs, ANNs, RFs, among others) that are more appropriate to solve it given the datasets available. This selection of a specific ML architecture comes together with another decision. Particularly, one needs to decide the values and parameters for every inner component that constitute that specific architecture. For example, for an ANN this translates in defining the number and type of output layers, the number, and type of hidden layers or the number of neurons. All this process is what is referred to as the model creation.

After creating the model, the next step is to train it. This training phase consists in feeding the model with the datasets and calculating an output, and through an iterative optimization process reduce the misfit between this output, that would be considered the theoretical result, and the data, the experimental observations. The outcome of this stage is then the values of the components of the designed architecture that best solve the ML task given the provided training data.

As it is possible to see, all the decisions that need to be made open an infinite possibility of combinations of parameters, and therefore models, that can bring a solution to our initial problem. Naturally, what is desired is to find from all those possibilities the best estimator or solution. For this, what is called a 'hyperparameters optimization ' and a model comparison is made. This process can be an infinite loop, but frequently time, computational power, and overfitting are the boundaries which define when the optimization problem needs to stop [James Bergstra, 2012].

The hyperparameters are defined as all those hidden elements that may be part or not of the ML model, but in the end, make an impact on the performance of the machine learning tool built with it [MacKay, 1999] [MacKay, 1996] . Some of the most common hyperparameters in machine learning are Batch size, Learning rate, Number of epochs, Model depth, Cost function and the Learning rate. The model optimization of such hyperparameters is one of the toughest challenges in the implementation of machine learning solutions, and it is mostly intuitive [James Bergstra, 2012]. This procedure is done through an iterative process of trial and error, where the performance evaluation of the models produced with the current settings of hyperparameters is compared with the previously chosen [MacKay, 1999] [MacKay, 1996].

Model Testing

The final stage after training the model is the model testing. The objective of this phase is to evaluate how good is the performance of the ML model in new data, to have an approximation of what would be the effectiveness of the built tool in practice. For making this evaluation, it is a metric. It is required to quantify the model performance. Depending on the final application, the type of task being solved, and the nature of the data being analyzed, there are more suitable performance metrics than others.

For binary problems, where two labels are assigned, we can translate the results of our models into correct and incorrect predictions. The two possible classes (labels) receive the name of the negative class and positive class.In this kind of tasks, the positive one represents the presence or occurrence of an event of interest, and the negative class its absence. A usual example that follows this logic is cancer detection, where having a positive result is interpreted as having the disease and the negative ones of being free of it. In this scenario there are four possible logical results that can be obtained from the comparison between the real classes in the data

and the predicted by a model. These are explained below and are commonly summarized in a confusion or error matrix like the one shown in Figure 3-2.

- **True Positive (tp)**: when the model correctly predicts the positive class.
- **True Negative (tn)**: when the model correctly predicts the negative class.
- **False Positive (fp)**: when the model incorrectly predicts the positive class.
- **False Negative (fn)**: when the model incorrectly predicts the negative class.

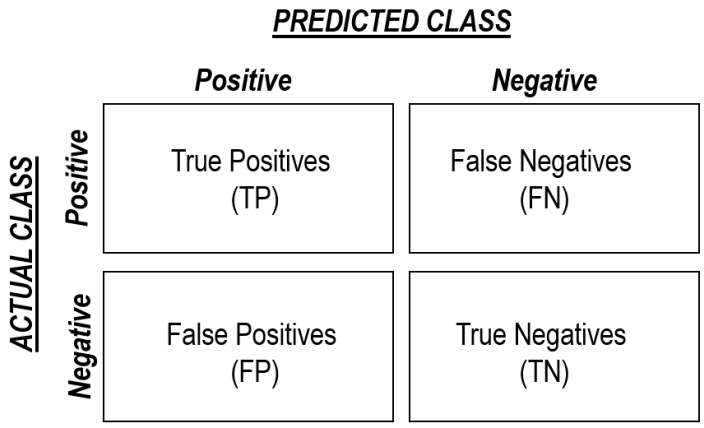

Figure 3-2: Graphical description of how a Confusion Matrix and its components for a Supervised Binary Classification application

Based on these four comparisons between the positive and the negative class, it is possible to calculate some performance metrics. I decided to use for this project three basic ones that are accuracy, precision, and recall. Accuracy is the fraction of predictions our ML model gets right, and it is expressed mathematically by Equation 3-1. Precision is the fraction of how many of the positive results are relevant. Recall is the fraction of how many of the actual positive results are not missed. The equations describing precision and recall are Equations 3-2 and 3-3. These last two metrics are more insightful measurements than the accuracy and are especially useful when having datasets with imbalanced classes.

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn} \tag{3-1}$$

$$Precision = \frac{tp}{tp + fp} \tag{3-2}$$

$$Recall = \frac{tp}{tp + fn} \tag{3-3}$$

Visual representations can be done to have a better understanding of the meaning and evolution of these performance metrics. These visualizations are shown in Figure 3-3. In Figure 3-3 [A] we have an accuracy plot, which shows a comparison of the evolution of the accuracy results in the training dataset and the validation datasets in terms of epochs (single pass or iteration through the entire dataset). When the training and validation accuracy plateau are close to each other, it means that little overfitting occurred during the training stage. In contrast, if these two curves are very far apart from each other, then the ML model is overfitted, and then only the validation accuracy is reliable to evaluate the performance. In Figure 3-3 [B] I show a Precision-Recall curve. This type of plot shows the evolution of the trade-off between precision and recall when the threshold of the ML model is modified. A high precision relates to a low false-positive rate, and a high recall relates to a low false-negative rate. The desired point is one where both of these metrics are 1, which means that the classifier is returning accurate and relevant results. In the one hand, a system with high recall but low precision returns many detections but most of its predicted labels are incorrect. In the other hand, a system with high precision, but low recall is returning few but correct detections.

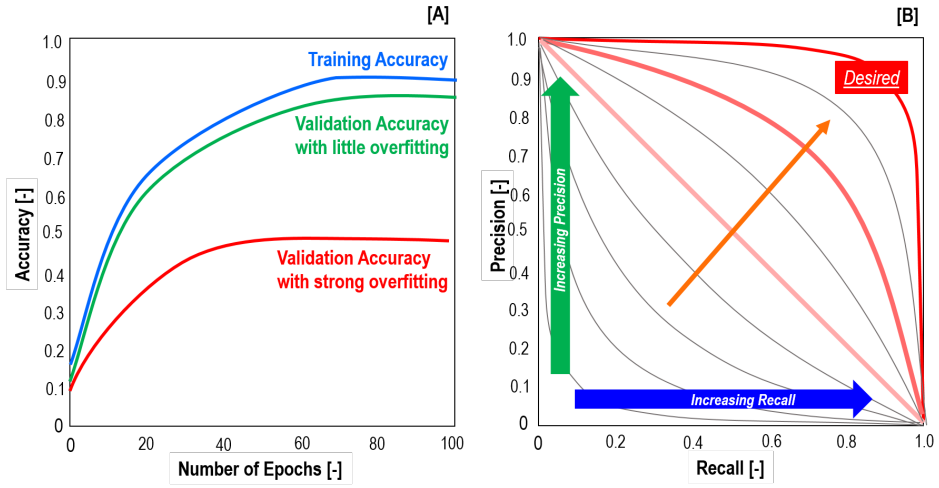

Figure 3-3: [A] Comparison between the training accuracy and the validation accuracy during the training stage of the model. When the training accuracy curve and the validation accuracy one are close to each other it is good indicator that little overfitting occurred in the training phase. [B] Comparison of possible outcomes of a precision-recall analysis during the testing stage of a model. The best outcome is obtained for both a precision and a recall equal to one. Models closer to the vertical axis have lower recall and tend to make lower number of detections. Models closer to the horizontal axis tend to have lower precision and more false alarms in their predictions.

.

3-1 Neural Networks

An Artificial Neural Network (ANNs) is an algorithm for applying machine learning in which we produce a system of interconnected layers containing neurons. These neurons produce mathematical transformations to the information transmitted to them from previous layers and transmit the result to the next layer. ANN is the ML architecture where Deep Learning is based on. Therefore, it is beneficial to understand how the general workflow that is shown

in 3-1 applies to them.

In Figure 3-4 I show the basic shape of a Multilayer Perceptron (MLP) or Fully Connected Neural Network (FCNN), which are the most common and basic ANN architecture. The visible components are the neurons represented by empty circles and their connections represented by arrows. The first layer is referred to as the input layer and is the one that first receives the data. The last one is called the output layer and is the one that delivers the result of the classification or regression. All the layers in between receive the name of hidden layers. Inside of each neuron, an operation that could be described as a linear regression occurs, and it is followed by a non-linearization performed by an activation function.

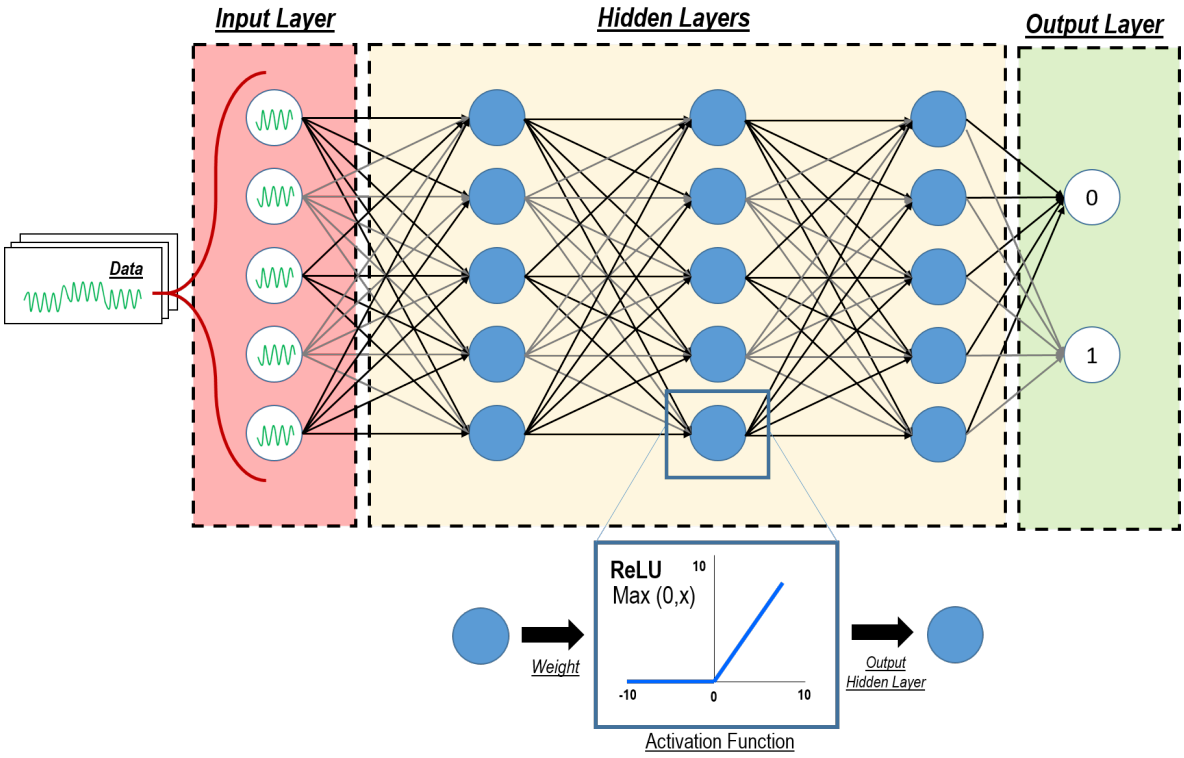

Figure 3-4: A graphical representation of a Multilayer Perceptron (MLP) the most basic type of Full-Connected Neural Network (FCNN).

In the output layer, a final result is obtained from the network which is a number between 0 and 1. This number represents a probability which allows to make a classification of the input data. A probability threshold (p) is set to define to which of the two classes the data, in a binary problem, the data belongs. If the output number is larger than the threshold, the given class is assigned. The trade-off behavior shown in Figure 3-3 [B] between precision and recall is explained by this threshold. If we choose a high threshold (level of confidence), the classifier is more rigorous with respect to the output of the network and makes fewer detections of the positive class . In the other hand, if we choose a low threshold the system can make more detection because is less rigurous, but the number of false detections increase. Commonly, a probability threshold of 0.5 is commonly used but by producing precision-recall plots it is possible to identify more suitable values for the specific application of our interest

and tune it.

3-1-1 Types of Networks

There are several types of architectures that an ANN can have. Two of the most commonly used, and that I implement in this project are the Fully Connected Neural Networks (FCNNs) and the Convolutional Neural Networks (CNNs).

Fully Connected Neural Networks (FCNNs)

A fully connected layer is one where all its neurons are connected with all the neurons belonging to its previous and next layers. This connections are constitute by a weight that allows measuring its activation. These weights can be regarded as an indicator of the importance that one feature or neuron has for assigning the ultimate output of the neural network. A FCNN is then a neural network which layers are all fully connected.

Convolutional Neural Networks (CNNs)

In Figure 3-5 I show a representation of how a CNN is usually constituted. Two main parts conform this architecture. The first part constituted by a feature extraction system made of convolutional layers, and the second part formed by a fully connected layer that assembles the results of the feature extraction system to produce the final output of the network. This type of architecture is very handy for the analysis of images and bidimensional information [Alex Krizhevsky, 2012] [Daniel Maturana, 2015].

Inside the feature extraction system, there are three sequential processes. The first one is the production of a feature map, which is the result of a convolution operation between the information coming to the neurons and filters. Standard sizes of these filters are 3x3, 5x5, and 7x7. The second part is a pooling done over the resulting feature maps reduce the number of features to be analyzed in the next layer to only the most significant information. Finally, the pooled feature map is transmitted to the next layer, and the process begins once again until reaching the fully connected layer.

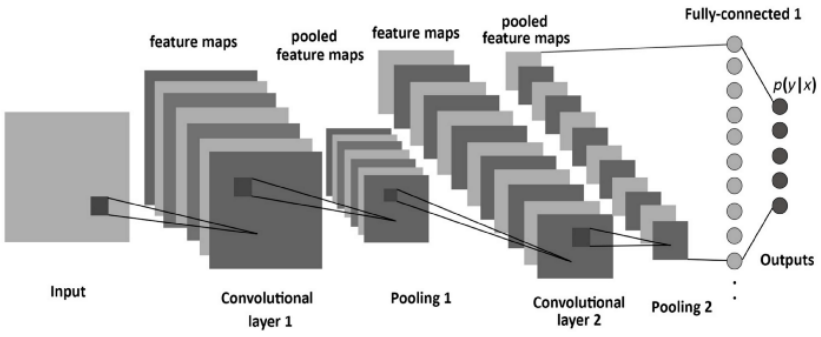

Figure 3-5: Graphical representation explaining how a Convolutional Neural Network works (CNN). The system receives data represented as a matrix (or vector) as an input and performs convolutions with filters that result in another matrices called feature maps. These feature maps are reduced by pooling layers to their most representative components, and are used as input for subsequent convolutions. In the end, the system gathers the information produced as a vector that is processes by a FCNN that gives a final input. Modified from [Steve Lawrence, 1997].

3-1-2 Neural Network Training and Testing

The training phase is the stage where are calculated the parameters that constitute the functions inside of each of the neurons of the ANN. The training algorithm that allows this in a neural network is called Backpropagation, and a visual representation of it can be seen in Figure 3-6. The logic of how it operates is similar to a Gradient Descent search. First, all the values of the parameters inside the neuron, and the weights that connect them are randomly initialized. The data is feed in batches to the input layers, and a first forward pass is calculated all the way until the output layer. Then the misfit error between the forward pass results and the actual data is obtained through a cost function of choice. Afterward, in a backward pass, all the error is propagated back by minimizing the gradients with respect to their fitting parameters in every neuron. Once this process is done with the whole dataset, an iteration (epoch) has finished. This process is repeated until the desired number of epochs are completed.

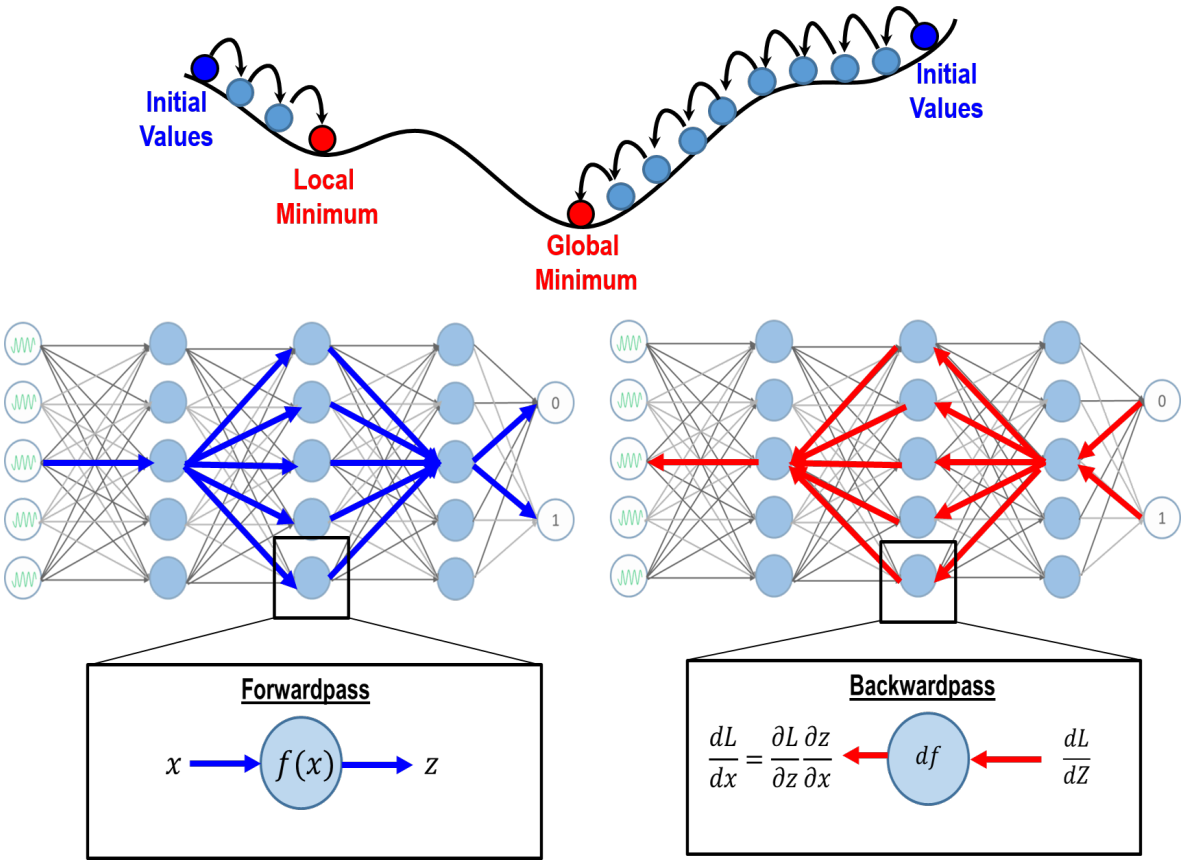

Figure 3-6: A scheme that depicts the backpropagation method, which is the core process taking place during the training phase. It works very similarly to a gradient descent optimum search. In a first forward pass, represented by the blue arrows, is done using random initial values until calculating a final output. This output is compared with the real data, and the difference between the two is expressed in terms of a metric or with a function. This error is then backpropagated, red arrows, to fit the parameters inside each of the neurons to make more accurate predictions. Then the system uses these new parameters and produces a new cycle of forward and backward passes until reaching a global optimum or willingly stopping the process.

Due to the random initialization, every training process tends to end up with a different neural network. Not choosing the right hyperparameters, especially the learning rate, can have the effect of getting stuck in local minima in the optimum search and not reaching the best model possible (global minimum). For this reason, it is advisable to produce several runnings of the training process for the same architecture, and in the end, select the network that presents the best performance.

Chapter 4

Methodology

4-1 Overview Problem

As discussed in the introduction at the moment of designing a Low Latency Detector for LISA, several difficulties need to be overcome. These are mainly to discriminate among responses of overlapping sources and dealing with situations of high levels of noise. Besides, other goals like deciphering the information of the generating source encoded in the gravitational wave are very valuable. To use deep learning for achieving these goals, it is necessary to translate these necessities into ML tasks. For this purpose, I have identified four major stages that are required for posing the problem and constructing the deep learning models. These stages are identified below and in Figure 4-1 I state the logical order for solving them.

