



CENTRUM ASTRONOMICZNE IM. MIKOŁAJA
KOPERNIKA
POLSKIEJ AKADEMII NAUK

Applications of machine learning
methods in gravitational-wave
detectors data analysis

Doctoral thesis

Author

mgr inż. Filip MORAWSKI

Supervisors

dr hab. Michał BEJGER

dr Elena CUOCO

All we have to decide is what to do with the time that is given us.
J. R. R. Tolkien

Abstract

The discovery of the first gravitational wave in 2015 marked the beginning of the era of gravitational-wave astronomy. The newly opened window on the Universe allowed us to observe the outer space around us from a completely new perspective. Gravity, the fundamental force in the Universe, has become a valuable source of information on the dynamic processes occurring in space. Space-time undergoes dramatic changes during violent phenomena such as merging black holes or neutron stars as well as supernova explosions. The gravitational wave's propagation distorts the space it passes through. Though the effect diminishes as the wave travels away from the source, the power emitted by dramatic events such as merging black holes exceeds the electromagnetic emission of all stars in the observed Universe!

Detection of such subtle phenomena as gravitational waves requires extremely precise equipment. The instruments that enable the registration of these signals are laser interferometric detectors built within the Advanced LIGO and Advanced Virgo projects. However, the use of extremely sensitive tools that can register gravitational waves does not equal actual detection. This signal is buried deeply in the noise, which at this level is generated practically through everything: the detector itself, human activity such as planes and cars, seismic activity, or atmospheric conditions. Therefore, gravitational wave detection requires the application of a sophisticated analysis of data collected by detectors. Methods such as machine learning, a branch of artificial intelligence that revolutionized computational capabilities in the twenty-first century, appear to be an ideal candidates.

Work on this thesis began in the autumn of 2017, at the dawn of machine learning applications in gravitational-wave astronomy. As a result, the thesis addresses scientific issues that were previously unstudied or studied only to a limited extent. In general, the thesis fits perfectly into the larger picture of extensive research on how machine learning can be used in gravitational-wave astronomy and what it can give us in return. In my thesis, I investigate the potential of machine learning as a novel method of detecting gravitational waves defined in terms of anomalies in the data. In addition, I discuss the use of machine learning in the search of yet undetected signals emitted during supernova explosions and by non-axisymmetric neutron stars. I am also looking into the idea of reconstructing fundamental theoretical equations describing the physics of neutron stars based on what we can observe.

The ongoing and future LIGO-Virgo-KAGRA observing runs are expected to yield more observations of both known and unknown signals. I hope that this dissertation will make a significant contribution to the development of modern methods for studying gravitational waves.

Streszczenie

Pierwsza detekcja fali grawitacyjnej w 2015 roku zapoczątkowała erę astronomii fal grawitacyjnych. Otwarte w tym momencie nowe okno na Wszechświat umożliwiło nam obserwację otaczającej nas przestrzeni kosmicznej z zupełnie nowej strony. Grawitacja – jedno z czterech znanych obecnie podstawowych oddziaływań – stała się cennym źródłem informacji na temat dynamicznych procesów zachodzących w kosmosie. W trakcie gwałtownych zjawisk jak koalescencje (łączenie) się składników układów podwójnych - czarnych dziur i gwiazd neutronowych - czy wybuchy supernowych, czasoprzestrzeń podlega dramatycznym zmianom. Rozchodząca się fala grawitacyjna kurczy i rozciąga przestrzeń, przez którą przechodzi. Choć efekt ten słabnie, gdy fala oddala się od źródła, to wyemitowana w trakcie koalescencji czarnych dziur fala grawitacyjna ma moc przewyższającą emisję elektromagnetyczną wszystkich gwiazd w obserwowanym Wszechświecie!

Detekcja tak subtelnych zjawisk jak fale grawitacyjne wymaga niezwykle precyzyjnego sprzętu. Instrumentami, które umożliwiają rejestrację tych sygnałów są laserowe detektory interferometryczne: Advanced LIGO i Advanced Virgo. Samo uruchomienie tych niezwykle czułych narzędzi, będących w stanie rejestrować fale grawitacyjne, nie oznacza jeszcze faktycznej detekcji. Sygnał ten bowiem ukryty jest głęboko w szumie, generowanym przez sam detektor, wpływ aktywności ludzkiej (samochody, samoloty), aktywność sejsmiczną czy też warunki atmosferyczne (falowanie mórz, wiatr i deszcz). Dlatego też rejestracja fali grawitacyjnej wymaga zastosowania wyrafinowanej analizy zebranych przez detektory danych. Metody takie jak uczenie maszynowe, gałąź sztucznej inteligencji, która zrewolucjonizowała możliwości obliczeniowe komputerów w XXI wieku, wydają się być idealnymi kandydatami.

Prace nad niniejszą dysertacją rozpoczęły się jesienią 2017 roku, u zarania badań nad zastosowaniem uczenia maszynowego w astronomii fal grawitacyjnych. W rezultacie rozprawa porusza zagadnienia naukowe, które wcześniej nie były badane lub badane były tylko w ograniczonym zakresie. Ogólnie rzecz biorąc, dysertacja ta doskonale wpisuje się w szerszy obraz szeroko zakrojonych badań nad tym, jak uczenie maszynowe może być wykorzystane w astronomii fal grawitacyjnych i co może nam dać w zamian. W pracy doktorskiej badam potencjał uczenia maszynowego jako nowatorskiej metody wykrywania fal grawitacyjnych definiowanych jako rodzaj anomalii w danych. Ponadto zajmuję się zastosowaniami uczenia maszynowego w poszukiwaniach jeszcze nie wykrytych sygnałów emitowanych podczas wybuchów supernowych oraz przez osiowo niesymetryczne gwiazdy neutronowe. Badam również koncepcję rekonstrukcji fundamentalnych równań teoretycznych opisujących fizykę gwiazd neutronowych na podstawie tego, co faktycznie możemy zaobserwować.

Trwające i przyszłe kampanie obserwacyjne LIGO-Virgo-KAGRA przyniosą więcej de-

tekcji sygnałów, zarówno tych które już znamy i oczekujemy, a także tych, których nie jesteśmy jeszcze świadomi. Mam nadzieję, że niniejsza praca wniesie istotny wkład w rozwój nowoczesnych metod badania fal grawitacyjnych.

Acknowledgments

First and foremost, I would like to thank my Ph.D. supervisor, Dr hab. Michał Bejger, for his never-ending enthusiasm for scientific research, willingness to test even the most absurd ideas and presence and support throughout my Ph.D. studies. My thanks also go to Dr. Elena Cuoco, my co-supervisor, who has provided me with numerous opportunities to grow while demonstrating how varied academic work can be and how important creativity is. I am confident that the skills I have gained from both of my supervisors over the last few years will make a significant difference in my life in a variety of ways.

I would also like to thank the LIGO-Virgo-KAGRA collaboration, which I am proud to be a part of. I have met many brilliant researchers on behalf of the collaboration over the last few years who were eager to share their expertise and knowledge with me. I would like to express my gratitude to my Italian colleagues, Dr. Alberto Iess and Dr. Barbara Patricelli, for many fruitful discussions and projects completed together. Furthermore, I would like to express my gratitude to Prof. Ik Siong Heng and Dr. Chris Messenger for their warm hospitality and the wonderful times I had in Glasgow. I would also like to thank my colleagues at the Nicolaus Copernicus Astronomical Center from the neutron star and gravitational waves groups for stimulating discussions and an overall enjoyable time at the institute. Dr. Paweł Ciecieląg and Dr. Jonas Pereira deserve special recognition for their advice and assistance.

I would also like to express my gratitude to my family, without whom I would not be where I am today. As I get older, I realize how much I owe to each and every one of you.

My friends are extremely important to me, and I owe them a lot of wonderful and crazy times we have had together. I consider myself extremely fortunate to have met you, Paulina, Adam, Bartosz, Martyna, Kasia, David, Jose, Ago, and Dominik. I could go on and on about how much I admire each of you and how much you mean to me. But first and foremost, please accept my heartfelt gratitude.

Last but not least, I would like to thank the Nicolaus Copernicus Astronomical Center's administrative, secretarial, and IT staff for all of their help whenever I needed it. Finally, I would like to thank the Polish grant agencies, NCN and NAWA, as well as the European COST initiative, for the grants and scholarships that helped me financially and funded research activities during my Ph.D. studies ¹.

¹The work in this thesis was supported by the Polish National Science Centre grants no. 2016/22/E/ST9/00037, 2017/26/M/ST9/00978 and 2020/37/N/ST9/02151, Polish National Agency for Academic Exchange grant PPN/IWA/2019/1/00157, European Cooperation in Science and Technology Action CA17137, NVidia seeding grant, PL-Grid Infrastructure and by the MNiSW grant for expansion of the strategic IT infrastructure for science.

Contents

I	Introduction	1
1	Introduction	3
1.1	Gravitational-wave astronomy	4
1.1.1	Fundamentals of gravitational waves	4
1.1.2	Detection principle	7
1.1.3	Astrophysical sources of gravitational waves	12
1.1.4	Challenges in data analysis of gravitational wave detectors	16
1.2	Machine learning	19
1.2.1	Machine learning paradigms	20
1.2.2	Machine learning algorithms	21
1.2.3	Artificial neural networks	22
1.3	Motivation and thesis overview	33
II	To Expect The Unexpected (Searches for irregularities in the data)	35
2	Paper I: Anomaly detection in gravitational waves data using convolutional autoencoders	37
III	Trouble Is My Business (Classification of difficult signals)	57
3	Paper II: Core-Collapse Supernova Gravitational-Wave Search and Deep Learning Classification	59
4	Paper III: Convolutional neural network classifier for the output of the time-domain \mathcal{F}-statistic all-sky search for continuous gravitational waves	75
IV	Relationship status: it's complicated (Reconstruction of the astrophysical parameters)	91
5	Paper IV: Neural network reconstruction of the dense matter equation of state derived from the parameters of neutron stars	93

V	What's next?	103
6	Summary and outlook	105

Part I

Introduction

Chapter 1

Introduction

Humans have always been drawn to the night sky. Myriads of lights illuminating the sky influenced the lives of poets, scientists, and travelers, shaping human history. We have learned a lot about our surroundings over the centuries. Many questions, however, remained unanswered. Many answers raised new questions, resulting in a never-ending quest to comprehend nature.

Albert Einstein's presentation of general relativity in 1916 [1] was one of the most groundbreaking steps on that journey. The presence of mass, according to general relativity, influences the fabric of space-time. In extreme cases, this influence may result in the formation of singularities surrounded by regions of space-time from which even light cannot escape. Such objects were dubbed black holes at the time. The possibility of the existence of these objects has irreversibly altered our perception of the Universe. The dark abyss of the night sky turned out to be full of structures and processes that previously escaped our attention. Furthermore, general relativity suggests the existence of other unusual phenomena, such as gravitational waves [2], i.e., ripples in the fabric of space-time that have the potential to stretch and squeeze everything in their path, and alter the pace of time.

The theory of general relativity expanded our vision in many previously unanticipated directions. It has, however, presented us with new challenges to overcome. Even though gravitational waves are emitted during one of the Universe's most violent and dynamic processes, they have a negligible impact on the space-time far from the source. The gravitational wave effect on the Earth is occurring on a much smaller scale than anything we have discovered so far. Thus, an attempt to discover it required a huge technological leap forward.

On September 14, 2015, these efforts led to a historic result [3]. The first detected gravitational wave, GW150914, was emitted by a binary black hole system inspiral and merger, ushering in a new era in astronomy. The ability to observe the Universe through gravitational waves allows us to investigate violent, dynamic astrophysical processes in ways that were previously inaccessible. Astronomers now have a new medium for observations that is completely independent of electromagnetic waves thanks to the newly opened window on outer space. We can now reach the interiors of the stars, but we can also peer into the vicinity of event horizons of black holes, which could lead to truly groundbreaking discoveries.