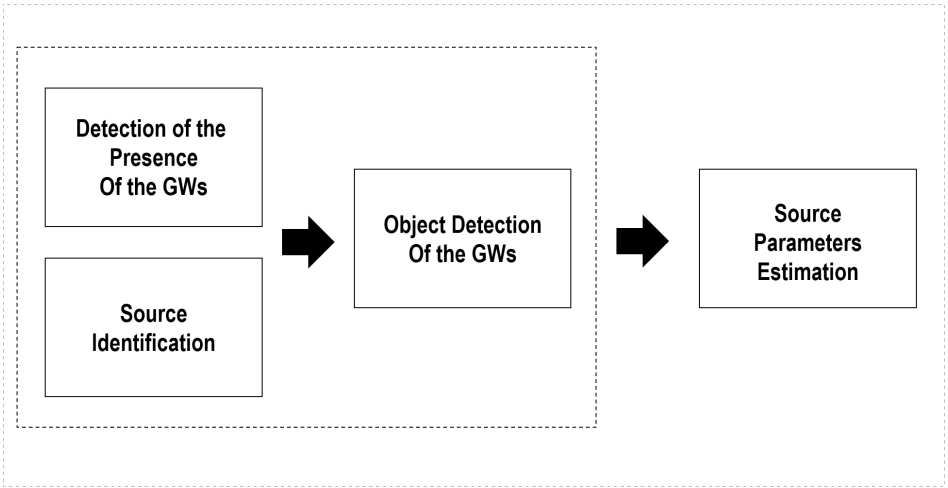

Figure 4-1: Scheme of flow of components for achieving the parameter estimation of gravitational waves

- **Detection of the Presence of Gravitational Waves**: It refers to developing an algorithm which can detect if, in a window of data, a gravitational wave is present or

not. This can be demanding due to the various sources of noise that can opaque the signals recorded.

- **Object Localization of the Gravitational Waves**: Once it has been confirmed that there is the presence of a gravitational wave in the recording, the next question to ask would be precisely where and when does it occur. This means in which given time and frequency range is the gravitational wave detected. One of the advantages of being able to achieve this goal is that it would be possible to discriminate between overlapping sources and quantifying them.

- **Source Identification**: It refers to identify what specific type of source is the one that has been detected. A possibility to solve this problem with machine learning would be to build up a specialized network for each different type of astronomical source. This is done by feeding during the training phase with the specific type of astronomical source searched. This way, multiple networks can analyze in parallel the same recordings and differentiate between the various gravitational waves present.

- **Source Parameter Estimation**: It refers to estimating the most probable parameters which could have generated the detected gravitational wave. Since gravitational waves parameters keep a deep relationship to the nature of the sources which produced them, this becomes an important question to solve. Even though, due to time and computational constrictions, this task is not addressed in this research.

4-2 Proposed Methodology for Gravitational Waves Detection

After having discussed in the previous section the main tasks of the analysis of a recorded gravitational wave, I propose the use of an algorithm consisting of a sliding window with a deep neural network as its classification mechanism. This has the potential of being an optimal solution for detecting gravitational waves under very noisy conditions. This approach consists of segmenting the information in smaller pieces that can be analyzed by the network to identify whether a gravitational wave is present or absent in it. A deep learning model is a right candidate for this task since it can perform quick classifications with high accuracy when the model structure has been optimized and fed with enough data. I have decided that for performing the detection of gravitational waves I will formulate the problem as a binary supervised classification task. . This means that the performance evaluation metrics and model testing explained in Chapter 3 are applicable for the models built in this project. In order to solve the classification task and to build up the LLD, I have defined three main steps. The sequence of these steps are shown in Figure 4-2, and explained in more detail in the following section. Finally, the definition I used for the positive and the negative classes concerning to this binary problem and to be assigned as labels in the datasets to built are the following ones:

$$class = \begin{cases} 0 & \text{if it is a recording with only noise } (\textit{Negative class}) \\ 1 & \text{if it is a recording with a gravitational wave present } (\textit{Positive class}) \end{cases} \tag{4-1}$$

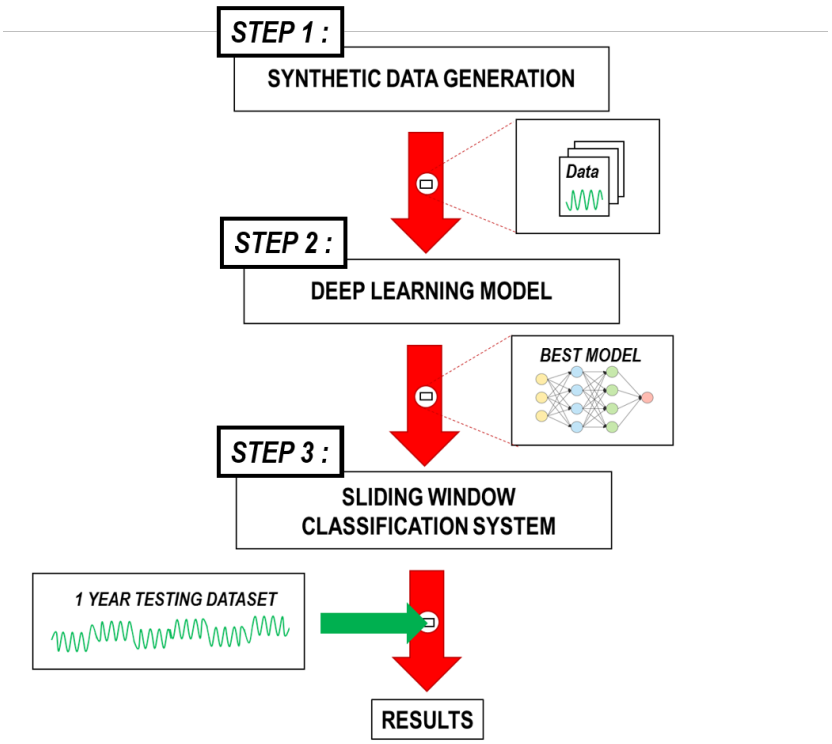

Figure 4-2: Steps of the Methodology for creating and testing the Low Latency Detector (LLD)

4-2-1 Synthetic Data Generation

The accuracy reach by a deep learning model relies heavily on the quantity of information that it is used to train it. I decided on using synthetically generated data for building and training the neural networks that are employed by the Low Latency Detector. The reason is that LISA is a project that is still in design. Therefore, there are no real measurements that can be provided to the network to be trained. Moreover, the ranges of frequencies recorded as catalogs by the existing ground-based radar-like LIGO belong to a different frequency band than the one LISA will measure.

Due to the complexity of the generation of these gravitational waves, I divided the generation of synthetic data and the training of the deep learning models in two stages. In the first part, I created simple sinusoidal waves that resemble the nature of one of the most commonly recorded gravitational wave sources that are the galactic binaries. These recordings, for example, do not consider the effects of LISA orbit in the signal. Therefore, I refer to this type of GWs as simplistic ones. The advantage of using these simple GWs is that I can reduce the degrees of freedom that are affecting the optimization task of the neural networks. This reduces the time of computation, the volumes of information and also allows to focus on tunning more hyperparameters. In the second part, I apply all the lessons learned from the networks constructed for these simple sinusoidal into two types of gravitational waves sources: realistic Galactic Binaries and Merging Black-hole Chirping Binaries.

Acknowledging the fact that LISA will record various types of events that can go in terms of amplitude from being very strong to the very week when compared to the noise, a mea-

surement needs to be used to skip having an unbalanced training dataset. For this reason, I include recordings that represent all these scenarios of signal strength. I do that by generating recordings with a different signal-to-noise ratio (SNR).The definition of SNR used in this project is shown in detail in Equation 4-2 and the type of noise used for building the datasets is Gaussian Noise.

$$SNR = \frac{Maximum\,Amplitude\,Signal}{Maximum\,Amplitude\,Noise} \tag{4-2}$$

Finally, as an additional aspect of this research, I decided to represent the data in the recordings in different domains. To be more specific, I did it in three: the time-domain, the frequency-domain, and the time-frequency domain. The reason behind of exploring these representations in different domains are three: as an attempt to help the neural networks to overcome the difficulties of overlapping sources, to search possible features that are more recognizable in terms of frequency content that can boost the efficiency of the neural networks, and because most of the research done about the use of deep learning on gravitational waves has been done only with waveforms or time-domain representations. The comparison of these different representations is schematically summarized in Figure 4-3

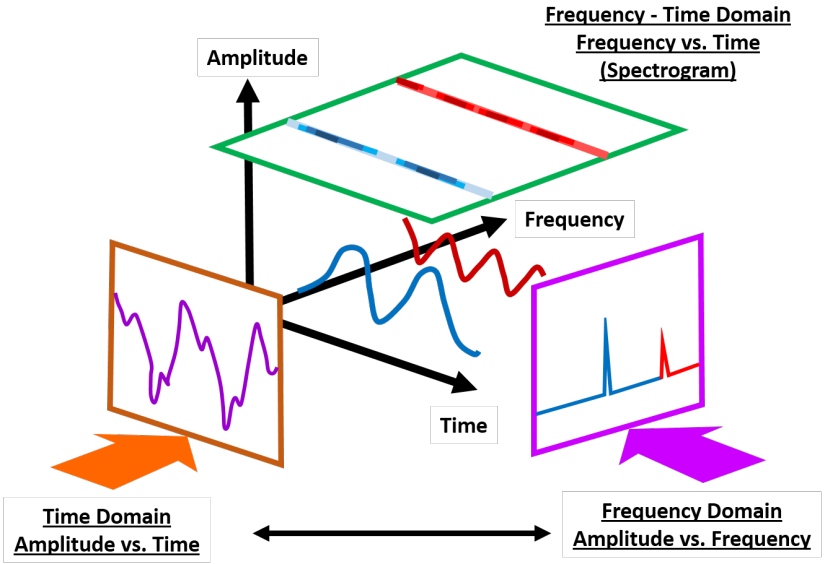

Figure 4-3: Different domains in which the data generated for the project was represented. In the spectrogram the quantity depicted is Power Spectral Density (PSD)

Time-Domain

For generating the first type of Galactic Binaries, the simplistic ones I made use of the expression shown in Equation 4-3. The parameters that I change to produce the different recordings were the maximum amplitude of the signal that oscillated between 10^{-22} and 10^{-20} , and the frequency between $0.1\,mHz$ and $100\,mHz$. These are the typical amplitude and frequency ranges expected from the gravitational waves expected for LISA as it has being shown before in Figure 2-2. In terms of the phase lag, it was assumed to be zero. Finally, it is important to mention that the total time used for these recordings was of 10 hours.

$$x(t) = (Max.\,Amplitude\,Signal) * Cos(2\pi ft + \phi) + Gaussian\,Noise \tag{4-3}$$

In Figure 4-4 I show the time-domain representation (waveform) of three examples of sinusoidal waves with different signal-to-noise that constitute one of the training datasets for the simplistic Galactic Binaries. The first row corresponds to a noisy recording where no gravitational wave is present; the second one is one with a low signal-to-noise ratio (SNR=0.5) and the last one with a high signal-to-noise ratio (SNR=100).

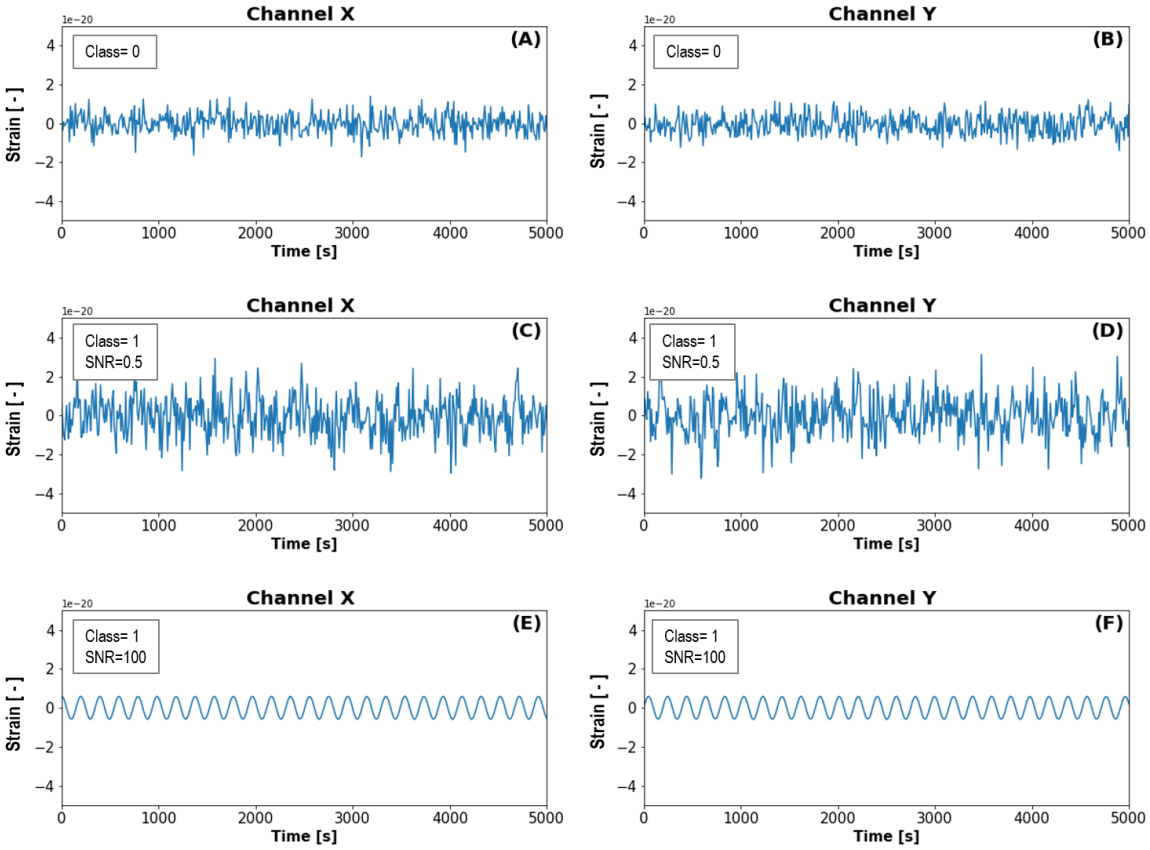

Figure 4-4: Time-domain representation for (A) and (B) Channel X and Y of a sample classified as noise (Class=0), **(C) and (D)** Channel X and Y of a sample classified as signal (Class=1) with a signal-to-noise ratio of 0.5, **(E) and (F)** Channel X and Y of a sample classified as signal (Class=1) with a signal-to-noise ratio of 100

Realistic Galactic Binaries

The synthetic data for this more realistic type of GWs was generated by using the MATLAB codes created for the project " Source Agnostic Exploration of Gravitational Wave recordings" [Edwards, 2018]. These files produce synthetic TDI 1.0 recordings for Galactic Binaries and is based in the methodology proposed in [Andrzej Krolak and Vallisneri, 2008a]. The input data for the code are the source parameters shown in Table 2-1 and Table 2-2 The methodology consist in performing transformations like rotations of the reference frameworks of the initial amplitudes of the polarizations (h_{plus} and h_{cross}) originally generated by the GB source. The first rotation expresses the data in the Solar-System Barycentric (SSB) source frame and a

second one into the LISA frame.Afterwards, the Doppler responses of the arms and then the modulation of the three Michleson interferometers that act in each of the corners of the LISA constellation. Until this point a TDI 1.0 signal is generated which is later approximated by using the *"Long Wavelength Approximation"*. For a more detailed description of the methodology, please refer to [Andrzej Krolak and Vallisneri, 2008a].In Figure 4-5 I show an example of a waveform generated using this procedure, it is possible to see that in contrast to Figure 4-4 the modulations due to LISA orbits are included.

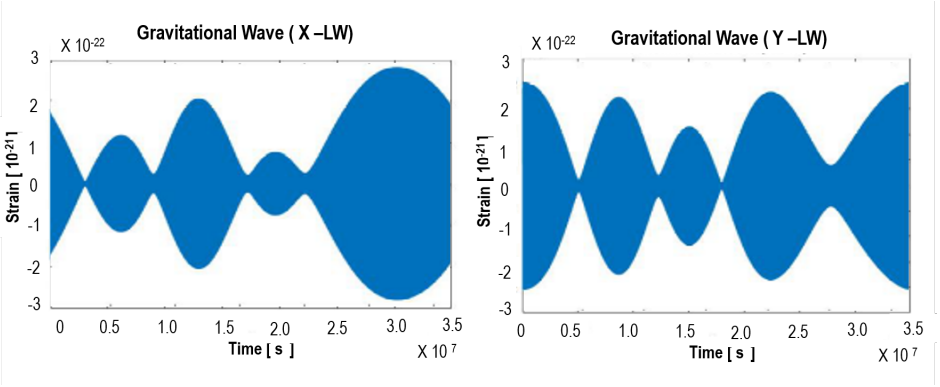

Figure 4-5: Time-domain representation of a galactic binary

Merging Black-hole Chirping Binaries

The Merging Black-hole Chirping Binary (MBCB) is the second type of gravitational wave recordings used in the project, they are more complex in nature and also the most promising ones to be identified by the Low-Latency Detector. For producing them, I used a mathematical simplification of their behavior, for synthetic data generation purposes. The source parameters used in this case are not exactly the ones stated in Table 2-3, but instead the same ranges of amplitude and frequencies as the ones used for generating the simplistic Galactic Binaries. In Figure 4-6 I show the time-domain representation of a typical MBCB. ,

$$x(t) = Max.Amplitude(t) * Sin(2\pi f(t)t + \phi) + GaussianNoise \tag{4-4}$$

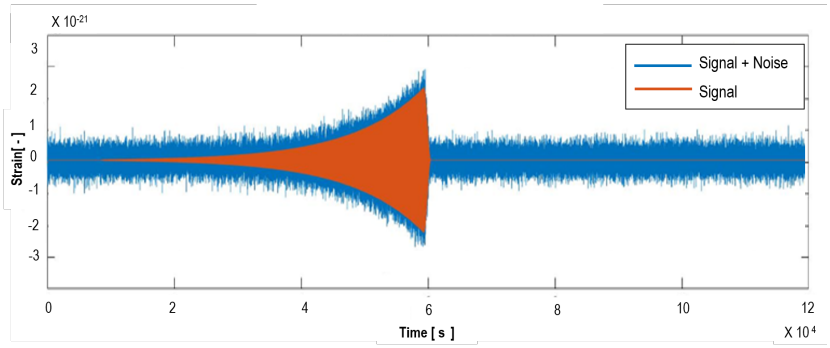

Figure 4-6: Time domain representation of a Merging Blackhole Binary. The noise-free signal of the MBHB is highlighted with an orange colour, whilst the complete signal considering the noise is represented in blue.

Frequency-Domain

The frequency-domain representation was obtained from the time-domain gravitational wave recordings produced in the previous step by calculating the Power Spectral Density (PSD) via a Fourier Transformation. The sampling frequency used was 0.1 Hz, which satisfies Nyquist sampling theorem for not having aliasing problems. It is ten times higher than the maximum frequency used for generating the datasets. For monochromatic waves, like the ones represented by simple sinusoidal and the Galactic binaries, we would expect a peak in of the power spectrum around the main frequency of such a wave. This is what is shown in Figure 4-7, where the frequency-domain representation of the recordings previously shown in Figure 4-4 is displayed.