This is only possible with steady advancements in detection and data-analysis techniques. Even though human brains enable us to achieve extraordinary feats, they are not without limitations. Machine learning is one method for breaking through those barriers. In fact, machine learning is currently playing an important role in the advancements of many scientific disciplines. Those novel methods of data analysis have become increasingly popular in astronomical applications in recent years.

My study is therefore extremely timely, as machine learning is thought to significantly improve the data analysis capabilities of existing and future gravitational-wave detection systems. However, this can only happen if we understand what machine learning can and cannot do. The application of machine learning to various aspects of gravitational-wave data analysis is the primary motivation for my research, which is presented in this thesis.

In the following chapter, I begin with a brief overview of gravitational waves (Sec. 1.1.1) and the detection principles (Sec. 1.1.2). I then discuss various sources of waves (Sec. 1.1.3) and the challenges in the data analysis of gravitational-wave detectors (Sec. 1.1.4). The next part of this chapter covers the overview of machine learning in the context of gravitational-wave applications with emphasis on learning paradigms (Sec. 1.2.1) and algorithms studied in the presented thesis (Sec. 1.2.2), in particular artificial neural networks (Sec. 1.2.3). With the motivation of this thesis at hand, I conclude with an overview of the remaining chapters which contain published works (Sec. 1.3).

1.1 Gravitational-wave astronomy

1.1.1 Fundamentals of gravitational waves

Gravity is caused by the curvature of space-time, which in turn is caused by a presence of mass-energy, according to general relativity (GR) [1]. When massive objects (as opposed to e.g. massless photons) move around in space-time, the curvature changes to reflect their motion. When the motion accelerates, the gravitational field begins to fluctuate due to changes in the curvature of space-time. These fluctuations propagate outward from their sources in a wave-like fashion at the speed of light. They are known as gravitational waves (GWs) [2].

As detailed derivations of GW equations can be found in many classical textbooks (such as [4]), I am only going to briefly summarize them here. The equations of GR known as the Einstein Field Equations (EFE), bind together space-time and mass-energy. They consist of ten, coupled, non-linear differential equations that represent the space-time content (curvature) on the left-hand side, and mass-energy content on the right-hand side as:

$$R_{\mu\nu} - \frac{1}{2}Rg_{\mu\nu} = \frac{8\pi G}{c^4}T_{\mu\nu}, \quad (1.1)$$

where the stress-energy tensor $T_{\mu\nu}$ contains the information about the mass, energy, and momentum distribution, and $g_{\mu\nu}$ stands for the space-time metric describing the distance between the infinitesimally-close events; G denotes the gravitational constant, and c the speed of light in vacuum. From the metric, one can derive the Ricci tensor $R_{\mu\nu}$ and the Ricci scalar $R \equiv g^{\mu\nu}R_{\mu\nu} \equiv R^\mu_\mu$, which describe the space-time curvature. The space-time

metric tensor $g_{\mu\nu}$ may be regarded as the core tool of GR, as with it one determines the features of space-time, describing distances, angles, volumes, as well as past and future directions.

Astrophysical objects which are of interest in the presented thesis are dense and compact - in the sense of the ratio of mass M to their size R , compactness of compact objects $2GM/(Rc^2)$ is close or equal to 1, as in the case of black holes (BH) - and as such are causing large curvature to the surrounding space-time. Derivation of $g_{\mu\nu}$ is non-trivial in such a case as the EFE becomes highly non-linear. However, in large distances from those astrophysical objects, the space-time metric can be approximated by a small linear perturbation $h_{\mu\nu}$ addition to the "flat" Minkowski metric $\eta_{\mu\nu}$ (a combination of three-dimensional Euclidean space and time into a four-dimensional manifold). This approximation is also called a weak field limit:

$$g_{\mu\nu} = \eta_{\mu\nu} + h_{\mu\nu}, \quad (1.2)$$

under the condition that the $|h_{\mu\nu}| \ll 1$. The solution to linearized vacuum EFE yields the wave equation for GW propagating with a constant speed of c :

$$\left(-\frac{1}{c^2} \frac{\partial^2}{\partial t^2} + \nabla^2 \right) h_{\mu\nu} = 0, \quad (1.3)$$

where ∇^2 is the 3-dimensional Laplace operator. The perturbation $h_{\mu\nu}$ has ten independent components however they can be reduced to two by introducing the transverse traceless (TT) gauge transformation [5]. In the transformation, we impose the Lorentz gauge (analogous to electromagnetism) and metric perturbations which are only spatial with a trace of the metric being equal to zero. The divergence of the spatial metric perturbation, by the Lorentz gauge, tells us that the spatial metric perturbation is transverse. For more details of TT gauge derivation see [6]. As a result of the transformation, the temporal components and the trace of the perturbation vanish. Far from the source, the perturbation h^{TT} can be written in the form of plane-wave solutions propagating along the z axis:

$$\begin{aligned} h_{\mu\nu}^{TT}(t, z) &= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & h_+(t - z/c) & h_\times(t - z/c) & 0 \\ 0 & h_+(t + z/c) & -h_\times(t - z/c) & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}_{\mu\nu} \\ &= \epsilon_{\mu\nu}^+ h_+(t - z/c) + \epsilon_{\mu\nu}^\times h_\times(t - z/c). \end{aligned} \quad (1.4)$$

The h_+ and h_\times components corresponds to the amplitudes of two independent polarization states of GW. These states are described by the $\epsilon_{\mu\nu}^+$ and $\epsilon_{\mu\nu}^\times$ tensors which are commonly referred to as *plus* and *cross* polarizations (depending on their effect on the test masses). States can be transformed into the other by rotating the wave vector by $\pi/4$ angle as shown in Fig. 1.1.

GW emission from the astrophysical sources can be approximately derived by using a multipolar expansion of the stress-energy tensor $T_{\mu\nu}$ as long as the internal velocities of

the source are non-relativistic [7]. Usually, the GW radiation is dominated by the mass quadrupole moment, resulting in the following relation:

$$h_{ij}^{TT}(t, x) = \frac{2G}{c^4} \frac{1}{r} \ddot{Q}_{ij}^{TT}(t - r/c), \quad (1.5)$$

where r is the distance from the source, and Q_{ij}^{TT} is the transverse-traceless projection of the mass quadrupole tensor, which for a non-relativistic source of mass density $\rho(t, x)$ equals:

$$Q_{ij}^{TT}(t) = \left[\int d^3x \rho(t, \vec{x}) \left(x^i x^j - \frac{1}{3} r^2 \delta^{ij} \right) \right]^{TT}. \quad (1.6)$$

The absence of a time-varying mass monopole or dipole is an important conclusion from the above equations. A comparison with the limit of Newtonian gravity tells us that the monopole radiation is forbidden by mass conservation and the dipole radiation by momentum conservation. As a result, a spherically symmetric collapse or explosion emits no GWs in GR (which relates to the Birkhoff theorem [8]). The other gravitational theories allow for a monopole or dipole radiation [9]. A large, rapidly-rotating, non-axisymmetric mass distribution, on the other hand, has a large time-varying mass quadrupole moment and is an efficient GW emitter.

The space-time perturbation h^{TT} , henceforth denoted by h , which is, in fact, the *amplitude of a GW*, is the fundamental variable measured by the GW detectors, as we will see in Sec. 1.1.2. As shown in Eq. 1.5, the GW amplitude decays as $1/r$ with the distance from the source (which is a direct consequence of the conservation of energy - the flux of energy $\propto 1/r^2$ through the sphere of radius r is constant). However, the energy received at the Earth from even distant sources is significant, e.g. a $10 M_\odot$ BH at a distance of $15 Mpc$ from the Earth results in the energy flux of $3 \text{ ergs/cm}^2\text{s}$ and the amplitude of 10^{-22} (for detailed derivation see [10]).

Derived above GWs have some striking similarities with their electromagnetic (EM) counterparts. First of all, they both transport energy from astrophysical sources. They also share the same speed of wave propagation equal to the speed of light c (in vacuum), as demonstrated by the detection of the GW170817 event [11]. Furthermore, GW and EM waves are both transverse, and have two polarization states, though in the latter case the angle between states is equal to $\pi/2$. From the mathematical point of view, they can be described by similar wave equations.

Despite those similarities, the key difference between GW and EM waves is the way they interact with matter. The EM forces interact significantly stronger with matter than gravitational forces. The coupling constant α_i (where i stands for a type of the force) defining the strength of the fundamental force relative to the strong force ($\alpha_S = 1$) for EM is equal to $1/137$ and for gravity, 10^{-39} [12]. As a result, photons carry information from the moment of the last scattering, which is usually the surface of an astrophysical object whereas GWs allow probing interiors of astrophysical objects. EM waves can also leave behind a significant portion of the information as an effect of interacting with e.g. cosmic dust. GWs, on the other hand, propagate almost unchanged throughout space, even from the most distant sources. They can probe central regions of active galactic nuclei and even neutron stars (NS) interiors. This feature is extremely important as it allows the study of

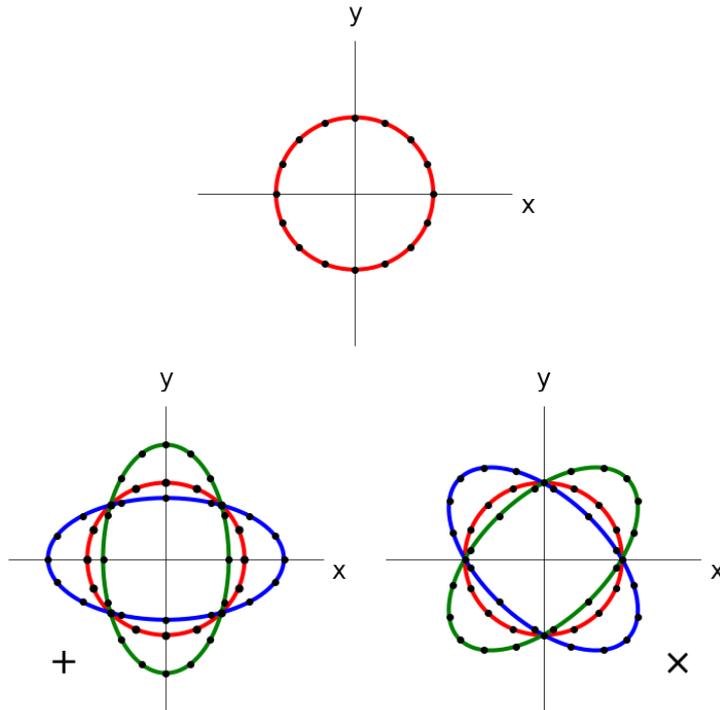


Figure 1.1: The effect of two possible polarization states of GW as seen propagating in the direction towards the reader (z axis) on a ring of test particles: *plus* polarization (left, bottom plot) and *cross* polarization (right, bottom plot). For the comparison, the initial state of the ring of particles has been added (top plot). Unlike EM waves, the GW polarization states are rotated concerning each other by $\frac{\pi}{4}$ (for EM waves the angle between states is $\frac{\pi}{2}$).

violent phenomena that are inaccessible for EM observations, such as BH and NS mergers or Core-Collapse Supernova Explosions (CCSN). These observations will without a doubt open up new chapters in science, allowing for the testing and derivation of various new astrophysical theories.

1.1.2 Detection principle

Since GWs are propagating fluctuations of gravitational fields in space-time, far from the source they behave like waves affecting objects in their path by tidal gravitational forces that act in directions perpendicular to the direction of wave propagation (the transversality condition). These forces cause the distance between neighboring points to change. The magnitude of the change is proportional to the distance between points. Distance is stretched in one perpendicular direction, while it is squeezed in the other (which is related to the traceless condition). A depiction of this effect on a ring of test particles is presented in Fig. 1.1. Detecting space-time distance changes between neighboring space-time events due to the GW tidal field is in a nutshell the GW detection principle [13].