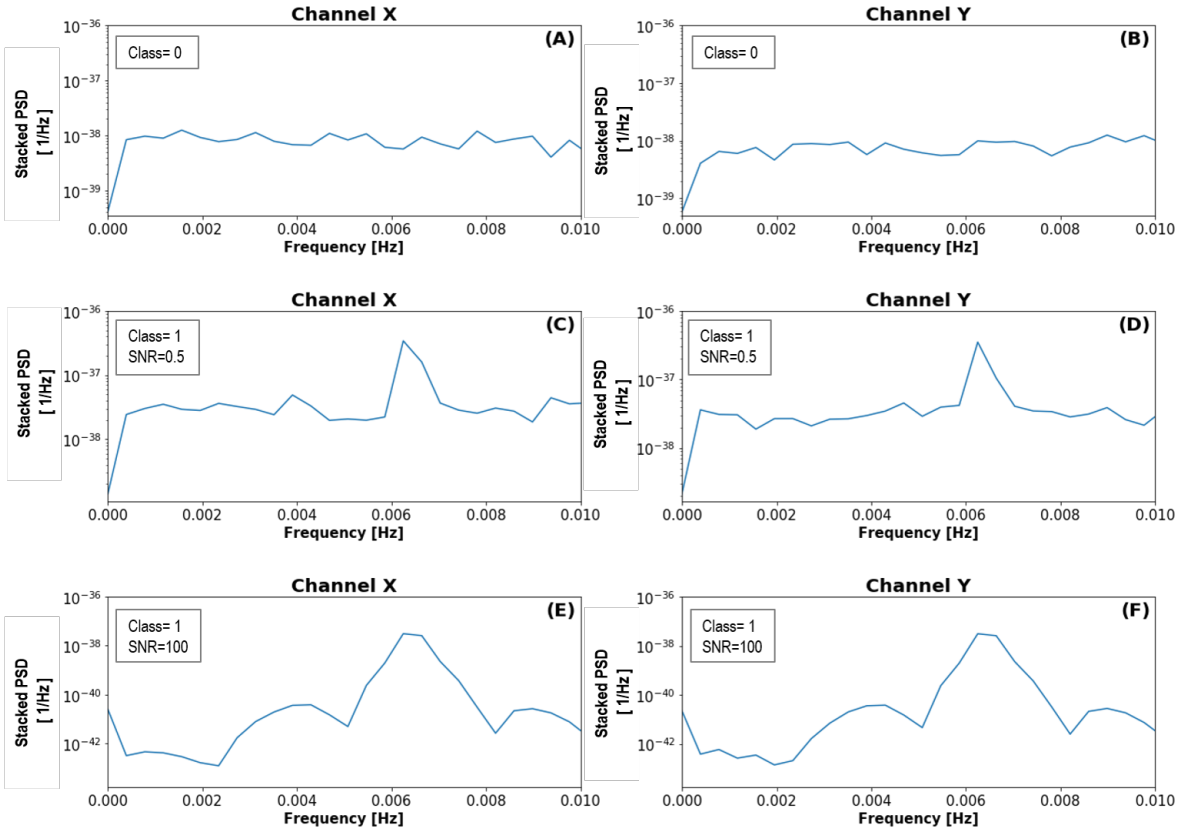

Figure 4-7: Frequency-domain representation for **(A) and (B)** Channel X and Y of a sample classified as noise (Class=0), **(C) and (D)** Channel X and Y of a sample classified as signal (Class=1) with a signal-to-noise ratio of 0.5, **(E) and (F)** Channel X and Y of a sample classified as signal (Class=1) with a signal-to-noise ratio of 100

Time-Frequency Domain

The third and final domain representation I used is the time-frequency representation. In my project, I made use of spectrograms, which were calculated using the Short-Time Fourier Transformation (STFT). One of the main advantages of including spectrograms as part of the ways of representing the recordings is that it because it is a two-dimensional instead of a one-dimensional representation of the information it allows to exploit all the benefits of

using convolutional layers in the deep learning models. The main disadvantage is that when using an STFT, there is a trade-off between the frequency and the time resolution. In these terms, I tried to maximize the frequency resolution to make it easier to differentiate between overlapping sources with similar frequency content, while maintaining enough time-resolution for later guaranteeing the building of deep enough neural networks.

The duration of the recordings used for the realistic GWs was determining for deciding the size of the window used for the Sliding Window Algorithm. This size is not arbitrary but depends on the design of the deep learning model used once the data is transformed from the time domain to the frequency domain, given a sampling frequency of 0.1. Because there is a trade-off between the time resolution and the frequency resolution a different shape of matrix (spectrogram) will result after performing the STFT transformation. Deep learning models that use convolutional neural network architecture tend to decrease the size of their input data after every layer of their architecture. This means that the larger the matrix, the deeper the model. In the case of the spectrogram the vertical dimension of the matrix is the frequency axis and the horizontal the time. Therefore, depending on the size of the bins in it a deeper or a shallower DL model can be produced for their training.

In Figure 4-8 I show the respective spectrogram for the simplistic Galatic Binaries shown in Figure 4-4. Moreover, In Figure 4-9 I show the representation of the spectrogram for three different recordings of realistic Galactic Binaries and MBCBs with different SNR.

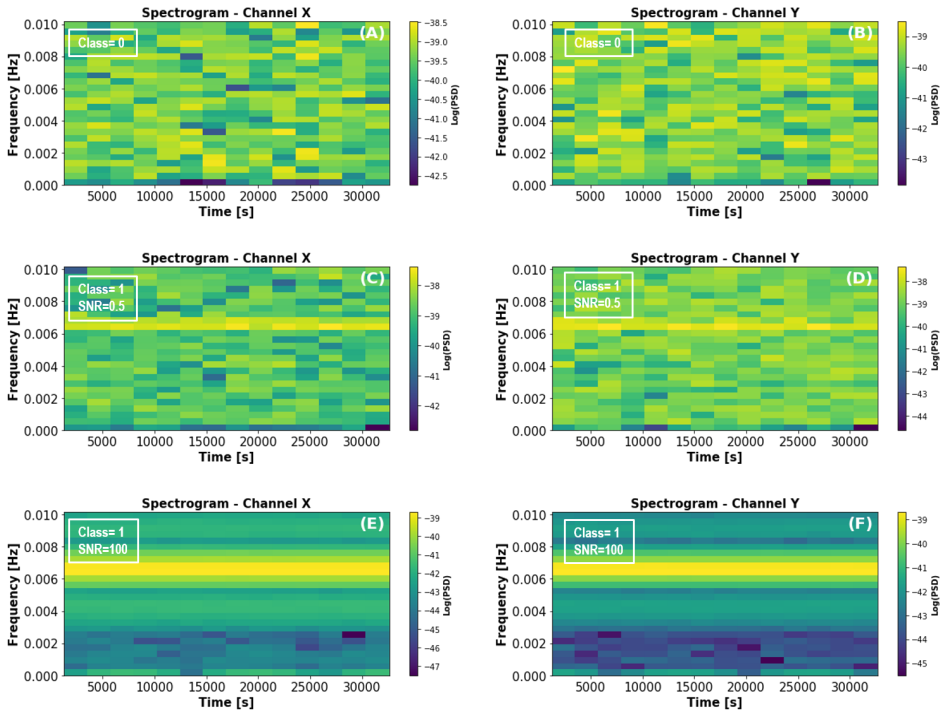

Figure 4-8: Time-Frequency-domain (Spectrogram) representation for synthetic simplistic sinusoidal waves recordings. In **(A) and (B)** Channel X and Y of a recording labeled as noise (Class=0), in **(C) and (D)** Channel X and Y of a recording labeled as signal (Class=1) with a signal-to-noise ratio of 0.5, and in **(E) and (F)** Channel X and Y of a recording labeled as signal (Class=1) with a signal-to-noise ratio of 100

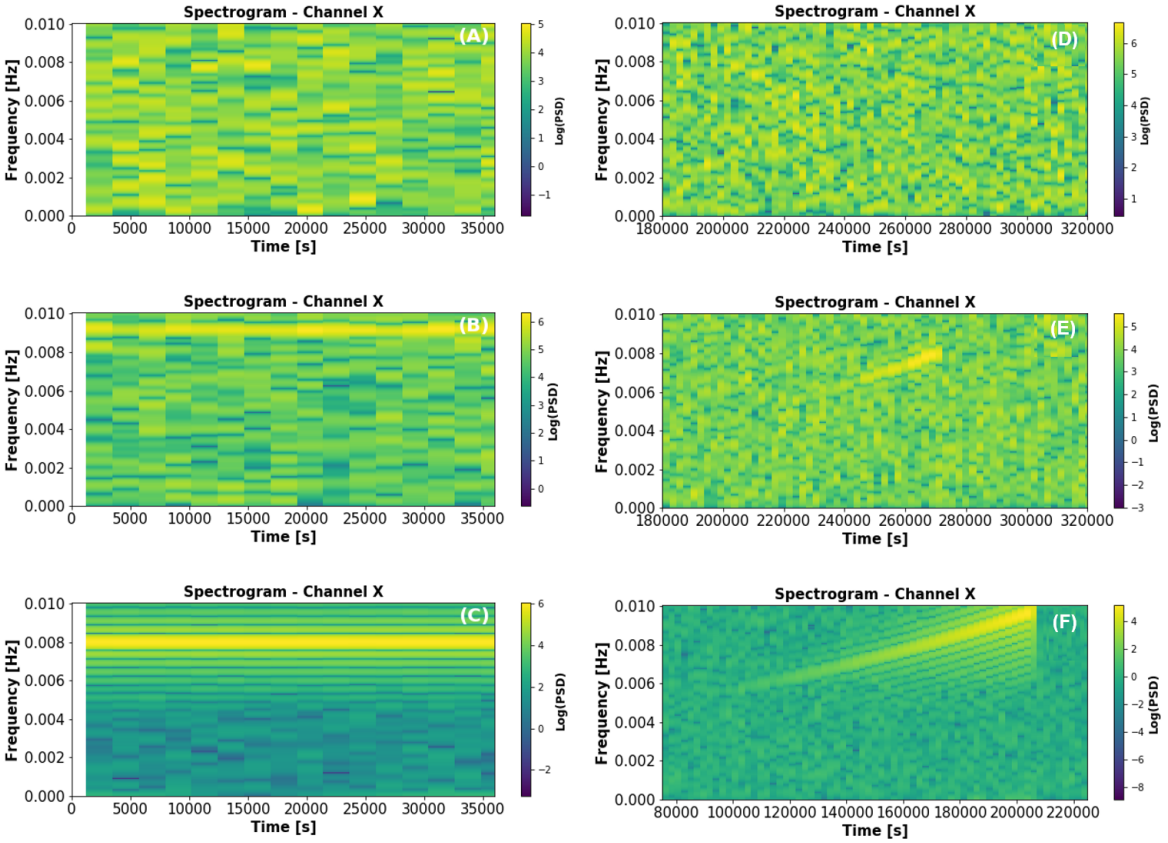

Figure 4-9: Time-Frequency-domain (Spectrogram) representation for synthetic white dwarf galactic binaries recordings. In **(A)** a recording labeled as noise (Class=0), in **(B)** a recording labeled as signal (Class=1) with a signal-to-noise ratio of 0.5, and in **(C)** a recording labeled as signal (Class=1) with a signal-to-noise ratio of 100. In **(D)** a recording labeled as noise (Class=0), in **(E)** a recording labeled as signal (Class=1) with a signal-to-noise ratio of 0.5, and in **(F)** a recording labeled as signal (Class=1) with a signal-to-noise ratio of 100

4-2-2 Dataset Constitution

To train a robust deep learning model, which can be accurate enough for the task of detecting the gravitational waves under very noisy conditions, it is necessary to produce balanced datasets. This is very important, especially because the data I am working with is not experimental but synthetic generated data.

For satisfying these requirements, different datasets with the different number of recordings were produced. This is particularly useful to assess the effect of several hyperparameters during the performance evaluation of the deep neural networks. The number of recordings used for generating them were of 1000, 2000, 5000, 10 000, and 20 000 wave recordings. In Figure 4-11 I show a graphical representation of how a typical dataset is constituted for the project.To make sure that these proportions were respected during the dataset creation, random uniform distributions of the different source parameters were used in the code.

As a general rule, the constitution of the datasets is of 50 % noise recordings (class 0) and 50 % with the presence of gravitational waves (class 1). For this last class, eight different levels

of signal-to-noise ratio were used (0.1,0.25,0.5,1,5,10,50 and 100). This translates for example that in the case of having a signal to noise ratio that is of 10 the maximum amplitude of the signal is ten times bigger than the maximum amplitude of the noise, being a case where we would expect to see the signal in the different representations.

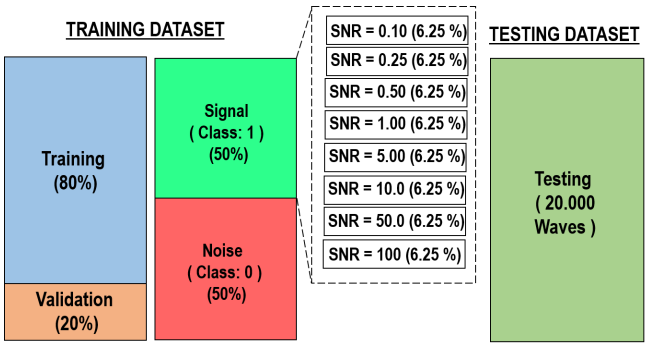

Figure 4-10: Schematic explanation of the percentages associated to the different components that constitute the datasets

In the training phase there is a series of random values that are used for initializing the fitting parameters of the neurons of the ANNs and then performing the training of backpropagation explained in Chapter 3. This random initialization makes that for the same architecture and the same data completely different models with difference performance are produce. For this reason and for been able of stabilishing comparison among different architectures I decided to produce plots like the one shown in Figure 4-11. There I show that I have trained a given number of models, 15 in total, and I have performed a evaluation using accuracy of their classification results in a testing dataset. The accuracies are discriminated by the levels of noise of the recording. In theory, a higher level of noise will make the network to have a lower accuracy in their prediction. Then from these different models, I calculate the average of the accuracy result per SNR to produce a "mean curve". This mean accuracy of the architecture is highlighted by a red line and is the criteria I used to evaluate the performance of this trained architecture.

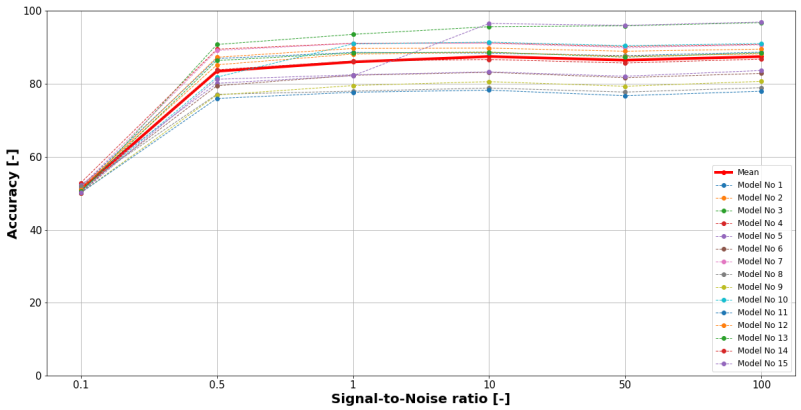

Figure 4-11: Schematic explanation of averaging the SNR results of accuracy

4-2-3 Deep Learning Model

The Deep Learning model is the responsible of performing the classification and indicating whether we have or no the presence of a gravitational wave in the segment of time that has been analyzed during the sliding window search. It consists in a series of layers of neurons that split the recording information and tries to find the most important features and patterns in it. This allows the model to differentiate the signals of interest from the areas of the recording that contain noise. The best model architectures used in this project are summarized in Appendix A and an example is shown in Figure 4-12. For producing and training these models, I used KERAS sequential models trained with a Tensorflow back-end. For the classification, I used a threshold of 0.5 that I then optimized when implementing the LLD by doing a precision-recall analysis. Lastly, I summarize in the chart shown in Figure 4-13 the duration and shapes of the input data I used for training the networks.

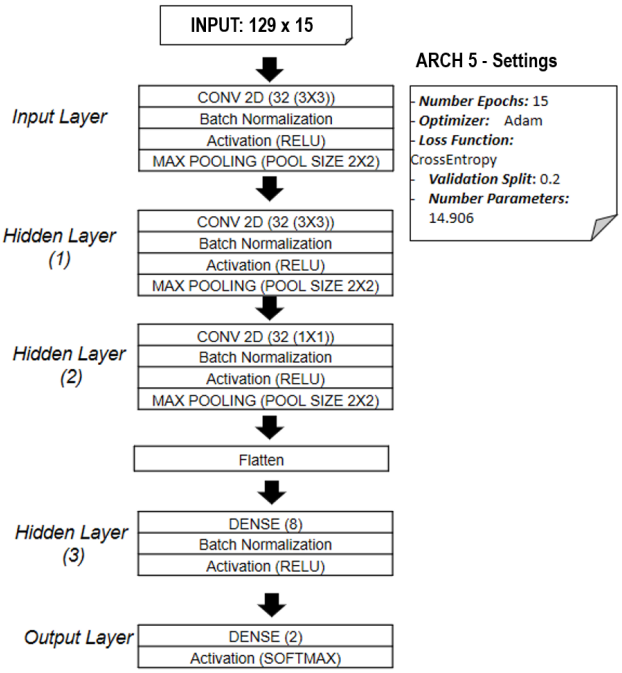

Figure 4-12: Example of the summary of one of the neural networks constructed for the Spectrogram representation

Type GW source		Duration in Time (S)	Sampling Frequency (Hz)	Input Time Domain (Samples)	Input Frequency Domain (Bins Frequency)	Input Frequency Domain (Bins Frequency x Bins Time)
Galactic Binaries	Simplistic	36000	0.1	3600	129	129 x 15
	Realistic	432000	0.1	43200	410	410 x 192
Merging Blackhole Binaries		432000	0.1	43200	410	410 x 192

Figure 4-13: Summary of the input data information for the different models

4-2-4 Sliding Window Classification

The third and last step of the methodology is to implement the sliding window algorithm with the best models found after the hyper-parameter optimization process. For testing this algorithm, I use recordings of one-year duration like the one shown in Figure 4-14. These recordings have been synthetically produced, and they contain the presence of multiple MBCBs with the presence of several MBCBs. I show in Figure 4-15 a graphical representation of how this sliding detector would perform the identification of a gravitational wave. In the figure, it is possible to observe a high signal-to-noise-ratio MBCB recording in red, the added Gaussian noise is depicted in green, The blue window represents the segment of time that is analyzed each time-step by the neural-network.

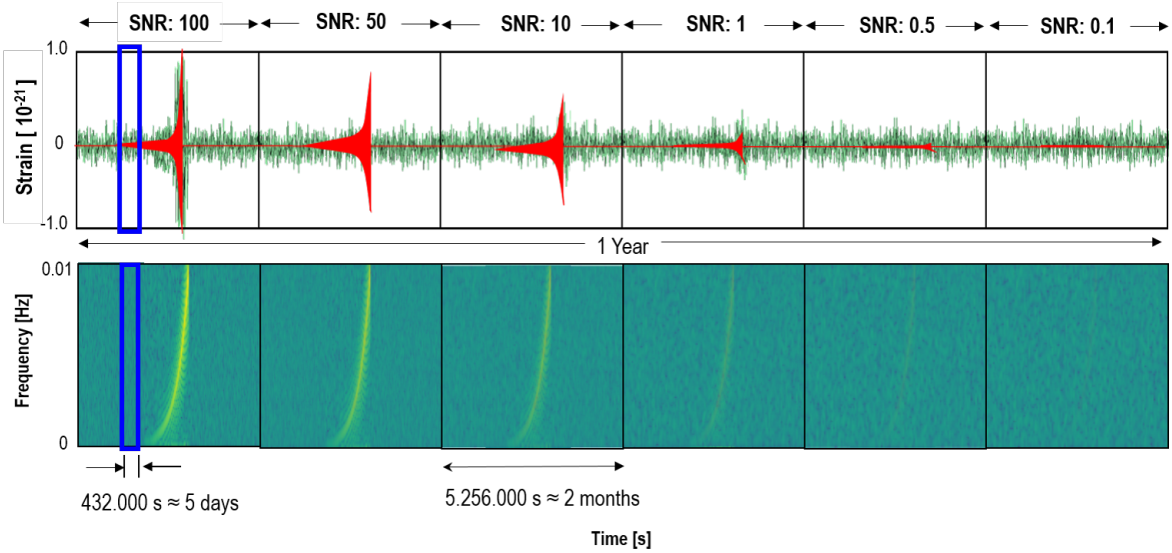

Figure 4-14: Schematic representation of the sliding window algorithm performing the classification on a recording of a Merging Black-hole Chirping Binary with a SNR of 100

In the case of the simplistic Galactic Binaries the duration of the windows is 36.000 s (10 hours) while for the realistic GWs I used 432.000 s (5 days). These windows sizes correspond to the size of the recordings used for training the data as it was shown in the chart of Figure 4-13. The time-lag used for having the best results was found to be 2240 s which corresponds to the highest time-resolution achieved after producing the spectrograms from the time-domain recordings.