To detect the GW on the Earth, the detector's antenna pattern functions (F_{\times} and F_{+}) must be considered. These functions describe the sensitivity of the instrument to wave polarization. The detector's response $h(t)$ is then defined as follows ([14–17]):

$$\begin{aligned}
h(t) &= F_{\times}(t)h_{\times}(t) + F_{+}(t)h_{+}(t) \quad \text{with} \\
F_{\times}(t) &= \sin(\zeta) (a(t) \cos(2\psi) + b(t) \sin(2\psi)) \\
F_{+}(t) &= \sin(\zeta) (b(t) \cos(2\psi) - a(t) \sin(2\psi)).
\end{aligned} \tag{1.7}$$

F_{\times} and F_{+} are periodic functions with the period equal to one sidereal day for detectors located on the surface of the Earth (due to the rotation of the planet). They also depend on the wave polarization angle ψ , the angle ζ between the detector's arms (in case of interferometers as described later in this chapter), and two amplitude modulation functions $a(t)$ and $b(t)$ that depend on the location and orientation of the detectors on the Earth and the position of the GW source on the sky (detailed information can be found e.g. in [17]).

In the literature, the change in distance between test particles is frequently denoted as ΔL over the initial distance L . The ratio of those values is the GW amplitude h , also known as the *GW strain*:

$$h = \frac{\Delta L}{L}. \tag{1.8}$$

As mentioned in the previous section, the GW strain has an extremely small amplitude. A displacement between particles it causes is orders of magnitude smaller than the size of a proton e.g. the very first detected GW [3] caused $\Delta L = 10^{-18} \text{ m}$ over a distance of $L = 4 \text{ km}$. Extremely sensitive experiments must be designed to directly detect such a small ΔL or even smaller. To simplify the problem, we can either focus on a specific source of GW and its corresponding frequency, or we can design a large-scale experiment for more generic studies. Both approaches were used in the past, but only one resulted in successful GW detections.

Historically, the first indications for the physical existence of GWs came from astronomical observations in the 1960s, notably from studies on the evolution of white dwarf systems [18]. The observed orbital decay of the Hulse–Taylor binary pulsar [19], on the other hand, is widely regarded as the first indirect evidence for the existence of GWs. As energy is lost to gravitational radiation, the results of the subsequent analysis of the decay agreed with GR predictions [20]. However, I focus on the direct detections of GWs in this thesis, as I believe they are more informative.

Resonant (or bar) detectors [21] were the first experiment to directly search for GWs. The devices were made up of several aluminum cylinders to which piezoelectric sensors were attached. The passage of GW through the detector caused the bar to vibrate at the wave's resonant frequency. The sensors could detect the vibration and convert it into electrical signals. It was possible to reconstruct the GW waveform by monitoring that vibration. Despite initial detection claims, no convincing proofs were demonstrated. Resonant detectors were superseded over time by different instruments as they were insufficiently sensitive and too susceptible to various noise sources which limited their usefulness. However, some of them are still operating and searching for signals from the Universe in the narrow frequency bandwidth [22–25].

The large-scale experiments that lead to the first detection of GWs were laser interferometers (IFO) [26, 27]. The concept of interferometry is attributed to Albert Michelson and

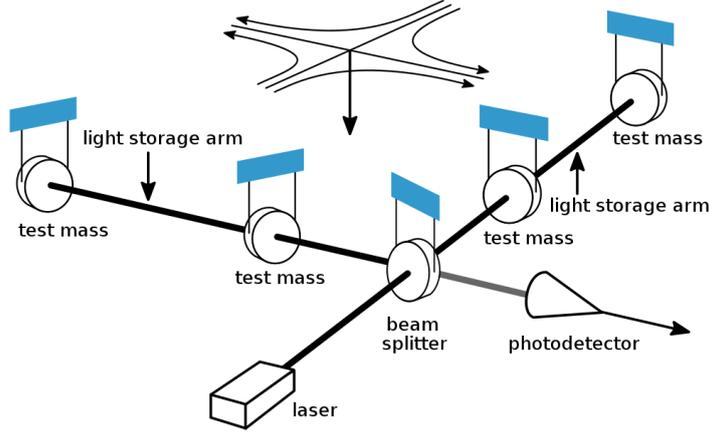


Figure 1.2: The GW interferometer schematic (from [30]). The beam splitter divides the light emitted by the laser into two perpendicular beams that bounce back and forth from the test masses in each of the light storage arms (Fabry-Perot cavities). The arm's bouncing allows the light's effective length to be multiplied hundreds of times. After traveling the specified distance, the light returns to the beam splitter via semi-transparent mirrors (internal test masses) and is collected by the photodetector. When a GW passes through (or when various noise sources are present), the collected light becomes out of phase, resulting in the appearance of interferometric fringes.

dates back to the late nineteenth century. With his colleague, Edward Morley, Michelson experimented using their IFO aiming to prove or disprove the existence of "Luminiferous Aether" – a substance at the time thought to permeate the Universe. Their IFO compared the lengths of two perpendicular arms using light. The result of the experiment was negative and is considered as one of the first steps towards the derivation of special relativity according to Einstein [28]. Though Michelson and Morley are considered the inventors of IFO, the interference fringes were observed and explained already in seventeenth century by Robert Hooke [29].

The IFO used in the GWs searches operates on the same principle as Michelson and Morley device. However, the key point in the measurements of GW is that the speed of light is always constant in the given medium. Moreover, GW IFOs are much more sensitive to detecting sub-atomic changes at arm's length. Figure 1.2 presents the schematic of the detector.

The arms of the GW IFO are defined by test particles - mirrors suspended by a system of intricate seismic isolation (called pendulums), allowing them to be as free-falling as possible on the surface of the Earth, i.e. they are free to move horizontally in response to passing GW. The arms are the same length in the absence of a wave, and light from one arm returns with the same phase as light from the other. However, in the presence of GW, the lengths of the arms change: one expands, while the other shortens. As a result, the light that returns will be out of phase, resulting in interference fringes between the two beams. The strain h in the case of IFO is measured by a phase shift between the two beams caused by passing GW. The displacement between arm's lengths for the known wavelength of the light used in the IFO can be computed based on the number of fringes.

A few additional upgrades have been implemented to improve IFO's ability to detect GW.

To begin with, the light in each arm is stored in the Fabry-Perot cavities [31], allowing the light to stay for more than one reflection. As a result, the light gains an additional phase shift with each reflection. The effective length of the IFO arm can increase many times over multiple bounces (which may be of the order of hundreds), as can the phase shift.

Furthermore, the GW IFOs' arms are 3-4 km long. Because GW is caused by tidal forces, the GW-induced phase shift of the light can be increased. The longer the arm, the greater the distance between test particles (mirrors) and, consequently, the displacement ΔL , according to Eq. 1.8. If the IFO is optimally oriented in relation to the source of GW, the length of the other arm will change by the same amount but in the opposite direction as the first arm.

Taking both of the aforementioned upgrades into account, the final phase shift of the light is significantly larger than in the case of Michelson-Morley IFO. However, each upgrade comes at the expense of additional noise sources, which, at the sensitivity level of GW IFO, have a significant impact on GW searches.

Several major noise sources influence IFO sensitivity: the gravity gradient noise, thermal noise, seismic vibrations, and shot noise (detailed descriptions of each noise source are available in [32]). Each of the sources is related to a different physical phenomenon and affects the detector's sensitivity in a different frequency range.

Gravity gradient noise is the first fundamental noise source in the IFO. Purely Newtonian gradients of local gravitational fields can be detected by GW detectors (i.e. they are responsible for moving the mirrors). These gradients are related to environmental factors such as atmospheric pressure changes or density waves in ground vibrations. Ground-based detection of GW is impossible below 1 Hz due to the strength of the gravity gradient noise. Unfortunately, the ground detectors cannot be screened out of this effect. A space-borne detector would be free of the effect of gravity gradient noise. Alternatively, one can monitor the changes of the gravity gradient and correct the mirrors' positions accordingly.

Thermal noise is produced by the vibrations of atoms in the IFO mirrors, suspension wires, and pendulums which for current generation detectors - Advanced LIGO and Advanced Virgo - are at room temperature. There are two methods for controlling this source of the noise. The first solution is to use cryogenic systems to cool the mirrors and pendulums to extremely low temperatures as the vibrations of those instruments are significantly reduced in those conditions. The second option is to take advantage of the fact that the amplitude of thermal noise is greatest at the frequency of the vibrational mode of the materials used to make mirrors and pendulums. It is possible to reduce IFO sensitivity in a narrow frequency band corresponding to the vibrational mode and significantly increase it outside of that band with proper design. The vibrational mode of modern GW IFO is around 1 Hz for pendulums and above 1 kHz for mirrors. For ground-based IFO, both frequency ranges are outside the GW observing band.

Seismic noise is caused by constant movements of the ground beneath the detector. This source of noise has a frequency range of less than 100 Hz . The IFO mirrors are suspended on pendulums, which function admirably as mechanical filters above the pendulum frequency. The suspensions are unaffected by frequencies greater than the pendulum frequency. Because suspensions are designed with a pendulum frequency of around 1 Hz ,

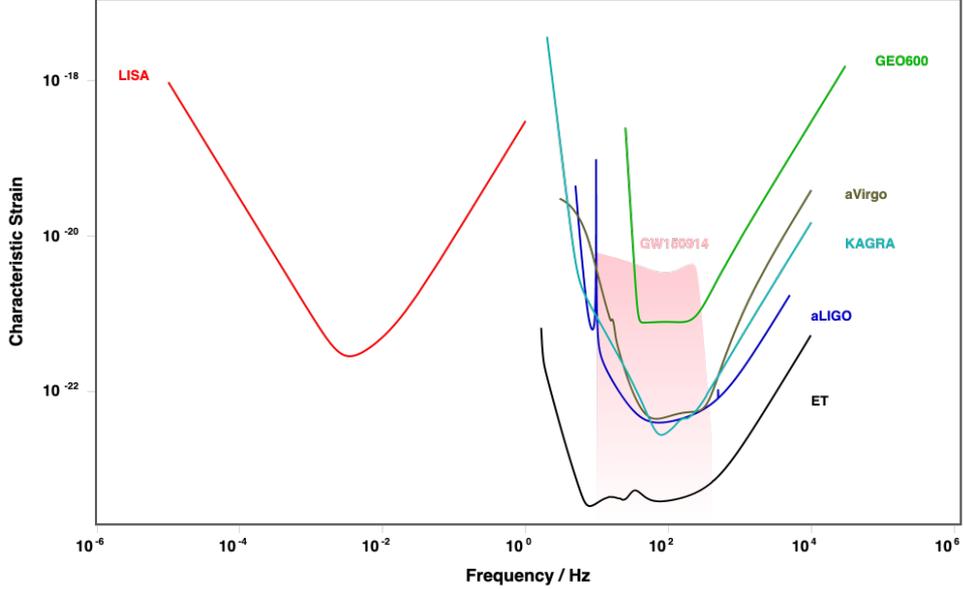


Figure 1.3: The comparison of the characteristic strains [35] as a function of GW frequency for existing and planned detectors (plot generated with an interactive tool at <http://gwplotter.com>). In comparison, the pink outline represents the GW150914 signal strain.

seismic vibrations reduce sensitivity below the GW observing band. Furthermore, IFO sites are outfitted with seismometers to warn GW detectors of the dangers of stronger vibrations such as earthquakes.