The sliding window performs a classification of the segment of the one-year recording that corresponds to the duration of the window size for the corresponding type of source. This data covered by the window is used as an input for the trained DL that will give an output of a number between 0 and 1. Then, the output is compared to the probability threshold chosen. In my case I decided to use a threshold of 0.5. Finally, this result is located in the average time of the time interval the window is covering, and the next classification is done by sliding to the next time segment. Once the whole detection process is done, the result looks very similar to the lower corner of Figure 4-15 where it is highlighted the position of the MBCB in one year-long recording.

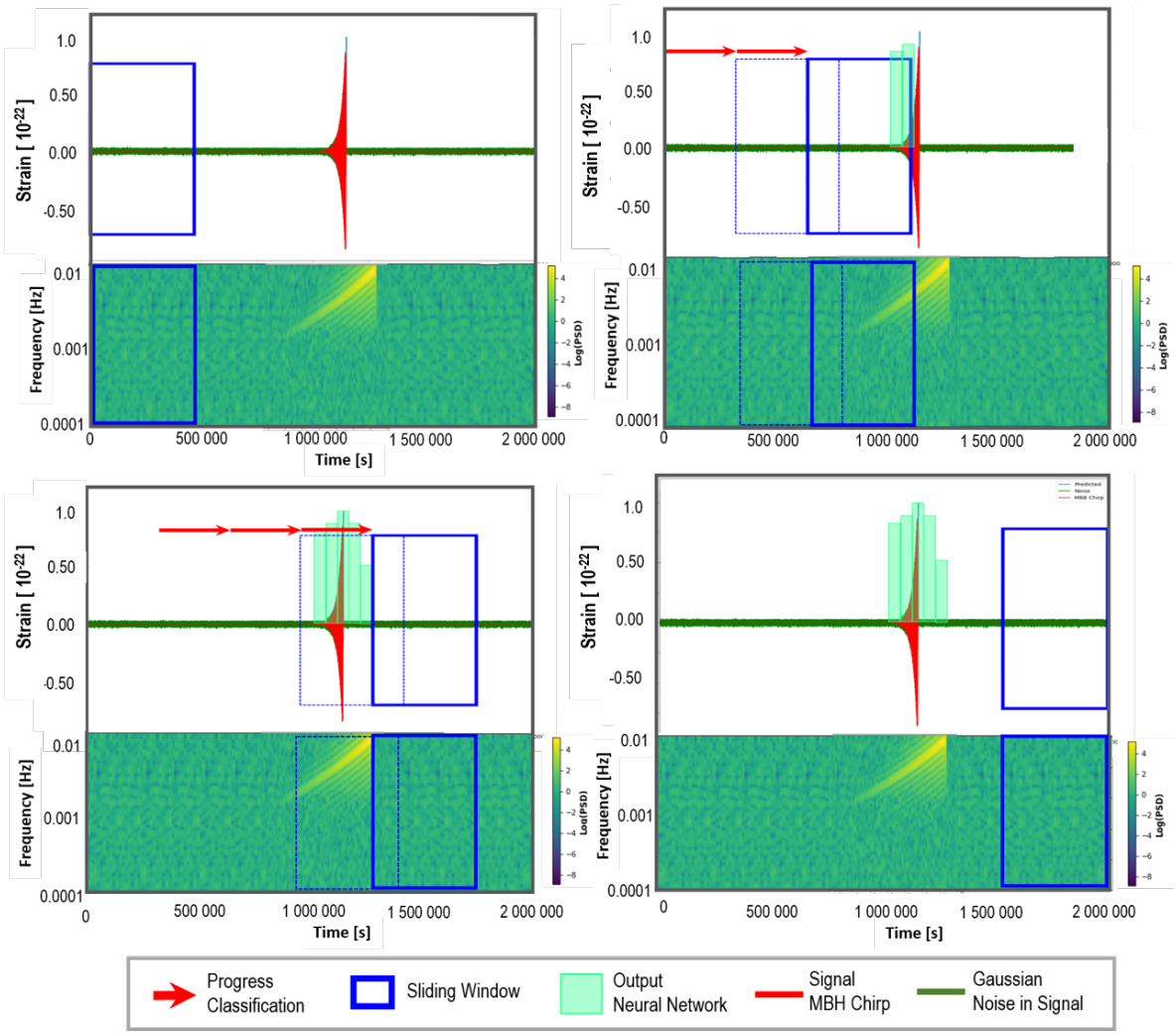

Figure 4-15: Schematic representation of the sliding window algorithm performing the classification on a recording of a Merging Black-hole Chirping Binary with a SNR of 100

Chapter 5

Data Analysis

5-1 Preprocessing Datasets

As it was explained in Chapter 3 and Chapter 4, a preprocessing of the data is needed to go from the first step of the methodology (the synthetic data generation) to the second one (training the deep learning model). This preprocessing is done to boost the efficiency of the calculations for the ANNs and allows us to have more robust models. A common preprocessing step, when working with neural networks, and the one I use in this project, is to normalize the data to be used for training and testing. This normalization is very handy when dealing with GW recordings because the amplitude of the waves is of the order of 10^{-20}, and when calculating the PSD of the data can reach orders of magnitude of 10^{-38}. This way, I save the computational cost needed for storing the information and use it better during the most consuming phase of the methodology, which is the training of the deep learning model. Additionally, the normalization helps the system during the training phase to find faster the global minima at error surface [MacKay, 1996].

With regards to this project, I perform the normalization of the datasets in two steps. The first one is to take the absolute value of the wave recordings. The second one, is to find the peak absolute amplitude in the recordings in their different representations (time, frequency or time-frequency domain), and normalize them by using Equations 5-1, 5-2 and 5-3. The result of this operation is stored and used as the input for the neural networks.

$$Normalized\,signal = \frac{Amplitude\,current\,recording}{Maximum\,Amplitude\,Dataset} \tag{5-1}$$

$$Normalized\,power = log\left(\frac{PSD\,current\,signal}{Maximum\,PSD\,of\,the\,dataset}\right) \tag{5-2}$$

$$Normalized\,power = \left(\frac{Cummulative\,PSD\,current\,signal}{Maximum\,Cummulative\,PSD\,of\,the\,dataset}\right) \tag{5-3}$$

5-2 Deep Learning Model

One of the main goals of this research is to analyze the impact of the choice of different types of ANN architectures, and representing the wave recordings in different domains in the task of detecting gravitational waves. To make a systematic analysis, I establish a classification to label the ANN models that are a product of the different combinations of these choices. This classification is summarized in Table 5-1. As it is possible to see I analyze five main categories of models, using for the time and frequency domain, both convolutional and fully-connected architectures, and in the case of the Time-Frequency one only convolutional due to its two-dimensional nature.

Table 5-1: Architectures Analyzed

ID Model	Domain Representation	Type of Architecture
ARCH 1	Time	Full-Connected
ARCH 2	Time	Convolutional
ARCH 3	Frequency	Full-Connected
ARCH 4	Frequency	Convolutional
ARCH 5	Time-Frequency	Convolutional

After defining the five underlying architectures to be tested, I make a hyperparameter evaluation for determining which are the most appropriate hyperparameter values for every single one of the five architectures. The architecture design of the best models can be found int Appendix A. They correspond to the deepest ones created, and the justification of their design is more thoroughly explained in the following section.

5-3 Hyperparameter Evaluation - Simplistic GWs

As it was mentioned in the methodology, the hyperparameter evaluation is performed first with the simple sinusoidal waves recordings. The main reason behind this decision is that the simplicity of these waves allows easier visualization of the effect of the changes in the hyperparameters during the performance evaluation than with more complex recordings like the ones from the MBHBs. This way, I can focus better the computational resources available in adapting this found optimal architecture to the specificities of the detection of MHBHs that is the ultimate goal.

Evaluation of the Effect of the Number of Recordings

The first hyperparameter to be tested is the number of recordings used to train the ANNs. For this purpose, I created five different training datasets with different numbers of recordings, namely: 1.000, 2.000, 5.000, 10.000, and 20.000 waves. Each recording had a duration of 36.000 s. I created the deepest model possible for each of the five architectures mentioned in Table 5-1. For the training process, I used 15 epochs and repeated it 15 times with different initialization values. Therefore, for each architecture, 15 realizations were obtained, which allowed analyzing the variability and consistency of the results and make a more reliable comparison among architectures. An average training accuracy and validation accuracy was calculated from the 15 realizations for producing an accuracy curve of the training process.

An example of these average accuracy curves can be visualized in Figure 5-1 for the architectures ARCH-3 and ARCH-4, trained using the dataset containing 20.000 recordings. In this Figure, it is possible to see that in the case of ARCH 3, which makes use of FCNNs, there is a better behavior in terms of overfitting being the validation accuracy curves and the training one close to each other. In contrast, in most of the training the ARCH 4, which make use of CNNs, the overfitting is more present being the two curves very far from each other.

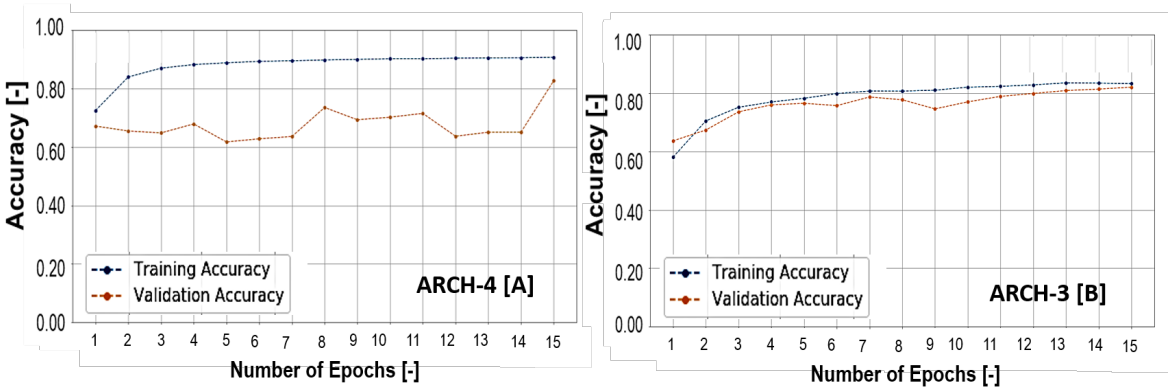

Figure 5-1: Visualization of the accuracy during training and validation phases for 20.000 waves recordings for architectures [A] ARCH 4 - Frequency Domain / CNNs [B] ARCH 3 - Frequency Domain / FCNNs

A testing dataset with different recordings was used for the performance evaluation. The testing dataset contained 20.000 recordings in total. The 15 realization models were used to classify the testing dataset, and this classification was then grouped according to their signal-to-noise ratio (SNR). From this discrimination by the level of noise of the recording, it was possible to calculate accuracy. Afterward, the mean accuracy per SNR was obtained from the results of each of the 15 realization model per architecture. The result of this operation is shown in Figure 5-2. In this Figure, a comparison among the different architectures is made. In the left column, it is possible to see all the architectures that make use of FCNNs, while in the right one are all the ones that use CNNs. In the first row, there are the results when using time-domain represented data, in the second one when using frequency-domain representations and in the last one when using spectrograms. From these results, it is observable that for one-dimensional represented data (time-domain or frequency domain) there is higher variability in the accuracy when using CNNs. In contrast, FCNNs show to have more consistent behavior across the different realizations for both the time and frequency domain represented data. From all these four architectures, ARCH-3 shows to be the only one adequate to have high accuracies in SNR lower than 1. On the other hand, for the time-frequency represented data (spectrograms) the results when using CNNs do not present a high dispersion and they exhibit to have the highest accuracy across all the five architectures being the only one reaching 100 % accuracy in the highest SNR scenarios and also the most accurate in the lowest SNRs.

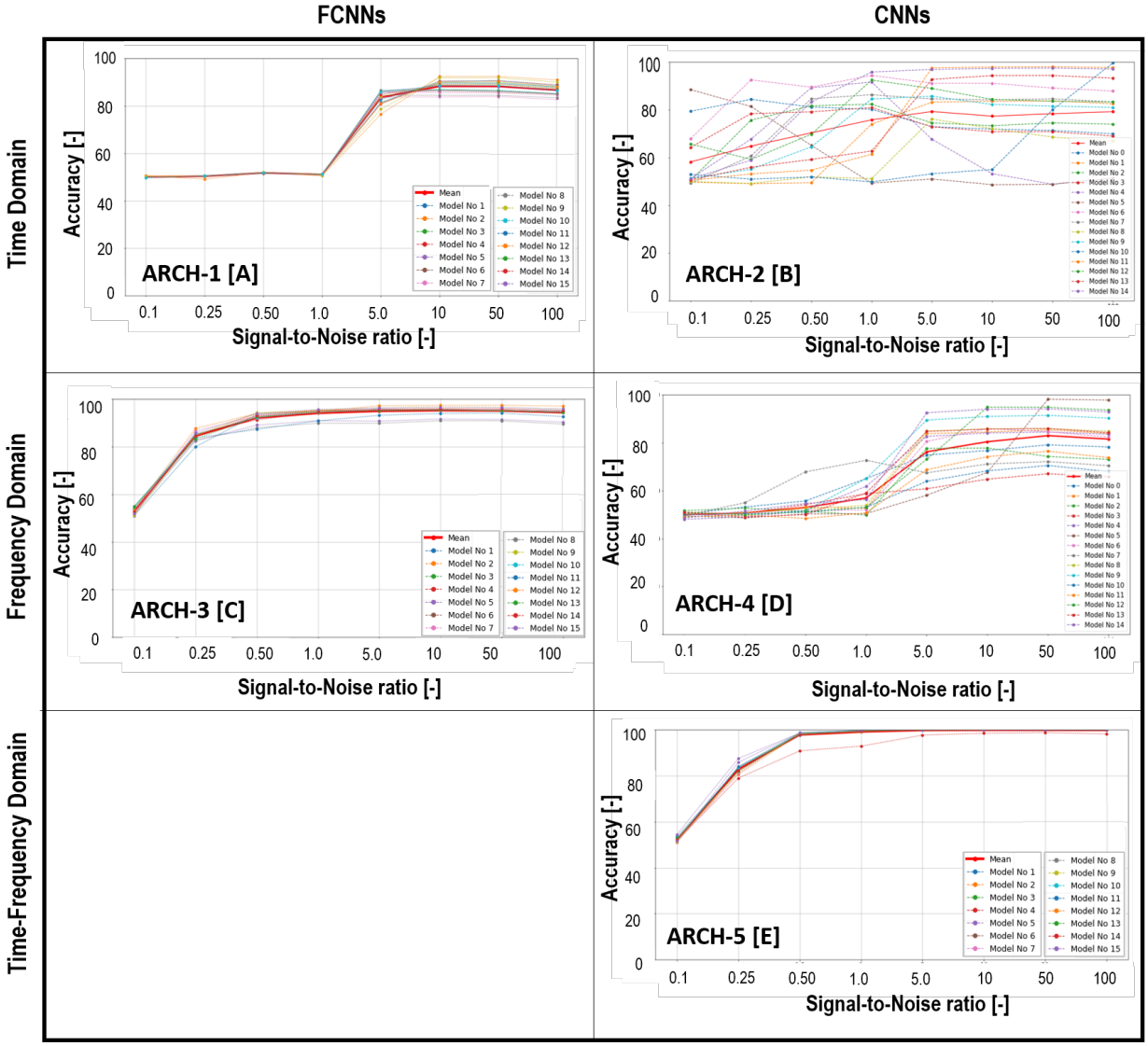

Figure 5-2: Accuracy results on discriminated by signal-to-noise ratio of signals for a testing dataset constituted by 20.000 recordings for [A] ARCH 1 - Time Domain / FCNNs [B] ARCH 2 - Time Domain / CNNs [C] ARCH 3 - Frequency Domain / FCNNs [D] ARCH 4 - Frequency Domain / CNNs [E] ARCH 5 - Time-Frequency Domain / CNNs

I repeated this performance evaluation for the datasets constituted with 1.000, 2.000, 5.000, and 10.000 recordings. Similarly, I calculated for each of the architectures a mean curve of accuracy from testing dataset from 15 realizations. A comparison of the results is shown in Figure 5-3 which are organized following the same logic order than in Figure 5-2. For the frequency-domain and time-domain representations, second and third rows, it is possible to see a general tendency were the largest the number of recordings in the training dataset the higher the accuracy reach in the different SNR levels. However, for the time-domain representation for both FCNNs and CNNs, there seems to be an oscillation of the results around 5.000 waves where the models trained with more records perform better in lower SNR recordings but decrease their performance in higher SNR ones. In contrast, the models

trained with the lower number of waves tend to have higher accuracies in the higher SNR recordings but lower ones in the noisiest ones. Overall, for the best performance are reached for the models from ARCH 3 and ARCH 5.

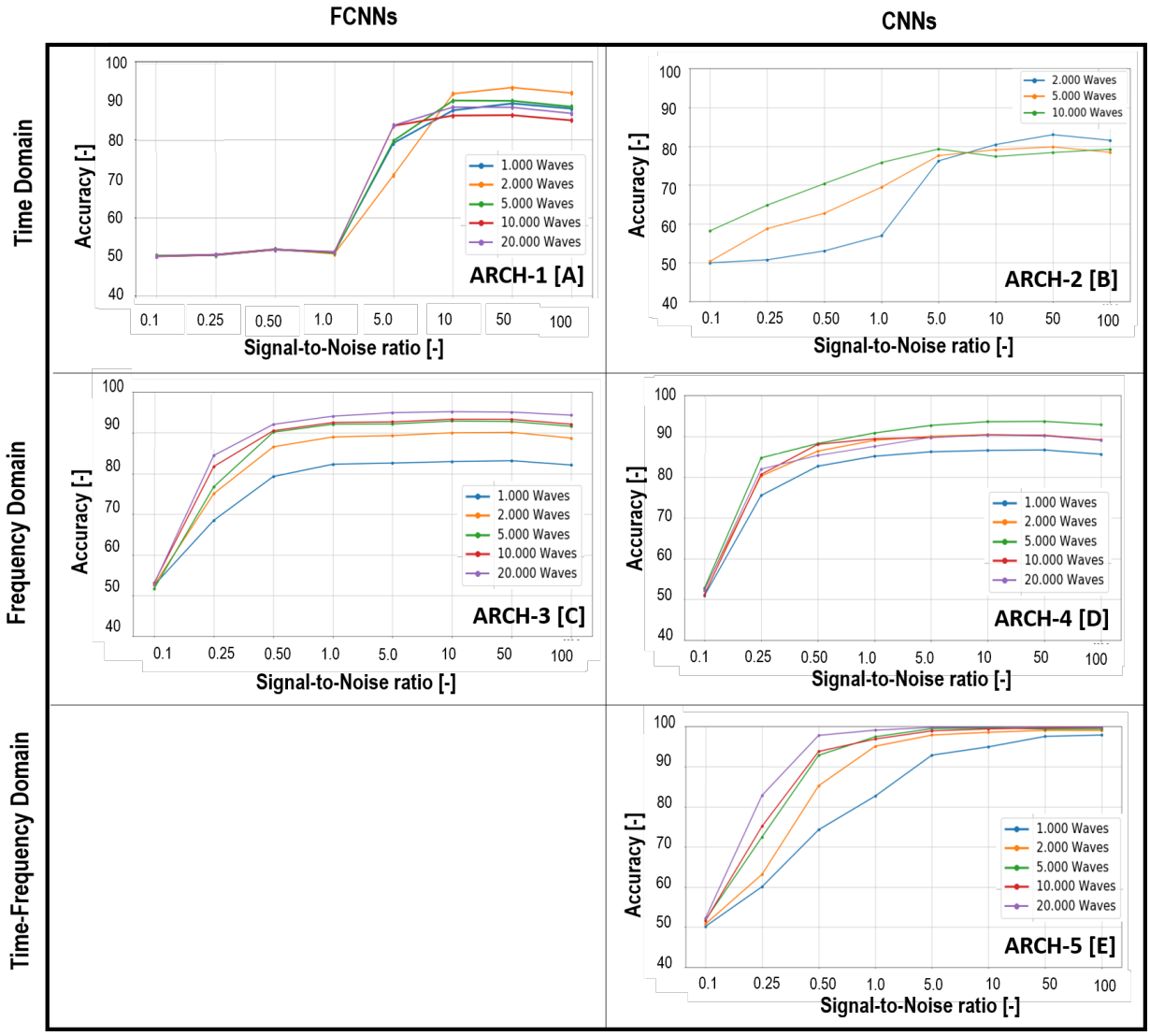

Figure 5-3: Comparison for different number of waves for FCNN for Power vs Frequency for [A] ARCH 1 - Time Domain / FCNNs [B] ARCH 2 - Time Domain / CNNs [C] ARCH 3 - Frequency Domain / FCNNs [D] ARCH 4 - Frequency Domain / CNNs [E] ARCH 5 - Time-Frequency Domain / CNNs

Evaluation of the Effect of the Number of Epochs

The second hyperparameter to be analyzed is the number of epochs set during the training process. The behavior of this hyperparameter during the optimization process showed a more consistent behavior than one of the numbers of recordings. For this reason, only the results of the ARCH-3 on a training dataset with 20.000 recordings are presented in Figure 5-4. For this analysis, the training process was done over the same training dataset with four different

number of epochs per training. These number of epochs were respectively 15, 30, 60 and 90 epochs. In Figure 5-4, it is possible to see that when increasing the number of epochs from 15 to 30, there is an overall increase in accuracy on the testing dataset. Even though, after 30 epochs, the number of epochs does not increase significantly being the accuracy results very similar after training with 60 epochs and 90 epochs. Therefore, the optimal results for the ARCH-3 architecture are reached when using 30 epochs. In the case, of all the other architectures the optimal number epochs are 15, meaning a change in the number of epochs when training does not have an essential impact inaccuracy as it does for ARCH-3.