The quantization of photons in the laser beam causes the *shot noise*. Photons arrive at random times at the photodiode, causing random fluctuations in light intensity (interference fringes). The resulting signal may resemble GW, especially above 200-300 Hz , which corresponds to the frequency band of the shot noise. More photons in the beam are required to control this source of noise, which translates to stronger lasers. However, no lasers exist that can continuously provide enough light to keep the interference fringes smooth. Lasers with hundreds of kilowatts of power are impossible to build. As a result, GW IFO employs a different technical solution known as power recycling to increase the power of the light in the arms [33, 34]. A second mirror is placed between the laser and the beam splitter. This mirror reflects wasted light from one of the IFO's arms in such a way that it adds coherently in phase with laser light. This squandered light is recycled until the required power levels in the arms are met. Power recycling cannot indefinitely increase light power because some energy is lost during absorption and scattering on various mirrors in the IFO. However, it still allows for a reduction in the required laser power to feasible values (see [26, 27] for more details about power recycling in Virgo and LIGO IFOs).

All these prerequisites lead us to the discussion of the *sensitivity curve* of an IFO. The sensitivity curve is the primary tool for characterizing the GW detector utility. It allows for the investigation of the effect of various noise sources on the detection of GW concerning frequency bands. More importantly, the curve is used to determine whether or not a GW source can be detected. The sensitivity curve plots the strength of the signal required to be detectable as a function of GW frequency. It is expressed either as the

square root of the power spectral density S_f of the noise, or the characteristic strain $h(t)$ as a function of the GW frequency f :

$$h_f = \sqrt{S_f}. \quad (1.9)$$

Figure 1.3 shows a few examples of sensitivity curves for existing and planned GW IFOs. aLIGO [26] stands for The Advanced Laser Interferometer Gravitational-Wave Observatory; it consists of two independent detectors located on the continental United States: in Hanford (Washington) and Livingston (Louisiana). Both Advanced LIGO IFOs have 4 *km*-long arms and are currently the most sensitive GW detectors in operation. The Advanced Virgo (aVirgo) [27] is a European detector located in Cascina, near Pisa in Italy. It has 3 *km*-long arms and specializes in sophisticated suspensions and seismic noise control.

KAGRA (Kamioka Gravitational Wave Detector) [36] detector is the newest IFO located in Gifu prefecture in Japan. It has also arms of 3 *km* in length. KAGRA specializes in cryogenic cooling of mirrors and pendulums to reduce thermal noise. It is also located underground (in an old mine) which partially screens vibrations from the ground.

GEO600 [37] is located near Hannover, in Germany. Although its arms are 600 *m* long, it is not primarily used for astronomical observations. GEO600, on the other hand, is intended as a testbed for advanced detector techniques such as signal recycling, resonant sideband extraction, and monolithic mirror suspensions.

Soon many ground-based GW IFOs are planned to be constructed. Among them are LIGO-India [38], NEMO [39], LIGO Voyager [40], Cosmic Explorer [41] and Einstein Telescope (ET) [42], the last to be located a few hundred meters below ground to reduce low-frequency noise sources. Moreover, ET is designed as a triangular detector with three 10 *km* long arms inclined at a 60-degree angle. As a result, the sensitivity of ET across a wide frequency range will be significantly improved compared to currently operating IFOs, resulting in a large number of GW detections.

The Laser Interferometer Space Antenna (LISA) [43] will be the first space-based GW IFO, with a launch date of 2030. LISA has been designated as a European Space Agency (ESA) Cornerstone mission for the twenty-first century. LISA, like ET, will consist of a triangular array of spacecraft with arm lengths of 5×10^6 *km* orbiting the Sun in Earth's orbit. LISA will be sensitive in a range of 0.3 *mHz* to about 0.1 *Hz*, and will not be affected by gravity gradients or seismic noise. With such a frequency range, it will be able to detect GW from binary star systems in our Galaxy as well as mergers of supermassive BHs anywhere in the Universe.

1.1.3 Astrophysical sources of gravitational waves

GR specifies the conditions that astrophysical objects must meet in order to emit GWs, namely to exhibit accelerated mass motion and to possess a time-varying mass quadrupole moment (Eqs. 1.5 and 1.6). According to the current astrophysical state-of-art, GWs are therefore emitted by a variety of objects. Many of them have GW frequency bands that are within the sensitivity range of current and planned IFOs. In the following chapter, I will attempt to classify four types of GW sources of the greatest scientific interest due to their current detectability or anticipated detection potential in the near future.

Compact binary systems

The GW emitted by binary systems has been one of the most widely studied sources of GWs for decades. Even before the era of IFOs, they provided the best proof of the dependability of GR for GW. The binary pulsar discovered in 1974 [19] was one of the first examples of an indirect discovery of GW (the other being system of white dwarfs by [18]). The emission of GW in that system causes a decrease in the orbital period, which agrees exactly with GR. Many compact binary systems as such should be detectable by LISA in the near future ([44] and references therein).

As a result of GW emission, the binary system is losing energy and decreasing the distance between components, eventually leading to the merger. As the orbit shrinks, the frequency and amplitude of GW increase. If such an effect is observed during a single observation, the system is said to *chirp*. The derivations of physical properties of the chirping system can be found in e.g. [45–47]. Its name comes from a characteristic time-frequency relation, which, at a Newtonian level, is:

$$f_{GW}^{-8/3}(t) = \frac{(8\pi)^{8/3}}{5} \left(\frac{GM}{c^3} \right)^{5/3} (t_c - t), \quad (1.10)$$

where t_c is the coalescence (merger) time, and \mathcal{M} is a function of component masses m_1 and m_2 , called the *chirp mass*:

$$\mathcal{M} = \frac{(m_1 m_2)^{3/5}}{(m_1 + m_2)^{1/5}}. \quad (1.11)$$

Because the frequency of the shrinking system is determined by the component masses and their separation, it determines the frequency range for detecting the corresponding types of astrophysical objects. Fortunately, ground-based detectors operate in the frequency regime that corresponds to the frequency range of BHs and NSs chirping systems [48], referred to in the literature as *Compact Binary Coalescence* (CBC). Furthermore, because the amplitude of the GW is affected by the system masses, the currently operational IFO are sensitive enough to detect CBC signals (though the amplitude is also affected by the distance to the source - see [46] for details).

According to the first and second gravitational wave catalogs (GWTC-1 and GWTC-2) [49, 50] all of the 50 confirmed GW detections to date have been chirping binaries. The majority of them (48 detections) are attributed to BH binary system mergers. The other two detections are associated with NS binary system mergers. The number of catalog events has been increased by the discovery of two BH-NS merges [51]. Figure 1.4 shows two sample detections: GW150914 - the first GW detection of merging BHs in both LIGO IFOs [3] (top plots) and GW170817 - the first detection of merging NSs in the LIGO-Livingston [52] (bottom plot). Both plots are presented as spectrograms that show how the GW frequency and amplitude change over time. The time-scale of presented events is different as BBH GW lasts a fraction of a second in the sensitivity frequency range of the detector compared to BNS GW.

The waveforms of CBC GWs are accurately predicted in both the inspiral and merger phases due to a theoretical and numerical understanding of the GR solutions. As a result, we can use matched filtering to look for specific waveforms in the IFO data (as discussed in Sec. 1.1.4). Furthermore, by determining which template corresponds to the astrophysical signal, we can infer the physical parameters of GW emission sources.

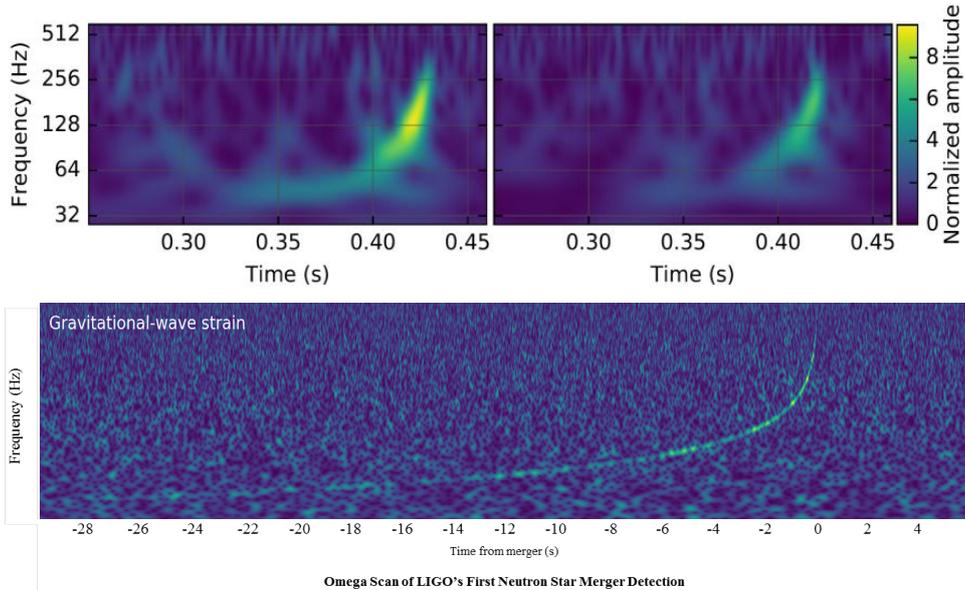


Figure 1.4: An example of two CBC detections: GW150914 - inspiral and merger of two BHs (top plots, [3]), and GW170817 - inspiral and merger of two NSs (bottom plot, [52]). The GW’s characteristic *chirp* corresponds to a rapid increase in frequency (y -axis) as well as amplitude (the brighter the color, the stronger the signal).

Gravitational collapse

The gravitational collapse of a massive star ($M \gtrsim 8M_{\odot}$ on the Zero Age Main Sequence), which results in a supernova explosion (referred to as Core-Collapse Supernova or CCSN) is a promising potential future GW source, which has yet to be observed [53–55]. Unfortunately, it is also one of the least-understood GW sources, making its discovery all the more difficult. The EM observations do not allow for the investigation of the interior of a collapsing star. Moreover, CCSNs are rare with rates for our Galaxy estimated to be around 1 per 100 years [56, 57]. As a result, when studying this phenomenon, we must make educated guesses about what processes are taking place.

Current theories suggest that CCSN is caused by the gravitational collapse of a degenerate core of an evolved star [58]. Explosions should produce either BH or NS depending on the progenitor mass. The gravitational collapse of a star releases a considerable amount of energy (of the order of 10^{53} erg [59], one order of magnitude smaller than the rest mass-energy of the Sun), which is mostly carried away by neutrinos. The amount of energy released in the form of gravitational radiation, on the other hand, is unknown, as it depends on the unknown magnitude of the time-varying quadrupole moment.

To estimate the amount of energy carried away by GWs, realistic 3-D simulations of gravitational collapse with various physical phenomena occurring during the process are performed [60–62]. Nuclear reactions, hydrodynamics, as well as photon and neutrino transport must all be included in such simulations. Moreover, to include gravitational radiation, the collapse’s non-spherical asymmetry must be taken into account, as perfectly spherical motion does not result in GWs emission in GR. It is a difficult task because we cannot predict the amount of asymmetry during the collapse. According to current models (see [63] and references therein), the asymmetry is related to the dynamical instabilities in the rapidly rotating core of the NS progenitor. Several potential instability mechanisms

are proposed such as standing accretion shock instability [64]. Various mechanisms and related models of CCSN explosions result in significantly different GW waveforms. The common aspect of the simulations is the transient character of the GW signal which lasts for a short period in the sensitivity frequency range of the detector.

Non-axisymmetric neutron stars

As mentioned in previous sections, NS are important ingredients in transient GW sources: last moments of binary systems (NS-NS and NS-BH inspirals and mergers), and collapsing massive stars (at the moment of their birth). However, NS may also contribute to persistent (long-lasting) GW with almost-monochromatic GW frequency, called continuous waves (CW).

The emission of GWs involving NS could be caused by a variety of mechanisms, including *r-modes*, crust deformation, or accretion from the companion star (see [65] for a review). The common component shared by those mechanisms is the signal's long duration (years time scale) and low amplitude. As a result, detecting GW requires long periods of continuous observation, potentially lasting months or even years. Nonetheless, using sophisticated matched filtering methods such as \mathcal{F} -statistic method [66], it is possible to detect the GW signal using currently operational ground-based IFOs.

The *r-modes* are a type of oscillations modes present in rotating NSs. The emission of gravitational radiation causes these modes to become unstable. For the description of mechanisms causing oscillations see [67]. *r-modes* are found in young NS that are thought to be hot and fast rotating. Because GW carries energy (and angular momentum), the NS slows the rotation and causes it to become stable. During this process, the star's spin can be reduced from kilo *Hz* to 100 *Hz* which corresponds to the most sensitive ground-based detectors frequency range (see Sec. 1.1.2). The energy carried away by GW from spinning down NS should be detectable by LIGO and Virgo IFOs as long as the NS is relatively close to the Earth (for example Virgo cluster) [68].

The next potential mechanism of GW radiation by NS is related to the accretion of matter from the companion star in systems like the low-mass X-ray binaries (LMXB) [69]. The accreting material may cause a density gradient in the star's deep crust. This, in turn, would result in mass asymmetry and gravitational radiation [70]. As a result, the NS could radiate gravitationally as much energy as it accretes making such stars a stable source of GW for as long as the accretion lasts.

Accretion can also cause deformation of an NS crust in the form of small *lumps* or *mountains* putting a strain on the crust [71]. Because such a star would have been spherically asymmetric, it could have radiated away energy in the form of GWs. Millisecond pulsars are one candidate for such a GW source. They are thought to have a small enough spindown to withstand the significant strain on the crust.

The misalignment between the global dipole magnetic field axis and the rotation axis of the NS leads to an asymmetry in the magnetic field distribution in the interior of the star [72]. This causes the time-varying quadrupole moment of the star responsible for the emission of GWs as a result of magnetic stresses. However, such signals are too weak to be detected by currently operating detectors. Nevertheless, with future improvements of IFO, GWs could be used to probe the magnetic fields of NS.

Gravitational waves background

The gravitational waves background (GWB) is the final type of gravitational radiation we expect to detect [73]. This background could be made up of GWs emitted by a variety of sources. The Big Bang is the most intriguing of them all. As the most violent event in history, it may have generated a significant amount of gravitational radiation, which could have formed a background similar to the Cosmic Microwave Background [74]. Other phenomena, such as the expansion of the Universe, may also leave a mark in the form of gravitational radiation.

The GWB can also be made up of early Universe chirping binaries. Those sources are too far apart for us to distinguish one from the other. Instead, the overlapping GWs could produce a common background detectable by even currently operational IFOs. The GWB strain is much smaller than the signal produced by nearby chirping binaries. However, by studying the output of many detectors coherently, we could be able to identify GWB as being present in all of IFOs.

Other sources

As previously stated, different sources of gravitational radiation do not exhaust all possibilities. Transient GW signals might be emitted during magnetar bursts [75] and by fast radio bursts [76]. Moreover, exotic objects like dark matter or gravastars may also emit GWs in unexpected ways. The Universe may also contain previously unknown astronomical objects to be discovered by GWs. The future of GW astronomy may yield astounding discoveries.

1.1.4 Challenges in data analysis of gravitational wave detectors

Searching for such weak signals as GWs in the noise-dominated data is a challenging task that requires sophisticated, precise data analysis methods. To aid in this task, GWs have been classified into four types of signals concerning their astrophysical source. This division was achieved by applying two criteria: signal duration and waveform knowledge. As a result, the four types of GWs listed below, along with their corresponding astrophysical sources, were formulated:

- short duration and known waveform - chirping binaries,
- short duration and unknown waveform - core-collapsing stars,
- long duration and known waveform - oscillating (or deformed) NSs,
- long duration and unknown waveform - GWB.

Each of these types of GW signals requires specialized data analysis methods, which can be classified as follows:

- matched filtering (short and long signals, known waveforms),
- excessive power (short signals; known and unknown waveforms),
- cross-correlation (unknown waveforms; duration not relevant).

Matched filtering (MF) is a signal processing method that searches for known waveforms embedded in noise (for example Wiener filter [77]). In GW astronomy, the MF algorithm scans the data using a bank of GW templates [78, 79], in extreme cases comprised of millions of waveforms, looking for a best-matching template, e.g., the MF algorithm returns the Signal-to-Noise Ratio (SNR) for a given template by maximizing the likelihood function. Out of all the templates in the bank, the one with the highest SNR is considered in further studies, as it most closely resembles the actual GW signal buried in the noisy data. The template bank spans a large astrophysical parameter space since we do not know *a priori* the true GW parameter values. Depending on the GW type, the parameter space might cover mass components and/or spin values (for CBC) or frequency and its derivatives (for oscillating NSs). Also, the information about the sky position of the potential source might be included - the search is performed then in the so-called *directed* mode. When more parameters are known (for example frequency), the *targeted* search is performed. Whereas, the most general type of search that makes no assumptions about parameters is called *all-sky*. The waveform fitting the data to the greatest extent can later be used to infer astrophysical parameters of the source, in the so-called parameter estimation process.

Among currently used implementation of the MF, the LIGO-Virgo-KAGRA Collaboration uses the PyCBC [80] and the GstLAL [81] pipelines for short duration GWs. Long-lasting signal searches, on the other hand, use pipelines such as TD-Fstat [82] (and other, see recent CW all-sky papers for a list of references [83]).

Despite its numerous advantages, MF has also significant disadvantages. In most cases, it is not suitable for use in low-latency analysis, which is critical in the context of rapid alerts for the multi-messenger astronomy (observation of single astronomical objects with various observatories around the world using both electromagnetic and gravitational detectors), due to the large parametric space to cover and the requirement of large computational resources.

Instead of focusing on specific waveform characteristics, alternative methods to MF look for GWs by measuring an excessive power in the time-frequency domain present in a coherent manner between multiple independent detectors. An example of such a method is the coherent WaveBurst (cWB) [84, 85]. cWB uses wavelet decomposition [86] to explore the data to find generic GW signals with weak (or none) assumptions on the source model. However, the method requires a network of detectors operating simultaneously. This is not always granted because the GW detectors operate autonomously and maybe "offline" for a variety of reasons. In theory, it is possible to use only a single detector dataset by shifting it in time and measuring the coherence between the original and time-shifted data. However, the excessive power methods in this approach are not as precise as in the case of the network of detectors (though this can be improved with help of machine learning [87]).

Alternative search methods utilizing multiple data streams are based on cross-correlation. If a signal resulting from the same GW is present in multiple detectors (with a delay related to particular sky localization of the source), it should be possible to extract it by cross-correlating the data stream outputs. Originally this search method was developed for the GWB [88, 89]. However, the cross-correlation can be tuned to search for different GW sources e.g. long-lasting periodic signals with known sky-localization and polarization emitted by NSs, as in the case of Scorpius X-1 NS [90].

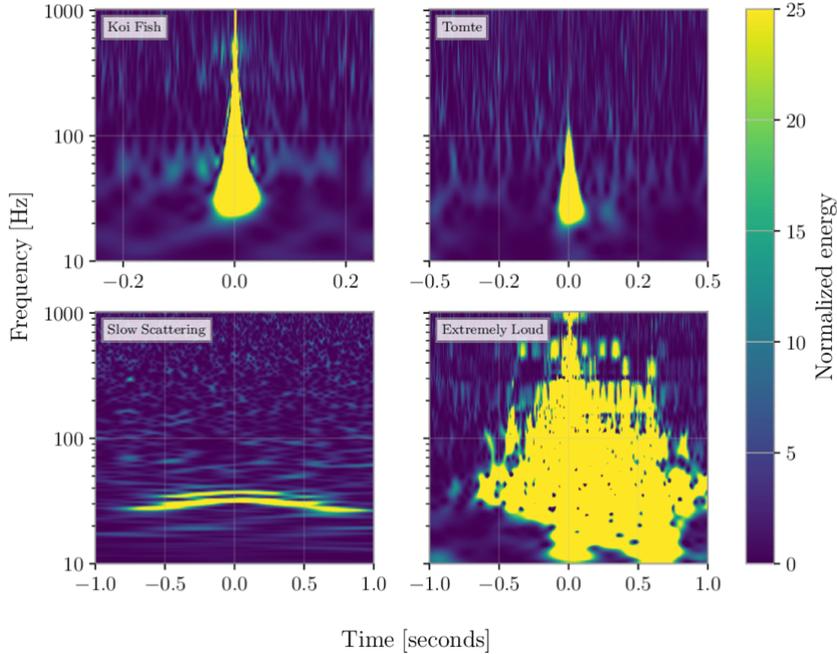


Figure 1.5: A selection of transient noise categories (glitches) in GW strain data [93]. Because of their complex morphologies, glitches are difficult to study. They differ in terms of signal duration, frequency range, and amplitude. Furthermore, some glitches may mimic GWs (for example, chirping binaries), resulting in false detections. They may also overlap with GW, complicating analysis of the underlying astrophysical signal, for example, in terms of estimation of the GW source parameters.

Searching for signals as weak as GWs is further complicated by a plethora of noise sources, both instrumental and environmental. These sources can cause changes in IFO arm lengths that are comparable to or greater than true astrophysical signals. This, in turn, reduces the quality of the data that the algorithms mentioned above must deal with.

Detectors contain state-of-the-art hardware solutions to isolate core instruments from the environment to reduce the influence of various noise sources. Furthermore, IFOs collect information from thousands of "auxiliary (witness) channels" monitoring the immediate surroundings as well as internal instrumental behavior. Seismometers, microphones, magnetometers, accelerometers, probes of various hardware components, etc. are examples of auxiliary channels. Later on, the auxiliary data may be used in the localization of the noise source, improving the overall data quality. However, due to complex noise morphology and variability of detector output signal caused by environmental and hardware issues that arise daily, localization of some noise sources is frequently not possible.

Local noise artifacts, similarly to GWs, can be classified based on their duration. Transient, non-stationary detector artefacts are traditionally called *glitches* ([91–94] and references therein, not to be confused with pulsar glitches). These noises can mimic or overlap transient, astrophysical signals complicating the process of detecting and inferring the physical properties of GW sources [95, 96]. Figure 1.5 presents a selection of common glitch types in the form of time-frequency spectrograms.

Stationary lines are persistent noise signals that impede searches for long-lasting GWs, such as those emitted by non-axisymmetric NSs. They appear at a specific frequency,

reducing detector sensitivity in the nearby frequency bands.

MF, excessive power, and cross-correlations methods are not the only ones used in GW astronomy, despite their popularity. With each passing year, artificial intelligence (AI) methods are used more often. The following chapter provides details about AI, with a focus on its application in GW searches.

1.2 Machine learning

Machine learning (ML) is a subfield of AI - a domain of computer science that studies systems that can approximate "natural" (e.g. human) intelligence [97]. Intelligence can be defined in terms of knowledge gained about specific problems, such as image-based cat vs. dog classification, which is a classic example. Algorithms must learn such knowledge from the data presented in order to acquire it. As a result, ML systems do not require pre-programming. Instead, they use algorithms that can utilize their own intelligence.

There are numerous ML algorithms available today, ranging from a linear regression through decision forests to biologically-inspired artificial neural networks (more in Sec. 1.2.2). Their common feature is that they use data to look for patterns that can then be used to solve problems like classification or regression, which are discussed later in the chapter. Preliminary assumptions about the relationships in the data are not required by ML algorithms. Instead, the relationships are taught during *training* stage. The patterns discovered can then be used to analyze previously unknown data. The accuracy of such prognostics is determined by two factors: the amount of data used during algorithm training and the degree of balance of individual cases in the analyzed data set. Data that does not meet these criteria may result either in *overfitting* or *underfitting*. In the first case, an algorithm learns "by heart" - even if it performs correctly during training, it is unable to generalize [98] to new, previously unknown cases. In the case of underfitting, the algorithm is unable to capture the relationship between the input and output data, resulting in a high error rate for both the training and subsequent *testing* stages.