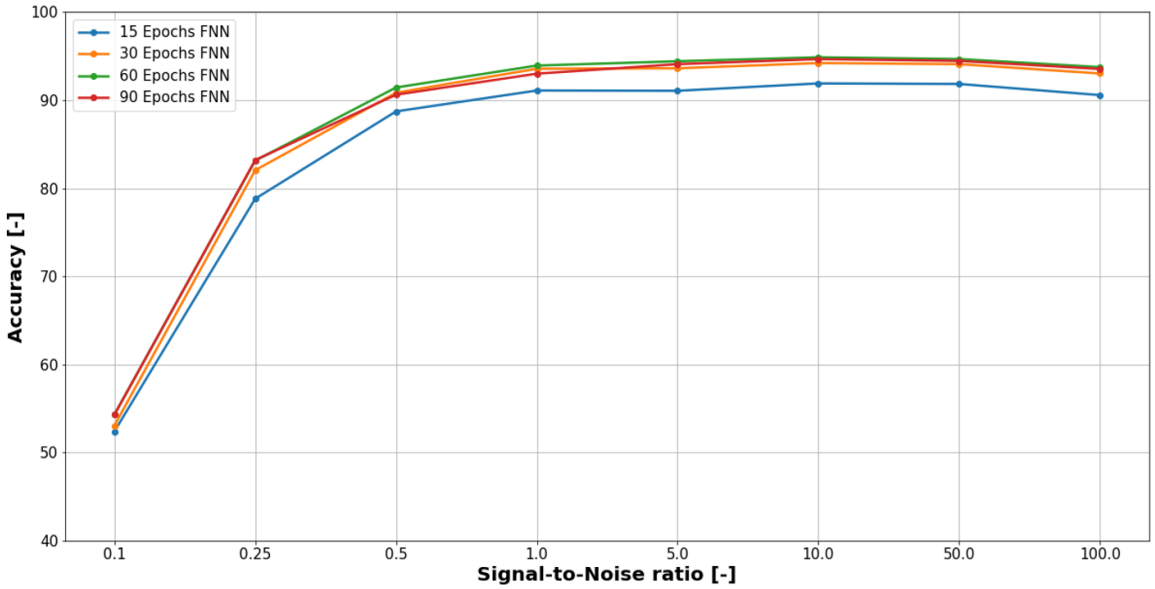

Figure 5-4: Comparison for different number of epochs on the ARCH 3 - Frequency Domain / FCNNs

Evaluation of the Effect of the Number of Layers

The third hyperparameter to be studied is the effect of the number of hidden layers in the ANN architecture. The results of this analysis are shown in Figure5-5, where only the results for ARCH-3 are shown for similar reasons to the case of the number of epochs. From the Figure, it is possible to state that when the number of layers used in the design of the neural network increases so does the accuracy of the classification. Nevertheless, after a certain depth, no significant changes or improvements are achieved. In the graph, this happens after having more than 4 FCNN hidden layers. For the ARCH-1 and ARCH-2, this happens after a maximum number of layers of 9, for the ARCH-4 is 5 and for the ARCH-5 it is 3 hidden layers.

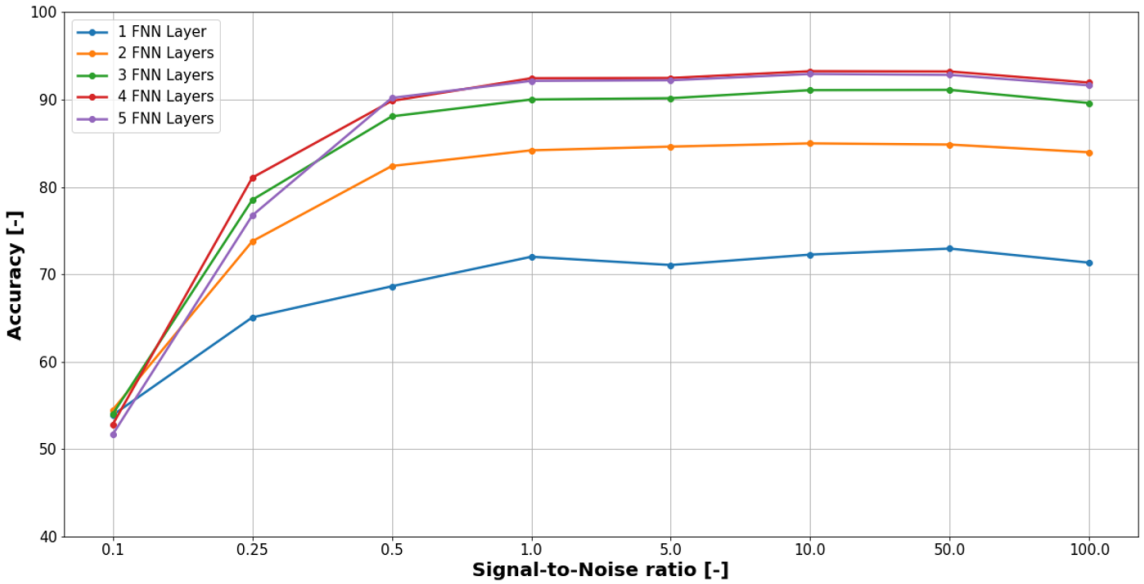

Figure 5-5: Comparison of the accuracy results on a testing dataset with 20.000 recordings for the ARCH 3 - Frequency Domain / FCNN with different number of layers

Evaluation of the Effect of the Type of Neural Network

In this hyperparameter optimization, a comparison is made to see the effect of the type of neural network used, either CNN or FCNN, in the architectures. This analysis can only be done for the one-dimensional representations, which are the time-domain and the frequency domain. In Figure 5-6 I show the results of this comparison for ARCH-3 and ARCH-4, where the accuracies are displayed for models trained with 1.000, 5.000, and 20.000 recordings, respectively. In the Figure, it is visible that for a fewer number of recordings the CNNs seem to perform better than the FCNNs and that this tendency progressively changes with an increasing number of recordings used in the training phase. Even though taking into consideration all the previous results shown, it is possible to say that CNNs tend to be more sensitive to aspects like initialization values and number of recordings showing more spread in them. Therefore, for the types of gravitational analyzed until the moment, it is safe to state that the FCNNs represent a more reliable architecture for performing classification. This behavior is consistent with their tendency of showing an overall improvement in accuracy with deeper models, more epochs during training and number of recordings, which are all the expected behaviors of this type of architectures in applications in other fields of knowledge.

Choice of the best model

After having optimized the five different architectures (ARCH 1,2,3,4 and 5), I make a final comparison among them that is shown in Figure 5-7. In this Figure, I contrast the mean accuracy results of the deepest possible models, trained with 20.000 recordings and 15 epochs during training per architecture. There it is possible to see that overall the CNNs trained with the time-frequency domain, blue line, reach the highest accuracies across all the different levels of noise. Only at an SNR of 0.25, it is surpassed by ARCH-3 by approximately 2-3%. Even though it is possible to say that this difference is not significant because these curves are the average of 15 realizations, and the mean is a statistical measurement which is highly influenced

by extreme values. Therefore, based on this result, I choose ARCH-5 as the architecture most appropriate to use in the galactic binaries and MBHBS gravitational wave recordings in the following steps of the methodology.

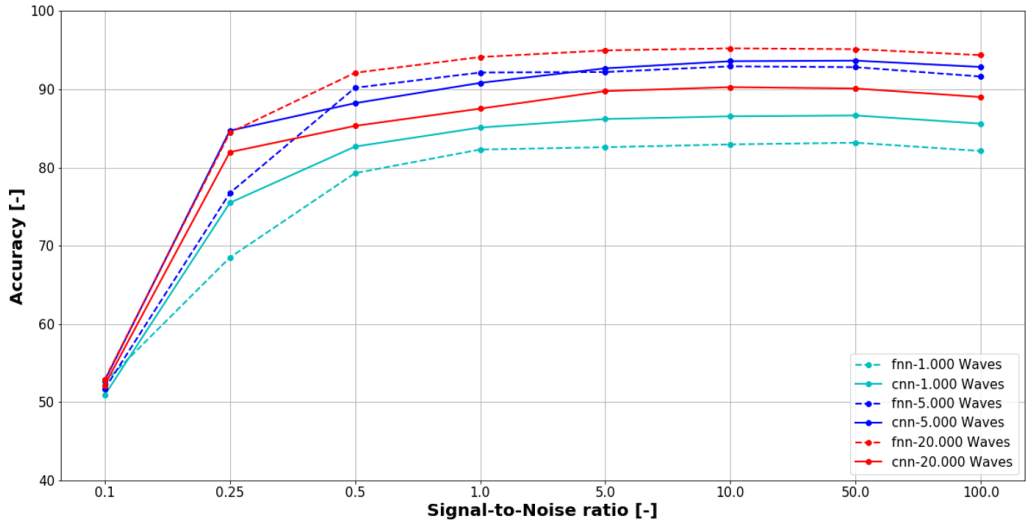

Figure 5-6: Comparison for different number of waves between the use of FCNNs (ARCH-3) vs CNNs (ARCH-4) for Frequency Domain representation

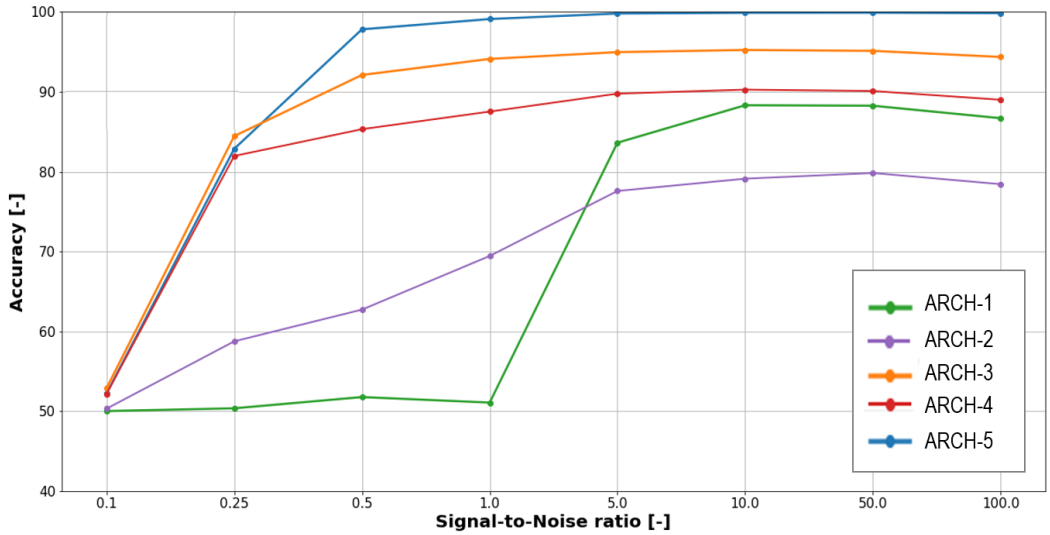

Figure 5-7: Accuracy results discriminated by signal-to-noise ratio of signals for a dataset constituted by 20.000 recordings for ARCH-1 (Time Domain / FCNN), ARCH-2 (Time Domain / CNN), ARCH-3 (Frequency Domain / FCNN), ARCH-4 (Frequency Domain / CNN) and ARCH-5 (Time-Frequency Domain / CNN)

5-4 Hyperparameter Evaluation - Realistic GWs

A new hyperparameter analysis very similar to the one done in the previous section with the simple sinusoidal gravitational wave recordings was done with training and testing datasets for Galactic Binaries (GBs) and Merging Black-hole Binaries (MBHBs) data. The analysis also indicated that the best detection performance was achieved when using CNNs fed with the largest possible number of wave recordings represented in the time-frequency domain. In Figure 5-8 I show the mean accuracy curves of the performance evaluation of the best models for each type of gravitational wave to study. In this case, the number of model realizations tested was just 10 instead of 15, and the number of signal-to-noise ratio were only 6 instead of 8.

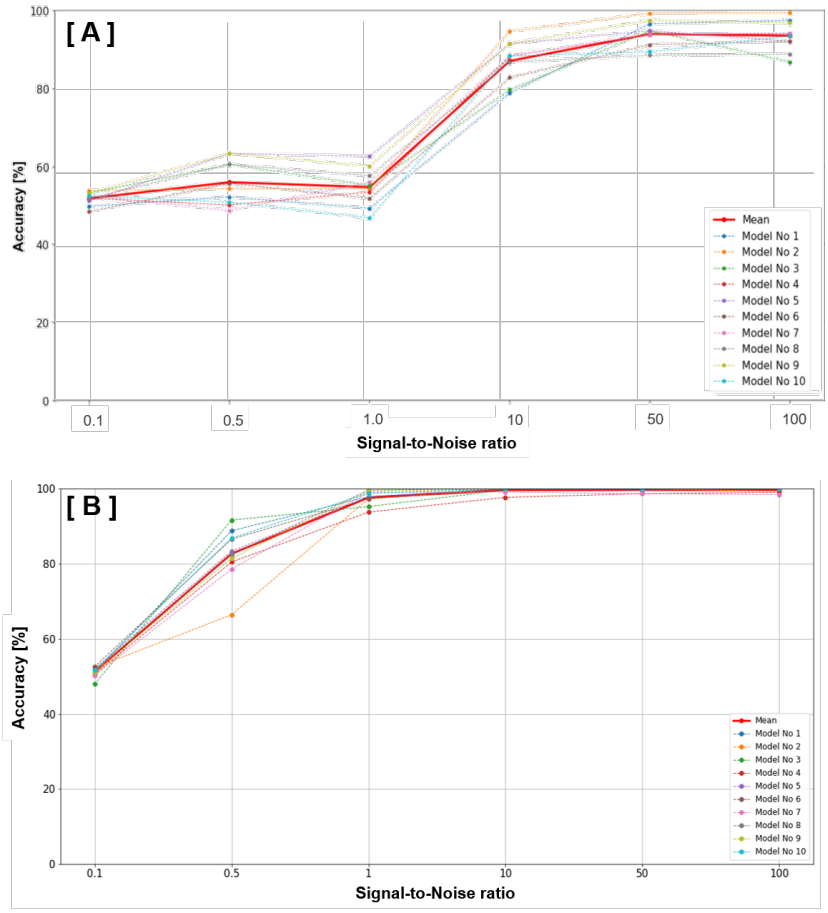

Figure 5-8: Final results of CNNs trained with a dataset of 20.000 gravitational waves recordings of [A] Galactic Binaries and [B] Merging Black-hole Binaries represented in the time-frequency domain (spectrograms)

For the Galactic Binaries, it is possible to see that the neural network didn't reach significant results for SNR lower than 1 having a very similar shape to the results obtained with ARCH-1 in Figure 5-7. In contrast, for the Merging Black-hole Binaries (MBHBs) it is possible to see that the CNN reaches very high accuracies even for an SNR of 0.5. Therefore, the best model realization out of the 10 presented in Figure 5-8 [B] was selected as the classification network

for the sliding window algorithm to be implemented as part of the Low Latency Detector (LLDs). This exercise was repeated with training datasets with 2.000, 5.000, and 10.000 recordings for comparison and a deeper analysis of the results.

5-5 Sliding Window Performance

For testing the performance of the Low Latency Detector (LLD), I produced a one-year recording with the presence of six MBHBs at different levels of noise like the one shown in Figure 4-14. The chosen DL model to be used as the classifier of the LLD was the Model No 3 out of the 15 in Figure 5-8 [B] which is the one that has the overall higher accuracy values across from all of the models tested. Once, I select the DL model that will act as a detector and produced the testing record and follow the procedure explained in section 4-2-4 for this purpose.

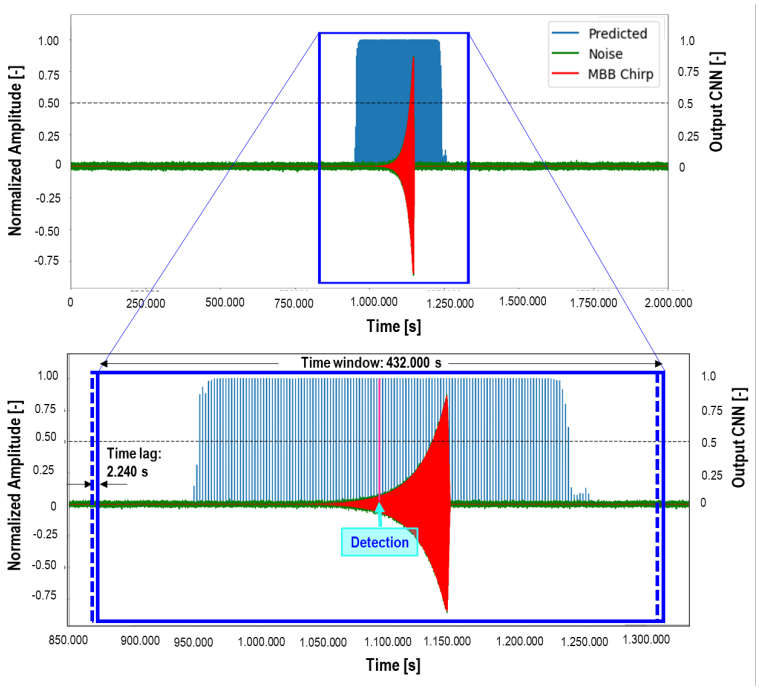

Figure 5-9: Scaled view of a Merging Black-hole Binary signal (red line) with an SNR of 100 and added Gaussian Noise (green line) with respect to the output of the Low Latency Detector (LDD) with classifier CNN with 20.000 recordings.

In Figure 5-9, I show a close view where is visible the dimensions of the classifying window (blue rectangle) and the time-lag I am using with respect to the MBHB present in that segment of the recording. There the red line highlights the presence of the gravitational wave if it was in a noise-free condition. The green line shows the total amplitude of the recording that includes the added Gaussian noise. The amplitude of both scenarios was being normalized to make it comparable to the output of the classifier CNN and evaluate whether it is detecting it correctly. This normalization is done with respect to the maximum amplitude of the MBHB. Finally, the blue vertical lines represent the raw output of the classifier CNN network (Model

No 3) trained with the time-frequency domain representation of the MBHBs.This output is a number between 0 and 1, the closer to 1, the higher the probability of having the presence of a gravitational wave inside of the window analyzed. As it is shown in the closer view, in a magenta color vertical line, the detection of the sliding window is located once it passes by in the middle of the time interval of the recording that it has evaluated. The segment shown corresponds to an SNR of 100, reason why the detectors show a very nice behavior, having a smoothing in the value of the outputs once the MBHB enters and leaves the time window analysis.