Furthermore, the proper application of ML requires a precisely defined goal. The data used to train the ML algorithm should include a statistically representative sample of the problem under consideration. For example, an algorithm designed to classify spectra from three different galaxy types will be unable to distinguish the fourth or fifth type in the future because no information about them (more specifically, no information about the existence of an additional class) was collected during the training stage.

The greatest advantage of ML is its ability to solve modern data analysis challenges. To begin with, the algorithms only learn from the data that is provided to them. The more sophisticated the data, the better algorithms can learn to make future accurate predictions. In the case of the self-driving car, for example, the data should include as many examples from real-life scenarios as possible. It is a difficult task, but it is critical as it determines the functionality of the algorithm. Another significant advantage of ML is its ability to generalize its behavior to unknown scenarios based on learning data. That is what makes them so useful in a wide range of applications, as they can predict the desired effect based on partially known or completely unknown premises. Furthermore, some of the ML algorithms described later in this chapter are not affected by missing or sparse data. This makes them even more useful in many modern technological applications where such situations are unavoidable. Finally, once trained, the ML algorithms are fast and can

process multiple data instances at the same time, making them an excellent choice for low-latency systems such as real-time analysis pipelines in astronomical observations.

Though ML has been developed for several decades, its significant rise in popularity began in the twenty-first century as a result of the remarkable development of computational power, particularly Graphical Processing Units (GPU) [99]. This allowed for a significant reduction in the training time of ML algorithms as well as an increase in the processing capabilities of large amounts of data.

1.2.1 Machine learning paradigms

Numerous ML algorithms available today can be classified into three ML paradigms, which differ in how data is used during algorithm training.

Supervised learning

Algorithms in supervised learning [100] learn from labeled data. Every instance in the studied data has a corresponding label, which could be a name (e.g., car brand, animal name, galaxy type, etc.) or a number (value of temperature, speed, energy, etc.). In the first case, ML algorithms are designed to solve classification problems, i.e. assign a specific class label based on a set of parameters (metadata, pixels on an image). In the latter case, the ML algorithms are designed to solve regression tasks, i.e. assign a specific value to the data instance based on a set of parameters.

Unsupervised learning

The algorithms in unsupervised learning [101] learn from unlabeled and unstructured data. They are used for tasks such as clustering and data reduction. Clustering allows to find patterns and structures in data and use that information to create similarity clusters, i.e. gather data instances with similar features into clusters. Unsupervised learning is similar to how our brains learn [102]. Consider, for example, a basket of fruits that we have never seen before. We will look for similarities between the fruits based on some patterns because we do not have prior knowledge about them (e.g. labels). The patterns could be related to color, smell, or texture.

In the case of data reduction, unsupervised learning allows for the reduction of data dimensionality while preserving key features that can later be used to reconstruct the initial data with minimal information loss.

Reinforcement learning

The algorithms face a game-like situation in the case of reinforcement learning [103]. The predefined agent learns how to achieve a specific goal by interacting with the environment using well-defined rules. The agent receives either rewards or penalties for the actions it takes. It aims to achieve the maximum pay. From completely random trials to advanced tactics and superhuman skills, it is up to the model to figure out how the task is carried out to maximize the award. Reinforcement learning is a type of ML in which the algorithm acquires knowledge by interacting with the environment rather than learning from data, as in previous paradigms.

1.2.2 Machine learning algorithms

Among the numerous ML methods, one class of algorithms is the most widely used. They are referred to as Artificial Neural Networks. I chose them as the main technique of the research presented in this dissertation due to their versatile capabilities. In the next chapter, I will cover an in-depth characterization of the ANN. However, first I will present alternative ML algorithms capable of application in a variety of problems.

One of the first and best-known ML algorithms is linear regression [104]. As the name implies it is used to solve regression problems, i.e. to predict the value of a variable based on the value of another variable. Fitting the best line establishes the relationships between the dependent value (the prediction outcome) and the independent variables. The main limitation of the linear regression is its linear character. Many real-world problems are non-linear (for example, image classification) and require more sophisticated methods.

Among other popular linear methods, however, used for the classification tasks, is the logistic regression [105]. By fitting data to a `logit` function and producing output values ranging from 0 to 1, the algorithm predicts the likelihood of an event occurring. Despite its name, logistic regression is considered a linear model because the decision boundary it generates is linear. The algorithm is limited to two-class (binary) classification problems, with multi-class extensions being possible. This, however, requires the transformation of the problem into multiple binary classification cases.

More powerful than the above-mentioned methods but still linear are support vector machines [106]. They are mostly used for classification, but with some modifications, they can also be used to solve regression problems (though in that case their name is changed to support vector regression [107]). The support vector methods use a linear hyper-plane to map input variables to points in a higher-dimensional space to maximize the separation between classes (classification) or continuous quantity (regression). They are powerful in many applications, but they struggle with large datasets. Furthermore, in the presence of noise in the input data, the resulting separation may contain large errors with overlapping data instances.

For the non-linearly separable data, a popular algorithm is a decision tree [108]. It is a supervised method that can be used for both classification and regression problems. The algorithm divides the data into two or more homogeneous sets based on the most significant attributes or independent variables. The resulting sets are as dissimilar as possible to one another. Each of the sets can be further subdivided, resulting in a hierarchical structure that resembles a reversed tree. The random forest [109] method is an extension to the method. It is made up of a collection of decision trees. Random forests are a powerful modeling technique that is far more resilient than a single decision tree. They aggregate many decision trees to reduce overfitting and error due to bias (a phenomenon in which some elements of a dataset are more heavily weighted and/or represented than others) and thus produce useful results.

Another approach utilizing an ensemble of ML methods is called boosting [110]. The boosting algorithm's basic working principle is to generate multiple weak learners and combine their predictions to form one strong learner. The distinction between strong and weak learners is linked to a degree of correlation with the true classification. The correlation is low for the weak learner (it is only slightly better in labelling examples

than random guessing). The strong learner, on the other hand, is well-correlated with the true classification and, as a result, can label samples much better than random guessing. By sequentially training weak learners to correct their predecessors, a strong learner is formed. Popular boosting algorithms include AdaBoost [111] and XGBoost [112].

All of the algorithms mentioned above are used in the domain of supervised algorithms. However, there are a plethora of unsupervised ML techniques associated with data clustering. The K-means algorithm [113] is one of the most popular. The algorithm tries to group similar kinds of items in form of clusters centered around k centroids, defined in some parameter space. As a criterion, K-means uses a similarity (or distance) measure. In the simplest case, the Euclidean distance is used to assign each data point to the closest cluster while keeping the centroids as small as possible.

An alternative method to clustering is based on a Gaussian Mixture Model (GMM) [114]. GMMs assume that there are a finite number of Gaussian distributions with unknown parameters, each of which represents a cluster. As a result, the GMM tends to group data points from a single distribution together regardless of cluster sizes and shapes. GMM has a significant advantage over K-means because of its probabilistic nature; for example, GMM can compute the probability that a given point came from each of the different fitted components. Furthermore, while K-means imposes a hard break between clusters, GMM can return overlapping clusters.

Clustering in the unsupervised domain can also be accomplished indirectly through data dimensionality reduction. These algorithms reduce data dimensions while preserving key characteristics of similarity between samples. Manual clustering can then be performed on such reduced dimensional representations that are visualized in 2- or 3-D (D for dimensional). Manifold learning algorithms such as t-distributed stochastic neighbor embedding (t-SNE) [115] and uniform manifold approximation and projection (UMAP) [116] are popular examples of dimensionality reduction techniques in the context of clustering. According to the manifold hypothesis, [117], real-world high-dimensional data lies on low-dimensional manifolds embedded within the high-dimensional space. UMAP and t-SNE assume that the dimensionality of the data is artificially high and attempt to reduce it to a lower dimension (usually 2-D). The main distinction between those algorithms is that t-SNE is deterministic (it uses deterministic methods for initialization such as principal component analysis [118]), whereas UMAP retains its stochastic character due to optimization of its cost function using stochastic gradient methods (more on the cost functions and optimization in the latter part of the chapter, see Sec. 1.2.3).

Many other interesting ML algorithms exist. Many more are still being researched and developed as the field is constantly expanding. For a more detailed overview of methods see [119].

1.2.3 Artificial neural networks

Artificial Neural Network (ANN) is the algorithm that revolutionized the field of ML at the beginning of the twenty-first century. ANNs are loosely based on the biological neural networks that constitute animal brains. The ANN is built on a network of connected units known as artificial neurons (AN), which mimic the neurons in the human brain. The first AN model has been proposed in 1957 under the name *perceptron* [120] (see Figure 1.6).

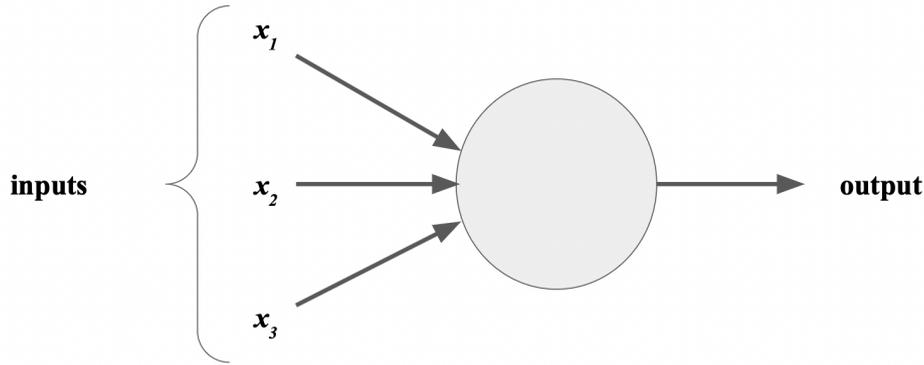


Figure 1.6: The perceptron was the first artificial neuron model proposed in 1957 [120]. In the centrally located node, a sum of weighted inputs is computed; if the result of the sum exceeds a predefined threshold, the output equals one, and the signal is processed further. In the other case, no information is passed on - the output is zero.

The perceptron is made up of *edges*, which are connections between the input variables and the centrally located *node*. Every edge is associated with a weight w_j , which adjusts the strength of the processed signal in that connection. The node computes the sum of weighted inputs. The perceptron output is binary, 0 or 1, depending on whether the weighted sum is less than or greater than some threshold value, as shown in Eq. 1.12. Weights and thresholds in the perceptron could be changed during training. To simplify the perceptron notation, the scalar product is used to represent the weighted sum, and the threshold is replaced by bias b , where $threshold \equiv -b$ as presented in Eq. 1.13.

$$\text{output} = \begin{cases} 0, & \text{if } \sum_j w_j x_j \leq \text{threshold} \\ 1, & \text{if } \sum_j w_j x_j > \text{threshold} \end{cases} \quad (1.12)$$

$$\text{output} = \begin{cases} 0, & \text{if } \mathbf{w} \cdot \mathbf{x} + \mathbf{b} \leq 0 \\ 1, & \text{if } \mathbf{w} \cdot \mathbf{x} + \mathbf{b} > 0. \end{cases} \quad (1.13)$$

In its original form, the perceptron has two major limitations. The first constraint is related to the binary output. The network's capabilities for processing complex data, such as image classification using pixels with values ranging from 0-255 or predicting stock prices in regression problems, are limited. The second limitation is the sensitivity of the perceptron output to changes in weights and biases. A small change in weight value, for example, can cause the perceptron's output to spin from 0 to 1. A single spin of a perceptron in the network could then cause a significant change in some complicated way.

Modern perceptrons use the *activation function* $f(\mathbf{w} \cdot \mathbf{x} + \mathbf{b})$ in the node to overcome these constraints, causing the neuron's response to become optionally non-linear. As a result, even minor changes in weight (or bias) result in only minor changes in output. The activation function gives us more control over how the ANN operates.

Among the most popular activation functions used in variety of ANN are (in the following equations z replaces $\mathbf{w} \cdot \mathbf{x} + \mathbf{b}$ for simplicity) [121]:

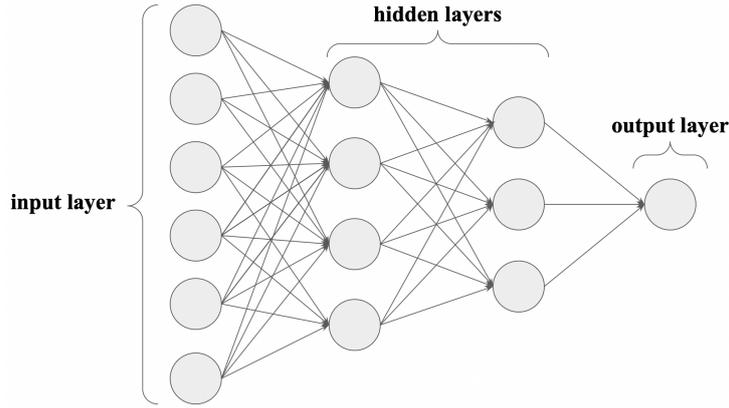


Figure 1.7: Schematic representation of ANN consisting of an input layer, two hidden layers, and an output layer. The network is made up of many perceptrons that have been extended with the activation function and are linked together. Each neuron in one layer is linked to all the neurons in the next layer, resulting in a complex network. However, depending on the architecture, ANN can also skip connections.

- binary step (see Eq. 1.13),
- linear: $f(z) = z$,
- sigmoid (logistic): $f(z) = \frac{1}{1+e^{-z}}$,
- hyperbolic tangent: $f(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$,
- rectified linear unit (ReLU): $f(z) = \max\{0, z\}$.

As previously stated, a perceptron is a fundamental cell upon which complex ANN architectures are built. Later in this chapter, three of the most popular network architectures will be discussed (Sec. 1.2.3). Despite their differences, some fundamental similarities can be identified. To begin, each neural network has three types of layers, as shown in Fig. 1.7: input, hidden, and output. The input layer is related to the data size and dimensionality used for algorithm training. The algorithm's size is fixed, which means that all data studied in the given problem must have the same dimension.

The type of problem to be solved determines the output layer. In the case of classification tasks, each neuron in that layer is associated with a distinct class or category. Every neuron in a regression problem corresponds to the predicted value (stock prices, temperature, etc.).

The hidden layer, or layers as they are more commonly known, refers to everything between the input and the output. This component of ANN could be made up of thousands of neurons linked together in a sophisticated manner. The deeper the algorithm is, the more abstract features processes. This is especially useful when dealing with complex tasks like classification based on images described by thousands of pixels.

Another point of similarity among various ANN architectures is the direction of information propagation through the network. The majority of networks use *feed-forward propagation*, in which information is transferred from the input layer to the output layer. However, as described in the latter part of this chapter (see Sec. 1.2.3), some networks can also propagate information backward, which is known as *feed-backward propagation*.

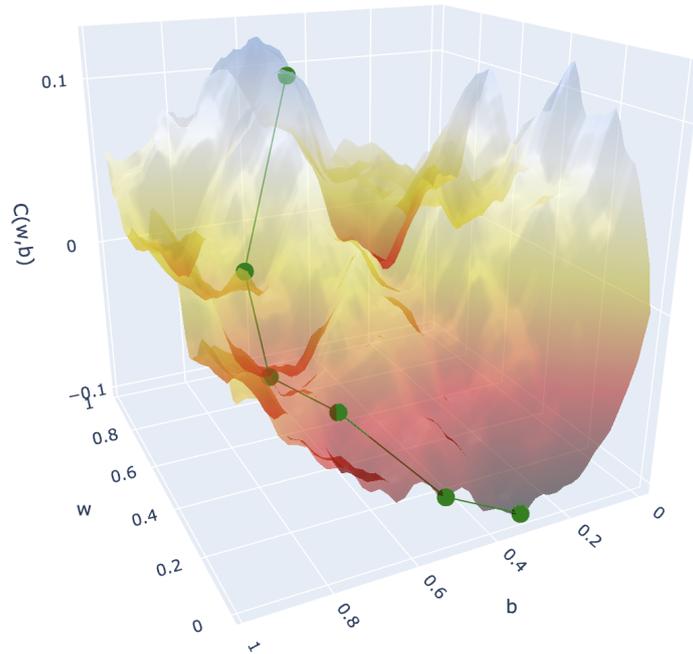


Figure 1.8: Visualization of gradient descent in the space defined by ANN’ weights w and biases b . The cost function $C(w, b)$ is presented on the z -axis. The green dots represent the cost function values calculated at each stage of the gradient descent computation (training epoch), starting with the maximum and ending with the minimum.

How they learn?

Every ANN, as described in the preceding subsection, has an output layer that refers to the studied problem. This feature allows us to determine how well ANN reconstructs the desired output $y(x)$ for all training inputs x . The function responsible for such a measure is known as the *cost function* [122] (also known as the *loss function*) $C(w, b)$, and it is a function of weights w and biases b . The most commonly used cost functions are:

- mean squared error (MSE),
- root mean squared error (RMSE),
- binary cross-entropy (for binary classifiers),
- categorical cross-entropy (for multi-label classifiers - multi refers to more than 2 classes).

The goal of ANN training is to minimize the cost function, that is finding a set of weights and biases that minimizes $C(w, b)$. Because cost functions are differentiable, the simplest way to find the minimum is to compute derivatives analytically and look for extrema. However, because modern ANN has millions of parameters, such an approach is suboptimal.

An algorithm called *gradient descent* is an alternative to the analytic approach [123]. This iterative algorithm finds the minimum of a function by moving in the direction of the steepest descent, which is defined by the gradient’s negative. The gradient descent is well described by the analogy to a steep valley (see Fig. 1.8 for a 3-D example). The goal is to find a way to the valley’s bottom. For the sake of simplicity, I will assume

that the cost function depends on two variables: w_1 and b_1 . When we make a move by a small amount Δw_1 in w_1 direction and Δb_1 in b_1 direction, the $C(w_1, b_1)$ changes as follows:

$$\Delta C \approx \frac{\partial C}{\partial w_1} \Delta w_1 + \frac{\partial C}{\partial b_1} \Delta b_1. \quad (1.14)$$

To move down the valley ΔC have to be negative. To meet this condition, we need to find a way of correctly choosing Δw_1 and Δb_1 . In order to do so, Eq. 1.14 can be rephrased using gradient of the cost function $\nabla C \equiv (\frac{\partial C}{\partial w_1}, \frac{\partial C}{\partial b_1})^T$ and the vector of changes $\Delta v \equiv (\Delta w_1, \Delta b_1)^T$. Then, the Eq. 1.14 transforms into:

$$\Delta C \approx \nabla C \cdot \Delta v. \quad (1.15)$$

Finally, the condition for the negative ΔC can be met by relating the vector of changes Δv to the gradient ∇C , where η is a small, positive parameter called the *learning rate*. The scale of changes in the directions of w and b is determined by η . If the learning rate is too high, it is possible that the minimum of the cost function will not be found. When the learning rate is set too low, the convergence to the minimum point may take too long to be useful in practical applications.

$$\Delta v = -\eta \nabla C. \quad (1.16)$$

This yields the following equation, which meets the initial criterion, allowing the motion down the slope of the valley to find minimum since $\|\nabla C\|^2 \geq 0$:

$$\Delta C \approx -\eta \|\nabla C\|^2. \quad (1.17)$$

With a generic description of the gradient descent, we can now return to the initial conditions, where vectors w and b are collections of variables. According to Eq. 1.16, we can update weights and biases of ANN using a gradient of the cost function and learning rate as follows:

$$\begin{aligned} w_k &\longrightarrow w'_k = w_k - \eta \frac{\partial C}{\partial w_k}, \\ b_l &\longrightarrow b'_l = b_l - \eta \frac{\partial C}{\partial b_l}. \end{aligned} \quad (1.18)$$

The procedure is repeated until the ANN has used up all of the training inputs, which is referred to as completing an *epoch* of training. At that point, the ANN begins a new training epoch and repeats the weight and bias updates.

The main limitation of the presented gradient descent approach is that it must be computed for each training input data instance and then averaged over the entire data size. From a practical point of view, this step can be time-consuming, resulting in the ANN learning slowly.

To overcome this problem, an improvement of the algorithm has been introduced. Instead of computing the gradients for the whole dataset, ∇C is computed for small batches of randomly selected data instances. Averaging over this small sample results in a good estimate of the true gradient of the cost function while substantially speeding up the learning. This modification to the algorithm is called *stochastic gradient descent* (SGD) [123].

Modern ANN use even more advanced forms [124] of gradient descent than the SGD. The *Adaptive Gradient* (Adagrad) [125], for example, adapts to the learning rate of the ANN parameters. It can converge faster to a global minimum of the cost function by using low learning rates for parameters associated with the most frequently occurring features and high learning rates for parameters associated with rarely occurring features. The main problem with Adagrad is that the learning rate may decrease so much during ANN training that the algorithm does not converge to the global minimum - the updates to weights will be too small to affect the performance of the ANN.

Another popular optimization algorithm is *Root Mean Squared Prop* (RMSProp) [126]. The RMSProp uses a moving average of the squared gradient to solve the problem of decreasing the gradient of Adagrad.

The *Adaptive Momentum Estimation* (Adam) [127] is an extension of RMSProp that takes advantage of the momentum algorithm: a method for accelerating the SGD in the "correct" direction on the slope of the cost function "valley" and dampening the oscillations on its path. In practice, the Adam algorithm tends to converge to the global minimum faster than the other optimizers while avoiding local minima.

During ANN training, the gradient descent algorithm does not result in automatic updates of weights and biases. The algorithm responsible for this is known as *backward propagation of errors*, abbreviated as the *backpropagation* [128]. The algorithm, first proposed in 1960, uses a method known as a *chain rule* - a formula for computing the derivative of a composite function - to effectively train ANN by fine-tuning the weights and biases of a neural net based on the error rate (cost function) obtained in the previous epoch.

The cost function between the desired and actual outputs is computed after the input data has been propagated through the network. The backpropagation algorithm then performs a backward pass while adjusting the ANN's weights and biases - computation is performed from the last layer to the first layer using gradient descent. Because the neural net parameters are interconnected, the chain rule can be used to generate a dependency sequence. The neurons in the k -th layer are determined by the weights and biases between the k -th and $k - 1$ -th layers, as well as the outputs of activation functions $f(\mathbf{w} \cdot \mathbf{x} + \mathbf{b})$ from the $k - 1$ -th layer. By computing partial derivatives of the cost function with respect to all dependent variables via the chain rule, it is possible to obtain values required to update weights and biases in Eq. 1.18.

The backpropagation is repeated until the desired cost function value is obtained. In practice, this means that the decreasing $C(w, b)$ settles around some value and does not change over the next training epochs.