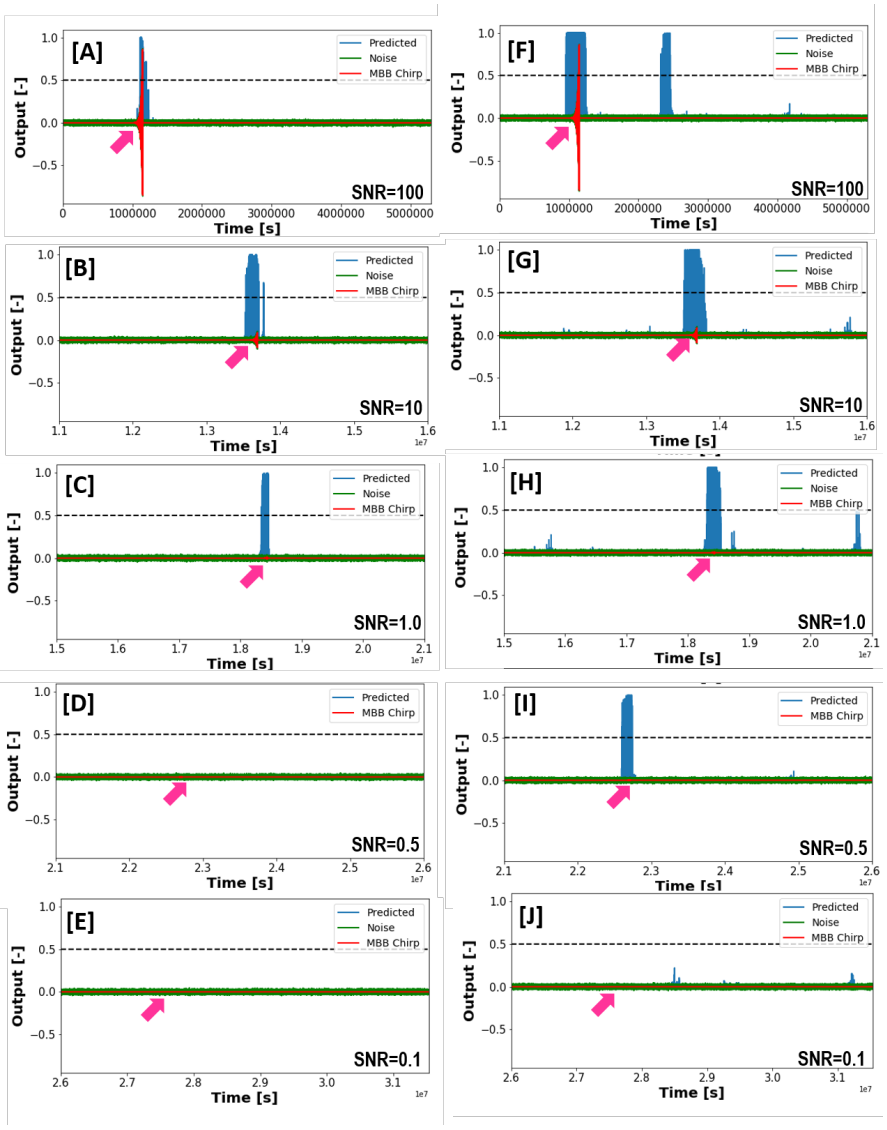

Figure 5-10: Results of the sliding window classification algorithm trained with Merging Black-hole Binaries (MBHBs). The red line represents the waveform of the MBHBs noise-free scaled for comparison purposes. The green line represents the recording with the added Gaussian noise, the blue line is the output of the classifier CNNs, indicating the presence of MBHB. The pink arrows indicate the position of the MBHB in the segment of time illustrated. [A], [B], [C], [D] and [E] are the results when using a detector trained with 10.000 recordings. [F], [G], [H],[I] and [J] are the results when using a detector trained with 20.000 recordings.

In Figure 5-10 I show the comparison results of running the prototype Low Latency Detector (LDD) with CNNs trained with a different number of recordings. For the left column, the detector has been trained with 10.000 recordings, while for the right column it has been trained with 20.000 recordings. For the results in the left column, detections are made only until an SNR of 1.0. In contrast, for the detector trained with the larger number of recordings, detections start to appear also in the segments under an SNR of 0.5. Another observation is that the outputs did with the detector of the right column, the areas of detection are much wider than the ones from the left column, indicating that the LLD is more sensitive to the presence of the GWs.

Nevertheless, this improvement in the number of detections comes with a cost in precision, that is in an increased the number of false positives. An example of this is shown in Figure 5-10 [F] where at approximately 2.500.0000 s a false detection is made. For having a greater insight into this trade-off phenomenon (described in Chapter 3), I do a precision-recall analysis that is shown in Figure 5-11 for the detector trained with 20.000 recordings. There it is possible to see that for SNRs of 1,10,50 and 100 I obtain perfect results, having both a precision and a recall equals to one. However, for the SNR of 0.5, it is more visible than the higher the threshold for the classification of the detector, the higher the precision and the lower the recall. In this scenario, I decide to change the threshold from 0.5, that is the one used until now for the detectors, and reduce it to 0.2. The reason is that it is the optimal point where the maximum precision can be maintained without having a significant drop in precision. This decision makes that the detector starts to have more false positives (lower precision) but guarantees not to be missing important signals (high enough recall).

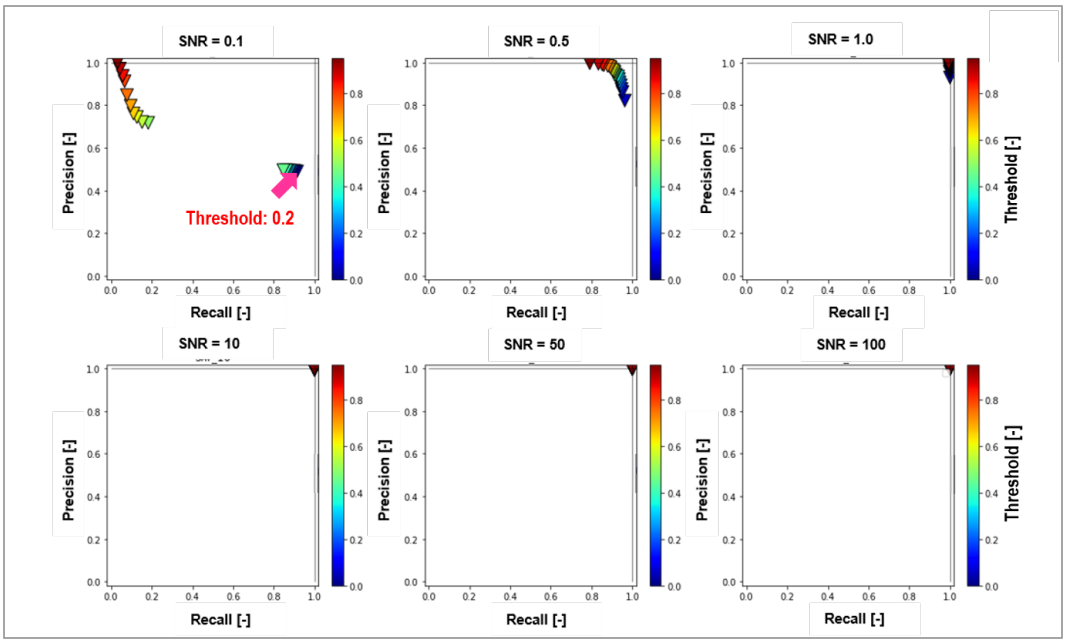

Figure 5-11: Precision-recall curves for the detector of the Sliding Window Algorithm differentiated per SNR. The model has been trained with 20.000 waves. The red arrow indicates the optimal threshold required to maximizing the number of detections even in the highest levels of noise (SNR=0.1)

I run once again the LLD in the one-year recording, using the detector trained with the largest number of recordings and a threshold of 0.2 for the classification. The results of the detections is shown in Figure 5-12. Now, with these changes the LLD can detect the GWs even under an SNR of 0.1. As it is expected, some false positives appear like the one between 30.000.000 s and 40.000.000 s shown in the segment with an SNR 100. The best prototype for the LLD is obtained using the trained model with 20.000 recordings, using a threshold of 0.2. This results are then the best achieved with the available quantity of data produced.

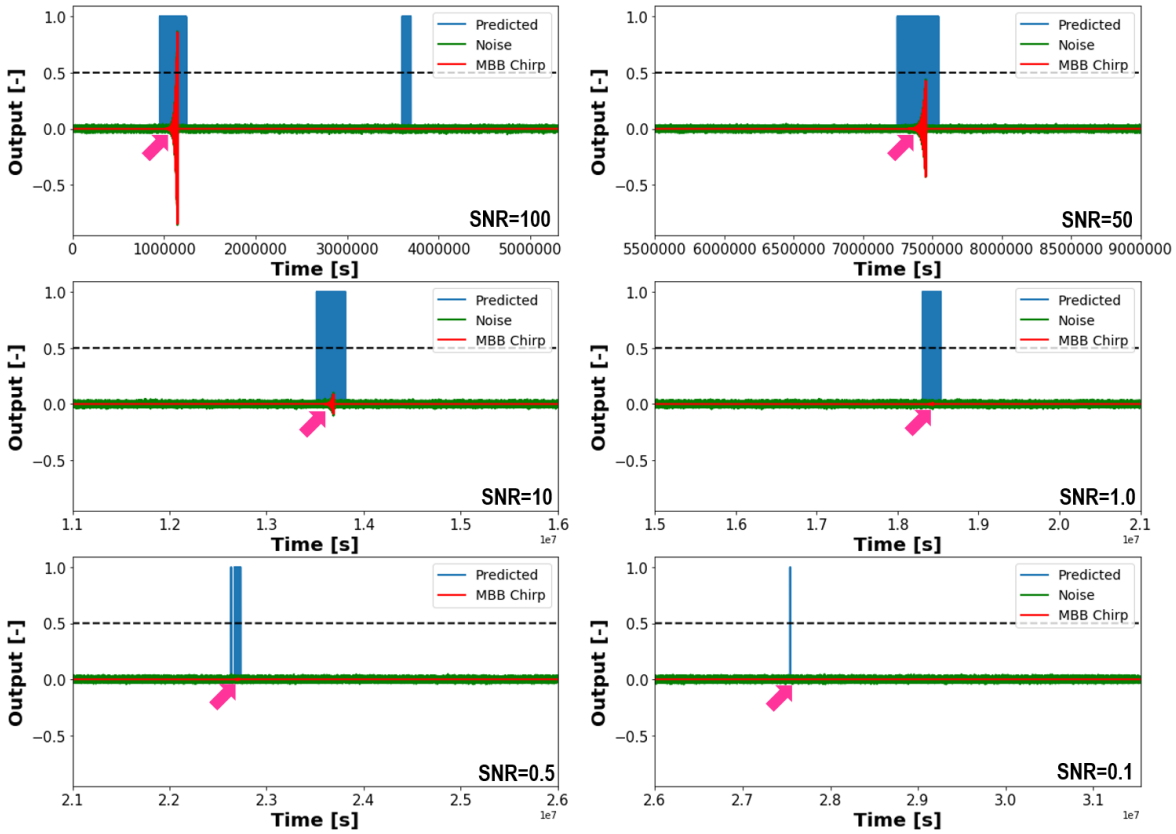

Figure 5-12: Results of the sliding window classification algorithm trained with Merging Black-hole Binaries (MBHBs). The detector has been trained with 20.000 recordings, but the threshold for determining whether a GW is present or not is now 0.3. The red line represents the waveform of the MBHBs noise-free scaled for comparison purposes. The green line represents the recording with the added Gaussian noise, the blue line is the output of the classifier CNNs, indicating the presence of MBHB. The pink arrows indicate the position of the MBHB in the segment of time illustrated. In this scenarion, detections are done even in the lowest SNR (0.1).

Chapter 6

Discussion, Conclusions and Outlook

6-1 Discussion

Improvement of SNR Distribution in Training Datasets

The quantity of recordings is an essential factor for the training of a very accurate and precise DL Model. During this project, I decided to organize the training and testing datasets in such a manner that it was possible to analyze the evolution of performance measurements like accuracy and precision as the hyperparameters of the Neural Networks were tuned. For this comparability, I distributed uniformly the proportions of the recording among eight different levels of SNR ranging from very low (SNR=0.1) to very high ones (SNR=100). This decision has the logic that the records with the highest SNR facilitate the DL model to extract the most characteristic features without being opaque by noise, and the noisiest one were the critical ones for discriminating their presence in noisy environments. Nevertheless,this decision discretizes the distribution of signals that in reality, would exist in a balanced dataset. This can introduce difficulties during the training of the neural networks, especially during the backpropagation process that can find challenging to find optima. This situation is illustrated in Figure 6-1 where it is compared the case of having instead of a smooth continuous distribution (continuous lines), local discrete distributions with specific values (bars). Moreover, having it may be easier for the DL model to learn the characteristic feature for cases with high SNR with less data, while it will require more recordings under high levels of noise to identify those same features. This could explain during the performance tests of most of the models they have very low accuracies in the noisiest situations SNR (0.1). In this sense, it will be beneficial to apply other distributions that increases the number of recordings with low SNR while maintaining continuity and a good proportion of other recordings with high SNR.

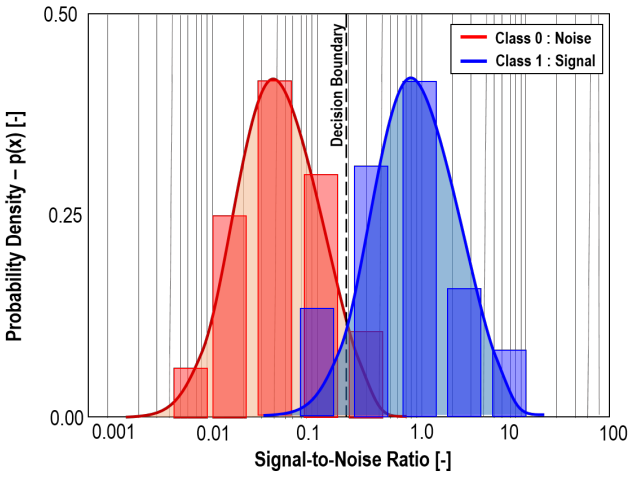

Figure 6-1: Graphical representation of the problem of finding the decision boundary between two classes. The bars represent the case of having a discretized distributions by having a dataset containing only selected SNR levels while the continuous curves represent the case where not inner selections are done. **Note**: The presence of an SNR number under the Class 0 which is noise does not imply the presence of signals in these recordings. This is just an attempt of representing the difficulties of inner discretization.

To overcome the problems of having training datasets which range in SNR is not continuous in Class 1(signal), I construct a new dataset following the workflow shown in 6-2. First, I take the range of SNR values, which are from 0.1 to 100 and calculate their logarithm. This transformation expresses the values in a scale where it is easier to apply increments in the proportions of the different orders of magnitude of SNR or guarantee continuity among them in the data. Later, I use the maximum, minimum, and a value in between for designing a random triangular distribution of values that will increase the proportion of data points which SNR is lower than 1. The result after applying this distribution for 20.000 generated recording, is then the new dataset shown at the end of the workflow. This dataset has an increment in the number of recordings that present the highest levels of noise while maintaining a large number of recordings where it is low.

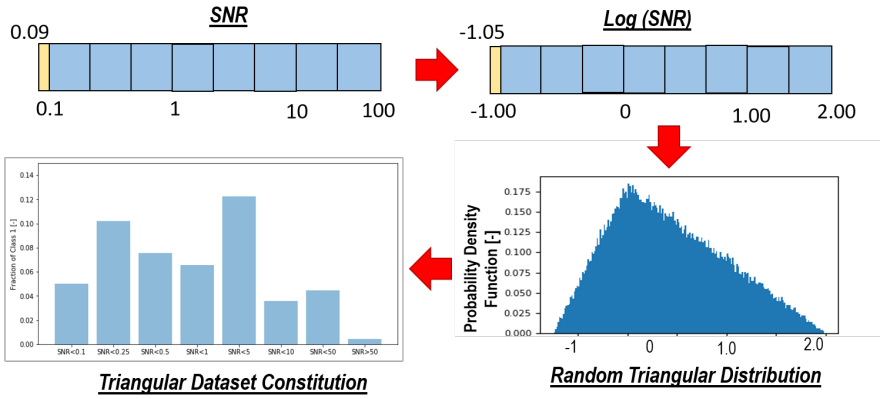

Figure 6-2: Workflow for the construction of a new dataset which can follow a more favourable proportion for reaching higher level of accuracies in the detection of GWs under the noisiest levels

Afterward, I performed an accuracy analysis using the DL model trained with the newly created dataset. In Figure 6-3[B] I show a comparison of the results of this analysis to the previous results shown in 6-3[A] when using a dataset that follows a uniform random distribution for the Class 1 (signal). It is possible to see that there are no significant differences between the performance of both models. The only remarkable observation is that for an SNR of 0.25, there is a decrease in the variability of the results for the new model. This similarity in findings indicates that the models trained before are robust against the problem of not having a continuous range of SNR and that for improving them the effort should be employed better into producing more recordings for the training phase.

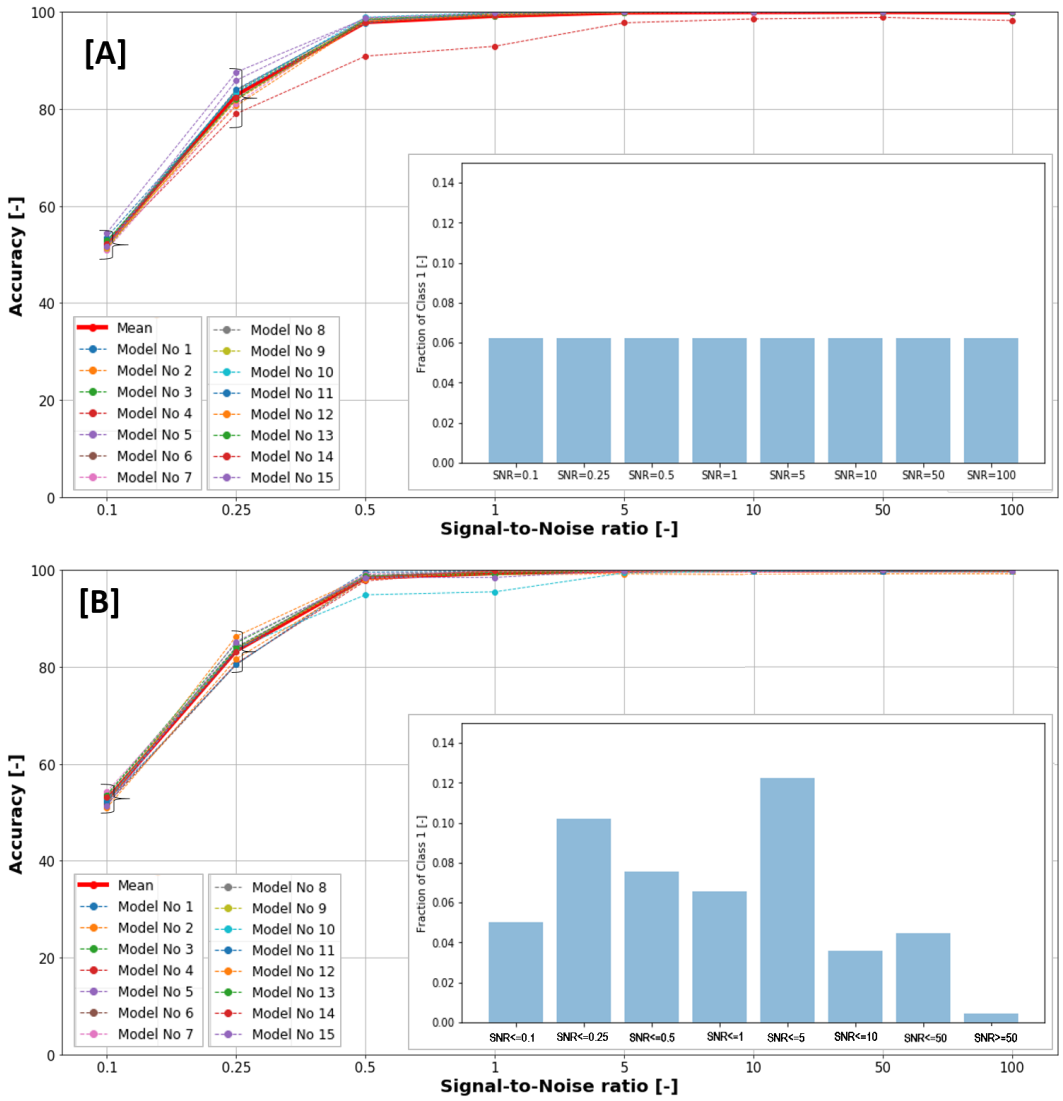

Figure 6-3: Comparison of the average accuracy results between CNN models trained with 20.000 recordings of time-frequency domain data where the [A] the distribution for assigning the SNR level for every Class 1 (signals) recording has been done following a uniform random distribution, and where the [B] the distribution for assigning the SNR level for every Class 1 (signals) recording has been done following a triangular random distribution

Comparison of Time-Domain VS Time-Frequency Domain Trained MBHB Detectors

In recent years, multiple authors and research groups have been testing the bounties of Deep Learning for the study of GWs, particularly for LIGO detected events. Some examples of these works include [George and Huerta, 2017],[Hunter Gabbard and Messenger, 2018] and [N. Muund, 2017]. All of them, have in common the building of DL models using only time-domain represented data recordings for their training datasets. Additionally, they coincide in demonstrating that models build using CNNs using large datasets can reach accuracies very similar to the ones of other methodologies that have proven to be very accurate in analyzing GWs like Matched Filtering. In the work of [Daniel George and Huerta, 2017], the idea of using time-frequency representation (spectrograms) for analysis of GWs recordings is introduced, but it is only used for identifying sinusoidal glitches that contaminate the data. Moreover, the maximum time duration of the data samples used is 4 s. In my analysis, I find that the use of time-frequency representations is beneficial for the detections of the GWs themselves in the recordings and obtain during the performance evaluations that they usually have better overall behavior in terms of accuracy when compared to other where time-domain represented have been used, like in Figure 5-7.

To make a more exhaustive comparison between these two domain representations, and to look at the excellent results that those previous works have made with time-domain represented data, I trained a group CNN models using 20.000 time-domain represented recordings of MBHBs. In Figure 6-4 I show the comparison between this model and the time domain representation model trained during the construction of the LLD. It is possible to see that overall, the results of the time-frequency domain are much better.

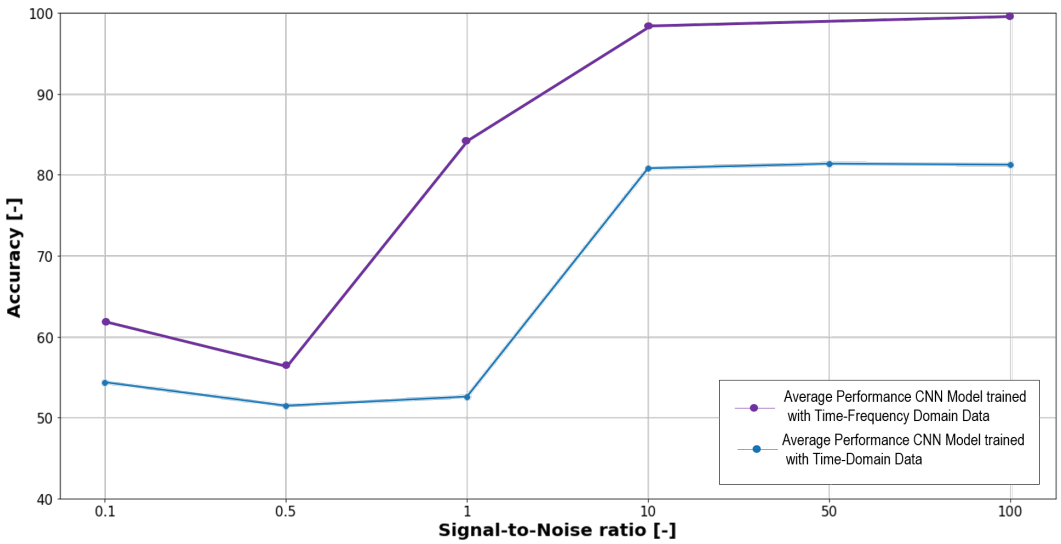

Figure 6-4: Comparison of the average accuracy results of obtained from the CNN models trained with 20.000 recordings using time-frequency domain represented data [purple curve] and time-domain represented data [blue curve],distinguished by SNR, during the testing phase

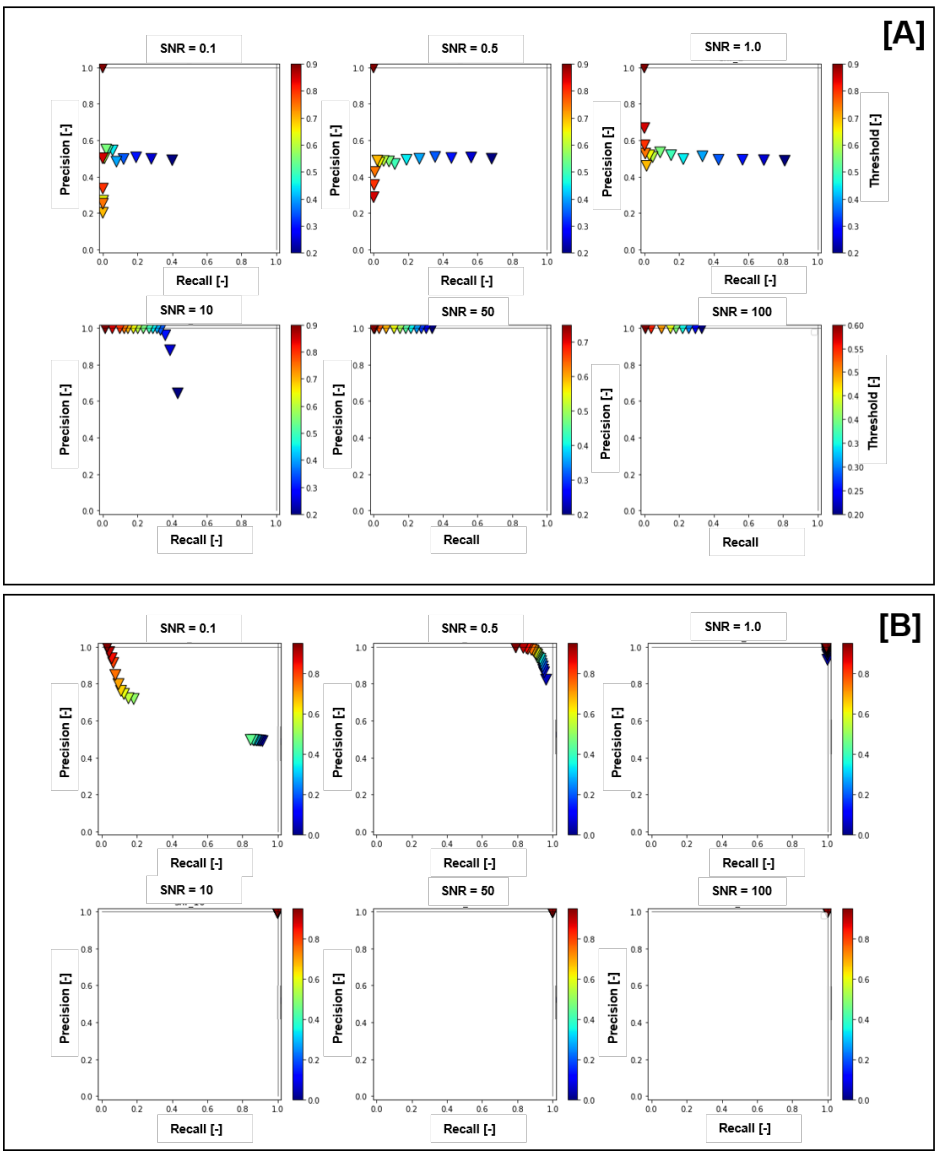

Figure 6-5: Comparison of the precision-recall plots distinguished by SNR of the best models trained using 20.000 recordings for the [A] CNN model trained using only time-domain represented data [B] CNN model trained using only time-frequency-domain represented data

Afterward, I make a precision-recall plot for the same model trained with time-domain represented data. The results are shown in Figure 6-5. There it is possible to observe a very interesting phenomenon. For the time-domain representation, the results are of high precision been very close to the vertical axis of the plots. This means that when using them, there are fewer chances of having false positives, but also that the number of correct detections that they do is lower. This can explain why even though they do not show an outstanding behavior when in accuracy plots, it is still possible to make some good detections like the best models shown by Model No 6 in Figure 5-7 [B] that reaches excellent results at SNR of 0.5.In contrast, in the case of the models trained using time-frequency domain represented data, there is a good recall that allows not to miss the majority gravitational wave events.

Therefore, following these, it is possible to say that the DL model that uses time-domain domain representation is good when solving tasks where the precision is more important to avoid having false positive detections. Meanwhile, the time-frequency domain representation is more suitable for other tasks where the recall is more crucial, like in the case of our prototype LLD.

6-2 Conclusions

I generated synthetic datasets for two types of GW sources, Galactic Binaries (GBs) and Merging Blackhole Binaries (MBHBs). This synthetic data were represented in three different domains: the time-domain, the frequency domain, and the time-frequency-domain. I trained and tested DL models built following two ANN architectures, specifically, Convolutional Neural Networks (CNNs) and Fully-Connected Neural Networks (FCNNs) using the different datasets mentioned before. In total, I had five different combinations of architectures and data representations, namely: ARCH-1 (FCNN trained with time-domain data), ARCH-2 (CNN trained with time-domain data), ARCH-3 (FCNN trained with frequency-domain data), ARCH-4 (CNN trained frequency-domain data) and ARCH-5 (CNN trained with time-frequency domain data).

I made an exhaustive hyperparameter evaluation analysis using the five different architecture-dataset combinations by using only simple Galactic Binaries data that does not consider the effect of LISA's orbit. I found the best models with the best performance from the five and replied the hyperparameter evaluation, but this time using more realistic datasets of both GBs and MBHBs. Then, I selected the best classifier model for MBHBs and built a prototype Low Latency Detector (LLD). This LLD uses a sliding window algorithm, with a window size of 5 days (532.000 s) and a time lag of less than 40 min (2.240 s) that performs classification by using the trained DL model over time-frequency domain represented data.

I tested the performance of the LLD in a one-year total duration recording, where six MBHBs where placed spaced in segments of equal duration (2 months) but under different levels of noise (SNRs of 0.1, 0.5, 1, 10, 50 and 100). I made a precision-recall analysis of the classifier and selected the optimal threshold for having detections even under the highest levels of noise (SNR=0.1).

Finally, I compared the effect in the performance of varying the distribution of the SNR in the datasets by using a uniform random and a triangular random distribution. Additionally, I compared the advantages of using classifiers that use MBHB time-domain represented data with others using frequency-domain represented data.

The following conclusions can be made after this process:

- Deep learning is a powerful tool for GW detection in recordings where the SNR is very poor
- CNN architectures trained with time-frequency domain data are useful tools for detection of GBs and MBHBs even in the circumstances with very high levels of noise. This combination reached accuracies close to 98 % in the simple GB data and 90 % in MBHB data with an SNR of 0.5. Moreover, when maximizing their recall by decreasing their

detection accuracy threshold to lower levels (0.3), they can detect GWs in recordings with SNR of 0.1.

- The production of two-dimensional data representations, like spectrograms, from data that is originally one-dimensional in nature, like GW channels recording, can be useful for increasing the accuracy of deep learning classifier models, especially CNNs.

- The use of time-frequency domain representations of the wave recordings can boost the accuracy and recall of the classification systems for detecting more GW events, especially under high levels of noise.

6-3 Outlook

Deep learning (DL) is a subtype of Machine Learning (ML) approaches that allow building ML models without having to go through the process of manual feature extraction. When looking at state-of-the-art applications of DL to detect gravitational waves, it is evident that the detection accuracies obtained and the classification results improved in terms of computational cost and speed in contrast to other tools previously used in LIGO recordings like Matched Filtering. Researches like [Andrzej Krolak and Vallisneri, 2008b], and [Daniel George and Huerta, 2017] show encouraging results of using DL for solving this detection task. However, due to the differences in the frequency bands in LISA, the types of noise and the effects introduced by having a spaceborne interferometer, there is the need of having more robust or alternative tools that can be applied for LISA mission. I have shown that using time-frequency domain representations of the wave recordings can boost the accuracy and recall of the classification systems for detecting more GW events, especially under high levels of noise. Also, the benefits of using CNNs on GW two-dimensional representations for Low Latency Detection purposes. Based on the findings of this work, I suggest the improvement and inclusion of the following areas for further investigation:

- **More Massive Synthetic Datasets:** I have shown the benefits and the improvement in the detection when increasing the number of recordings used to train the DL models. In the current research, there was a limitation in the number of recordings that could be produced, limiting the accuracy, precision, and recall that the Low Latency Detector (LLD) could reach. The maximum number of recordings I used was 20.000. Therefore, by using more efficient algorithms that can produce and handle more massive amounts of data, it would be possible to train networks with 500.000 or 1.000.000 recordings. These new models trained with a larger number of recordings could allow achieving detections, even when having lower SNRs than the ones studied in this research. For instance, Generative Adversary Networks could be implemented to produce more data from the existing data challenges produced by The Mock LISA Data Challenge (MLDC) for training for the LISA mission.

- **Alternative Time-Frequency Domain Representations:** The Time-Frequency domain representation I used were Spectrograms calculated by using Short Term Fourier Transformation. An adequate Time-Frequency representation that would be interesting to implement and to compare if better results are achieved is the Continuous Wavelet

Transform (CWT). This approach counts with a better trade-off between frequency and time resolution that STFT in small overlapping segments and provides a better representation when dealing with especially low-frequency data like the one expected in LISA's recordings.

- **Alternative Two-Dimensional Representations:** I showed that using two-dimensional representations can exploit better the benefits of DL models built using CNNs. For this, I used as my two-dimensional representation a time-frequency domain one. Even though, it would be possible of using other signal processing representations in other domains to train the networks and evaluate if they have other advantages that have not been seen so far by the use of spectrograms but also effective for CNNs.

- **TDI 2.0 generation Training Datasets:** A significant simplification I did for generating enough synthetic data was to omit the effects of the orbit of LISA in the modulations in the gravitational wave recordings. It would be very beneficial to create datasets produced with second-generation Time Delay Interferometry corrections that consider LISA as a non-rigid and includes the effect of rotation of the constellation. This would, among its benefits, make the detector more resilient to sinusoidal glitches.

- **Measurement-based Datasets:** I used synthetically generated data for training my DL models. The main reason being that LISA is a project that is still in construction and has not been launched for taking measurements. Even though, once it is in the space, it would be a very valuable improvement to use real recorded data to produce new DL models that can act as the core of LLDs. Using this realistic data would also make more robust the detectors against glitches as is the case of the models build by [George and Huerta, 2017] that uses realistic data measured by LIGO.

- **Non-Gaussian Noise Inclusion:**, The type of noise that I used in this research, was Gaussian Noise, and the PSD was considered stationary. It would be essential to reproduce more realistic non-gaussian noise types and considering the situation of having non-stationary PSD of the noise and GWs recorded. In [George and Huerta, 2017], it is shown that DL models trained using datasets that considered non-gaussian noise conditions had a very nice performance when using time-domain represented data. Therefore, it would be attractive to proof if this also applies when training with time-frequency represented data.

- **Better Balanced Datasets:** I showed that using a uniform or a triangular probability distribution for assigning the inner proportions of the level of noise present in the recordings labeled as signals (Class=1) has little effect on the accuracy of the DL models built from them. Even though, for comparison purposes, both models were trained in testing datasets containing only eight values of SNR (0.1, 0.5, 1, 5, 10, 50, 100). It would be attractive to proof whether these models are robust enough when tested instead with different values of SNR.

- **Linear Regression Benchmarking:** I made the comparison between the performance of different DL architectures, namely CNNs and FCNNs. It would be important to do a systematic evaluation and comparison using more simple models like Linear Regression Classifiers, this exercise would allow making a better judgment about whether the

benefits of building the more complex and deeper models justify the efforts employed in their training when compared to the performance of more simple models.

- **Source Parameter Estimation:** The parameter estimation task was not addressed in this research. Constructing DL models also for solving this task would be an important continuation of this research. Such an approach using deep learning techniques have proven to work in time-domain represented data in [George and Huerta, 2017]. Therefore, this could be extended to a time-frequency domain represented recordings to see if the results can be improved.

Bibliography

[Abbott, 2016] Abbott, B. e. a. (2016). Observation of gravitational waves from a binary black hole merger. *Physical Review Letters*, 6:061102.

[Alex Krizhevsky, 2012] Alex Krizhevsky, Ilya Sutskever, G. E. H. (2012). Imagenet classification with deep convolutional neural networks.

[Andrzej Krolak and Vallisneri, 2008a] Andrzej Krolak, M. T. and Vallisneri, M. (2008a). Optimal filtering of the lisa data.

[Andrzej Krolak and Vallisneri, 2008b] Andrzej Krolak, M. T. and Vallisneri, M. (2008b). Optimal filtering of the lisa data. *Phys. Rev. D*, 70:022003.

[Arkadiusz Blaut, 2010] Arkadiusz Blaut, Stanislav Babak, A. K. (2010). Mock lisa data challenge for the galactic white dwarf binaries. *Phys.Rev.D*, 81:063008.

[Banerjee, 2010] Banerjee, Sambaran; Baumgardt, H. K. P. (2010). Stellar-mass black holes in star clusters: implications for gravitational wave radiation. *arXiv:0910.3954*.

[Benjamin J. Owen, 1999] Benjamin J. Owen, B. S. S. (1999). Matched filtering of gravitational waves from inspiraling compact binaries: Computational cost and template placement. *Phys. Rev. D*, 60:022002.

[Birjoo Vaishnav and Deirdre, 2007] Birjoo Vaishnav, Ian Hinder, F. H. and Deirdre (2007). Matched filtering of numerical relativity templates of spinning binary black holes. *Phys. Rev. D*, 76:084020.

[Branchesi, 2016] Branchesi, M. (2016). Multi-messenger astronomy: gravitational waves,neutrinos, photons, and cosmic rays. *Phys.: Conf. Ser*, 718:022004.

[Christensen, 2018] Christensen, N. (2018). Stochastic gravitational wave backgrounds. *Reports on Progress in Physics.*, 82:016903.

[Cornish, 2017] Cornish, N. J. (2017). Mapping the gravitational wave background. *Quantum Grav.*, 18:4277.

[Cornish, 2019] Cornish, N. J. (2019). The discovery potential pf space-based gravitational wave astronomy. *arXiv:1904.01438 [astro-ph.HE]*.

[Curt Cutler, 2019] Curt Cutler, Emanuele Berti, K. J. E. D. K. L. R. S. V. (2019). What we can learn from multi-band observations of black hole binaries. *arXiv:1903.04069 [astro-ph.HE]*.

[Daniel George and Huerta, 2017] Daniel George, H. S. and Huerta, E. A. (2017). Deep transfer learning: A new deep learning glitch classification method for advanced ligo. *General Relativity and Quantum Cosmology*.

[Daniel Maturana, 2015] Daniel Maturana, S. S. (2015). Voxnet: A 3d convolutional neural network for real-time object recognition.

[Danzmann, 2017] Danzmann, K. (2017). Lisa laser interferometer space antenna: A proposal in response to the esa call for l3 mission concepts.

[Duncan Brown, 2004] Duncan Brown, Stanislav Babak, P. B. N. C. (2004). Searching for gravitational waves from binary inspirals with ligo. *Class.Quant.Grav.*, 21:S1625–S1633.

[Edwards, 2018] Edwards, D. H. C. (2018). Source agnostic exploration of gravitational wave recordings. *IDEA League. Master of Science in Aplied Geophysics. ETH Zurich*.

[Emanuele Berti and Cavaglià, 2007] Emanuele Berti, Jaime Cardoso, V. C. and Cavaglià, M. (2007). Matched filtering and parameter estimation of ringdown waveform. *Phys. Rev. D*, 76:104044.

[George and Huerta, 2017] George, D. and Huerta, E. A. (2017). Deep learning for real-time gravitational wave detection and parameter estimation: Results with advanced ligo data. *Physics Letters B*, 778:64–70.

[H. Sana, 2012] H. Sana, S. de Mink, A. d. K. N. L. C. J. E. (2012). Binary interaction dominates the evolution of massive stars.

[Hunter Gabbard and Messenger, 2018] Hunter Gabbard, Michael Williams, F. H. and Messenger, C. (2018). Matching matched filtering with deep networks for gravitational-wave astronomy. *Phys. Rev. Lett.*, 120:141103.

[James Bergstra, 2012] James Bergstra, Y. B. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13:281–305.

[Keith A. Arnaud, 2006] Keith A. Arnaud, Stanislav Babak, J. G. B. (2006). A how-to for the mock lisa data challenges. *AIPConf.Proc*, 873:625–632.

[Kotsiantis, 2007] Kotsiantis, S. B. (2007). Supervised machine learning: A review of classification techniques. *Informatica*, 31:249–268.

[MacKay, 1996] MacKay, D. J. C. (1996). Hyperparameters: Optimize, or integrate out? *Fundamental Theories of Physics*, 62:43–59.

[MacKay, 1999] MacKay, D. J. C. (1999). Comparison of approximate methods for handling hyperparameters. *Neural Computation*, 11:1035–1068.

[Michele Armano, 2019] Michele Armano, Heather Audley, J. B. P. B. (2019). Lisa pathfinder. *arXiv:1903.08924*.

[N. Muund, 2017] N. Muund, S. Abraham, S. K. S. M. N. S. P. (2017). Transient classification in ligo data using difference boosting neural network. *Phys. Rev. D*, 95:104059.

[Nelemans, 2013] Nelemans, G. (2013). Galactic binaries with elisa. *arXiv:1302.0138 [astro-ph.HE]*.

[S. E. de Mink, 2016] S. E. de Mink, I. M. (2016). The chemically homogeneous evolutionary channel for binary black hole mergers: rates and properties of gravitational-wave events detectable by advanced ligo. *arXiv:1603.02291*.

[Sathyaprakash and Schutz, 2009] Sathyaprakash, B. S. and Schutz, B. F. (2009). Physics, astrophysics and cosmology with gravitational waves. *Living Rev. Relativ.*, 12:2.

[Schutz, 1983] Schutz, B. F. (1983). Gravitational waves on the back of an envelope. *American Journal of Physics*, 52:412.

[Shaun Hooper and Luan, 2010] Shaun Hooper, Linqing Wen, D. B. S. K. C. Y. C. and Luan, J. (2010). Low-latency detection of gravitational waves. *AIP Conf. Proc.*, 1246:211.

[Steve Lawrence, 1997] Steve Lawrence, C. Lee Giles, A. C. T. A. D. B. (1997). Face recognition: A convolutional neural-network approachs. *IEEE TRANSACTIONS ON NEURAL NETWORKS*, 8:1.

[Tyson Littenberg, 2019] Tyson Littenberg, Katelyn Breivik, W. B. . L. (2019). Astro2020 decadal science white paper: Gravitational wave survey of galactic ultra compact binaries. *arXiv:1903.05583 [astro-ph.HE]*.

Appendix A

Artificial Neural Networks Models

In this appendix, I show the the best ANN models that I developed during the research.

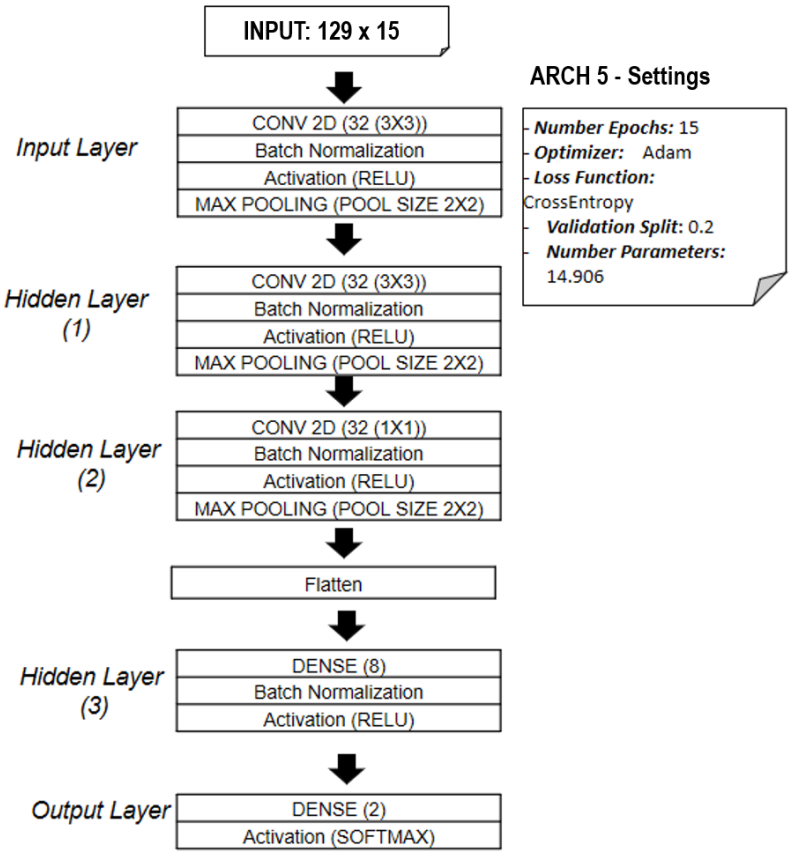

Figure A-1: Deepest ANN model generated for the ARCH-5 architecture-domain combination (Convolutional Neural Networks trained using time-frequency domain represented data).

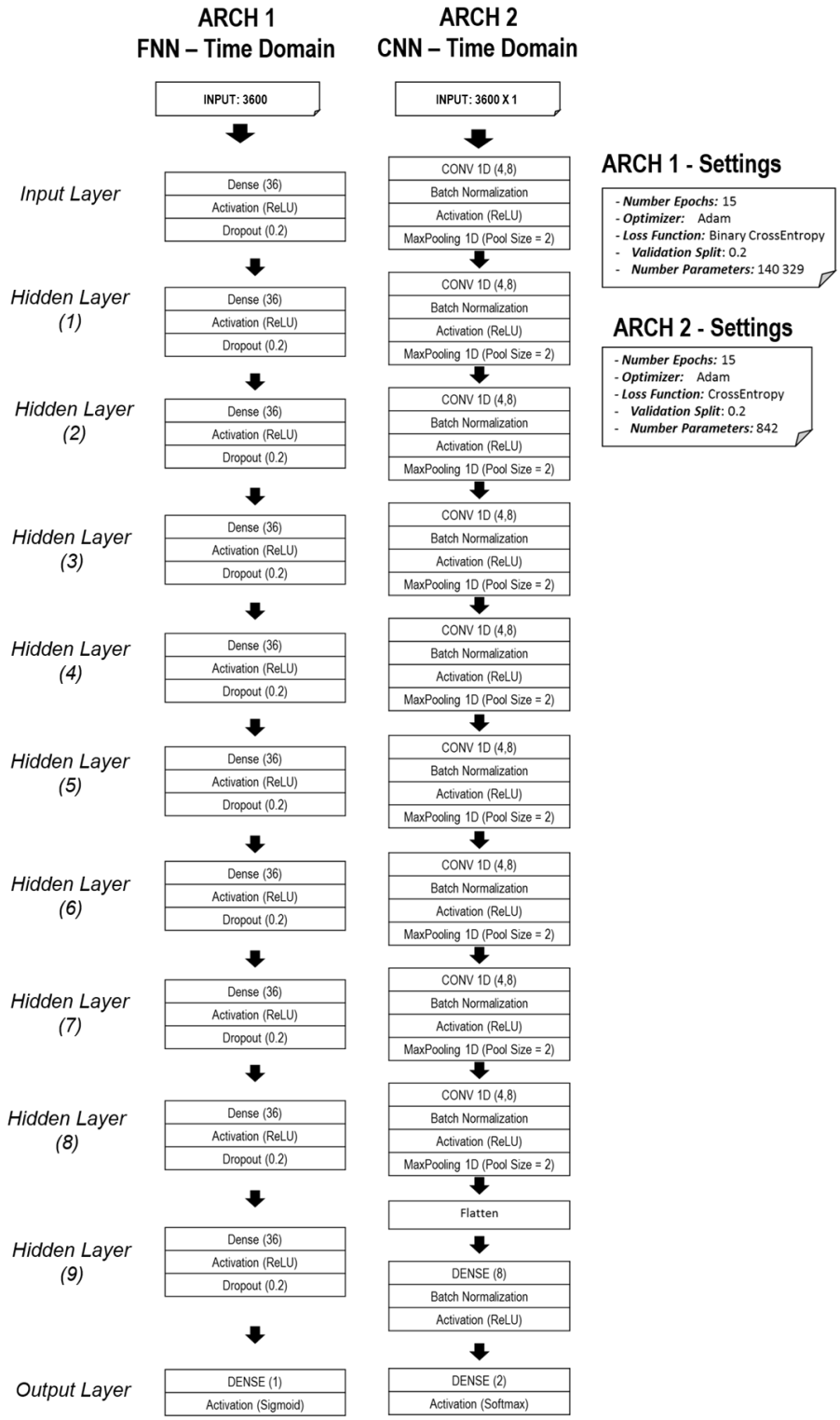

Figure A-2: Deepest ANN models generated for the ARCH-1 (Fully Connected Neural Networks trained using time domain represented data) and ARCH-2 (Convolutional Neural Networks trained using time domain represented data) architecture-domain combinations.

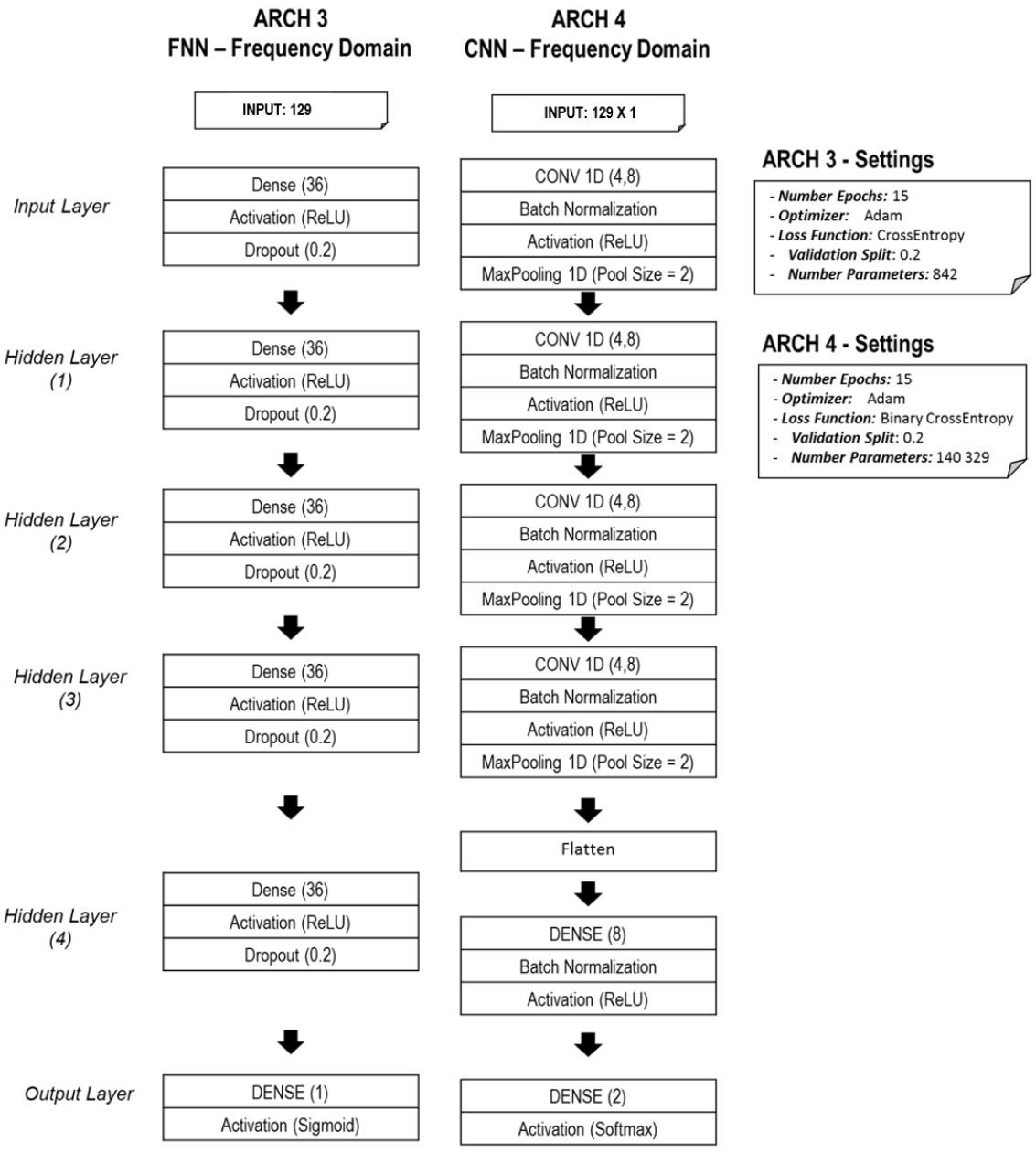

Figure A-3: Deepest ANN models generated for the ARCH-3 (Fully Connected Neural Networks trained using frequency domain represented data) and ARCH-4 (Convolutional Neural Networks trained using frequency domain represented data) architecture-domain combinations

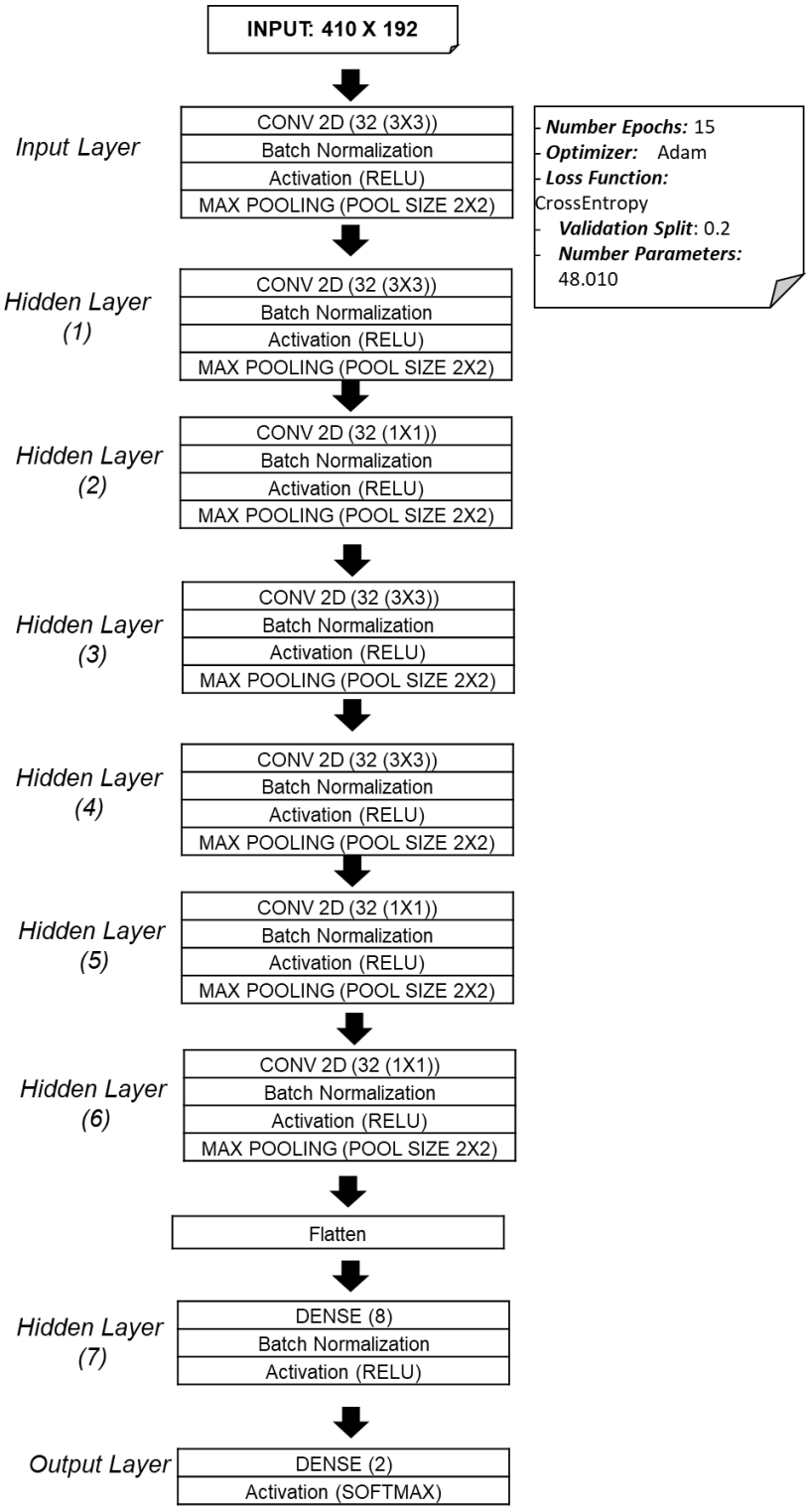

Figure A-4: Deepest model generated for the realistic gravitational waves (Galactic Binaries and Merging Blackhole Binaries) using a Convolutional Neural Network architecture-domain trained with time-frequency domain represented data)

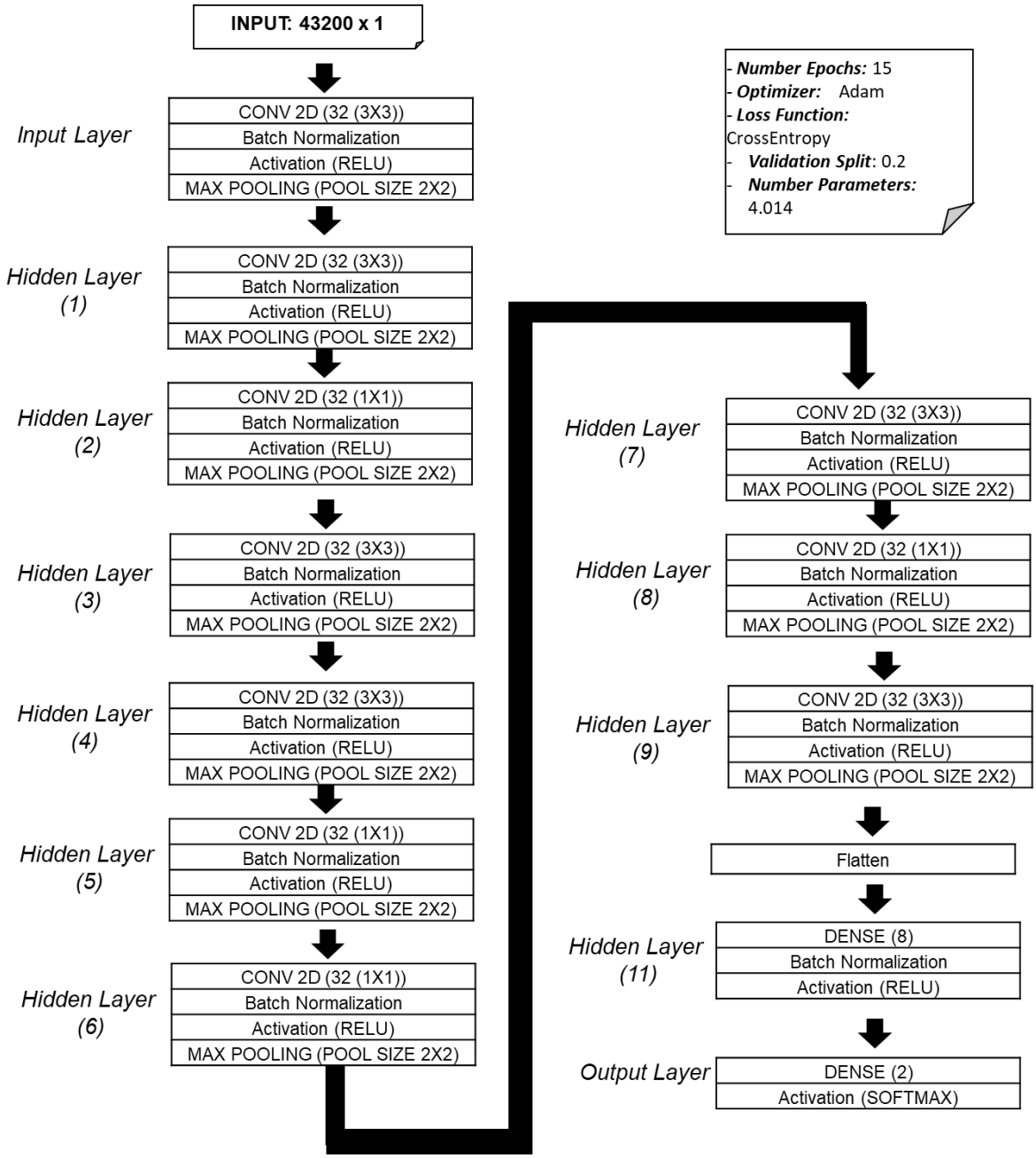

Figure A-5: Deepest model generated for the realistic gravitational waves (Galactic Binaries and Merging Blackhole Binaries) using a Convolutional Neural Network architecture-domain trained with time domain represented data)