Convolutional neural networks

In the last 20 years, a new area of ML has emerged under the umbrella of deep learning (DL) [129]. DL stands out due to its algorithms' ability to process raw data without the

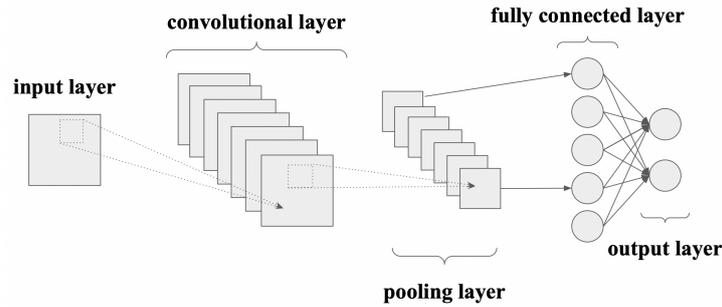


Figure 1.9: Schematic representation of CNN consisting of an input layer, convolutional layer, pooling layer, fully connected layer, and output layer. The fully connected layer, as shown in Fig. 1.7, corresponds to the *classic* ANN as all of the neurons in this layer are connected to all of the neurons in the subsequent layers. Multiple instances of data from previous layers are obtained by applying different convolutional kernels to the convolutional layer. The pooling layer, on the other hand, has the same number of instances as the convolutional layer but is smaller due to the pooling operation.

need for manual feature engineering. DL was sparked by the popularity of a specific ANN architecture known as the Convolutional Neural Networks (CNNs) [130].

The CNN is a deep, feed-forward ANN whose structure was inspired by studies of the visual cortex in mammals, the part of the brain that specializes in visual information processing. As a result, CNN was initially used to process images in the context of classification tasks. CNNs such as VGG-19 [131], Inception [132], or ResNet [133] are currently the most complex and deep neural networks e.g. the first version of Inception ANN included 27 convolutional layers.

Figure 1.9 shows an example of CNN architecture. CNN’s central feature is known as a convolutional layer. It detects local feature conjunctions in the input data and maps their appearances to a feature map. As a result, the input data is divided into sections, resulting in the formation of local receptive fields and the compression of feature maps. The size of the receptive field (also known as the *kernel*) corresponds to the scale of the data details to be examined. The smaller the kernel, the finer the patterns that CNN can detect.

The convolutional layer consists of a set of filters. Each filter is in charge of extracting the specific type of feature maps. The filters from shallow hidden layers may contain human-understandable features, such as edges, shadows, and gradients present in the input images. The deeper the layer and its associated filters, however, the more abstract features it extracts from preceding layers.

Another distinguishing layer of the CNN architecture is the pooling layer [134]. The feature maps’ dimensions are reduced by pooling layers. As a result, they reduce the number of parameters to learn and the amount of computation performed in the network, while also making it more resilient to noise and translations. The maximum pooling and average pooling layers are the most common types of pooling layers. The maximum pooling operation computes the largest values in each patch of each feature map. In the case of average pooling, the average value for the patch’s features is computed. In this context, the pooling layers can be thought of as a compression tool.

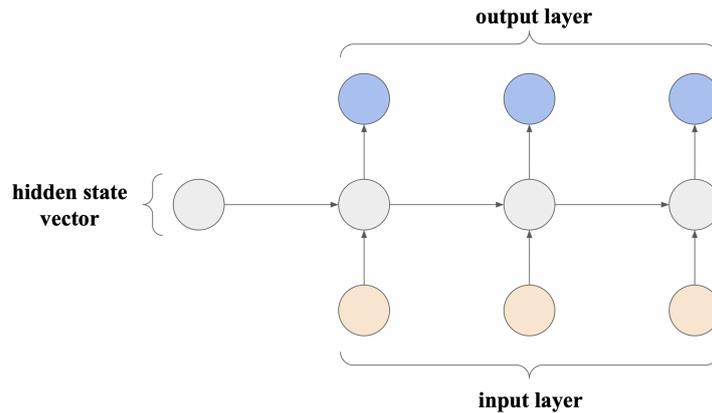


Figure 1.10: Schematic representation of RNN consisting of an input layer, an output layer, and *hidden state vector*. The network’s hidden state vector serves as a memory. It stores information about previous elements of the input sequence that the RNN is already aware of.

Because the weights in the convolutional and pooling layers are shared, CNNs are faster than traditional ANN. Sharing weights reduces the number of neurons, allowing for faster training while maintaining the model’s generalization capabilities.

Modern applications of CNN are not limited to the images as illustrated in this thesis. CNNs are now also used in e.g. the studies of time-series data which are 1-D vectors of data, as demonstrated in this thesis in the applications on the GW strain data.

Recurrent neural networks

Recurrent neural networks (RNNs) [130] are the next most popular ANN architecture, and they are particularly well suited for applications in sequence analysis. In contrast to traditional ANN or CNN, RNN allows for the processing of input data of any size. They view the data as a sequence of events and capture meaningful relationships between inputs. These relationships can then be used to extract information about the data’s evolution over time, which is especially useful for time-series data (e.g. speech recognition, music generation).

Figure 1.10 shows an example of an RNN architecture. Specifically, it includes the *hidden state vector*, which represents the context based on previous inputs and outputs. As a result, depending on previous inputs in the series, the same input applied to the RNN may produce different results. The same input would always result in the same outputs in the case of non-recurrent neural networks because it is only influenced by the weights applied to that input.

The ability to share weights across time is both an advantage and a disadvantage of RNN. On the one hand, it enables taking the time evolution of the input data into account. However, it significantly increases the training time. Furthermore, access to information between distant states is restricted. Because weights (and neurons) in RNN are shared between states during gradient descent computation, backpropagation must update all neurons far back in time. The gradient decreases as the number of operations performed by backpropagation increases. Some changes to the cost function may be so minor that they do not affect training. This problem has been defined under the term *vanishing*

gradient.

The Long Short-Term Memory (LSTM) network [135] has been proposed as a solution to the vanishing gradient problem. LSTM is a variant of RNN with recurrent weights equal to unity. As a result, the cost function gradient's exponential decaying (or explosion for very large weights) is absent. Nevertheless, the LSTM still has some limitations when it comes to accessing information between distant states, but this does not preclude successful implementations in time-series analysis.

Autoencoders

While the CNNs and RNNs were initially designed for supervised tasks, the autoencoder (AE) [136] is a type of ANN architecture that is typically used in unsupervised problems. AE is made up of two parts: the *encoder* and the *decoder*. The encoder is in charge of encoding (compressing) the input data into a space known as *latent space*, which is typically smaller than the number of input features; sometimes this part is referred to as the architecture bottle-neck. The decoder, on the other hand, is supposed to decode (decompress) the data to its original representation. To put it another way, the AE forces a compressed knowledge representation of the original input data. While doing so, the AE learns key properties of the data by forcing it through the latent representation, allowing it to reconstruct the input. The latent representation containing these key features could later be used to cluster samples based on the underlying patterns in the data.

The ability to combine AE with other ANN architectures is a significant advantage. AE, for example, can be built on top of either CNN or RNN to process images or sequential data. Furthermore, instead of a single point, the encoding part of AE can return distribution over a latent space. The architecture based on this concept is known as a variational autoencoder (VAE) [137]. The model assumes that the input data has an underlying probability distribution (e.g. a Gaussian distribution) and then attempts to find its parameters. The latent distribution could later be used to generate artificial samples that resemble the original data, such as fake images. The ability of VAEs to generate artificial data has been enhanced further by combining the model with additional ANN aimed at distinguishing fake samples from original. The overall architecture that can provide such utility is known as Generative Adversarial Network (GAN) [138].

Chapter 2 contains more information about AE training and its characteristics.

Advantages and disadvantages of artificial neural networks

As mentioned in the preceding section, ANNs are extremely versatile algorithms. They can be used in supervised paradigm frameworks i.e. in classification and regression tasks, as well as in unsupervised tasks, i.e. in clustering and dimensionality reduction. ANNs can process a wide range of data representations (images, time-series, etc.). However, as with any method, while ANNs have significant advantages, they also have drawbacks that should be carefully considered before applying them to a specific problem.

The greatest advantage of the ANNs is their application flexibility and versatility. Because neural networks are a mathematical model with an approximation function, they can be applied to any data that can be represented numerically. With the addition of an activation function, ANN can be transformed into a highly nonlinear model that can be applied to a wide range of nonlinear data, such as images. Because the only limitation

is the computing capability of the hardware used, ANNs can be extended to any number of layers and input features. Since ANN has a layered structure, even very large inputs with many features can be processed successfully by dividing the considered problem (e.g., regression) into a layered network of simpler elements processing only parts of the input (e.g. CNN). Only the training of an ANN may take a significant amount of time, as once trained, they are very efficient at returning predictions. Speed of computation is a crucial advantage in the application to the low-latency systems in data processing pipelines used in scientific experiments.

The main drawback of ANN is its *black box* nature [139]. ANNs can approximate any function due to their multi-layered nonlinear structure. The structure of the function being approximated, on the other hand, is unknown. Even by studying the ANN architecture, one cannot always decipher how much influence each independent variable of the model has on the dependent model. The performance of the ANN heavily depends on the statistical representability of the training data concerning the studied problem. As ANN has a non-transparent character, they can extract from training data biased information without our knowledge.

Another significant drawback of ANNs is the absence of reliable uncertainty estimates for their output [140–142]. Furthermore, ANNs are prone to overconfidence in their predictions [143, 144]. The field of ANN uncertainty is undergoing intensive research, and several solutions to the problem have been proposed [141]. Solutions, on the other hand, are frequently problem-specific and should be thoroughly considered before being applied to the studied case.

Machine learning in the context of this thesis

As previously stated, the data collected by GW detectors are difficult to analyze. Non-linear character of the data, presence of both short- and long-lasting detector artifacts, as well as complex morphology of astrophysical signals buried deep within the noise, are just a few of the challenges that data analysis methods must overcome. These difficulties, however, are not a major issue for ANNs. Indeed, because of their non-linear nature, ability to detect complex patterns, and resistance to noise, ANNs appear ideal for use in the analysis of GW detector data. Because ANNs are extremely efficient at processing large amounts of data, they appear to be an ideal candidate for future low-latency pipelines combining data from various astronomical experiments all over the world in the domain of multi-messenger astronomy.

Furthermore, different ANN architectures enable the processing of data from GW detectors using different representations. This allows us to explicitly focus on different features of potential astrophysical signals buried within the noise, such as the time evolution of signals in time series or the time-frequency relationship in spectrogram images.

In the context of supervised learning, ANN can be used to classify various GW types and/or glitches, as well as to predict astrophysical parameters of the emission source. In unsupervised learning, ANN can be used for data denoising (removing excessive noise around the signal), dimensionality reduction, and clustering. Many other intriguing applications of ANN (and ML in general) in the study of GWs and related topics have emerged in recent years. I have listed a few of my favorites below.

The classification and detection of different GW types is the main field of interest in the

application of ML methods in the GW detectors data. As a result, the majority of research has concentrated on this subject. One of the first works in the field is a great example: George et al. [145] developed the *Deep Filtering* method for the signal processing, based on a system of two deep CNNs, designed to both detect as well as estimate parameters of compact binary coalescence signal in the very noisy time-series data stream. A similar approach was presented by Gabbard et al. [146] where authors reached similar sensitivity as MF using only CNN. Another interesting work was done by Gebhard et al. [147] where authors presented limitations of CNN towards adversarial examples of data.

As for the continuous GWs emitted by NS, a few interesting works have been also published. Dreissigacker et al. [148, 149] have been using deep neural networks as a search method in both simulated as well as real detector data. Miller et al. [150] presented a comprehensive study of the effectiveness of CNNs to detect long-duration transient GW signals from isolated NSs covering the limitations of the method on the search of that particular GW type. Bayley et al. [151] developed a robust ML pipeline allowing detection of not only continuous GWs but also stationary lines in the detector's data.

The GW emitted by collapsing massive stars have also been studied with ML. Detection capabilities of CNN on the supernova GW signal have been presented by Chen et al. [152]. Astone et al. [153] show that CNN can outperform currently operating data analysis algorithms devoted to selecting GW transient signals from CCSN.

Another area of intense research is the application of ML methods to the estimation of GW source parameters. A fascinating example of such research is the work of Gabbard et al. [154], in which the authors develop a conditional VAE for Bayesian-like parameter estimation for CBC sources and compare the results to the currently used Bayesian inference methods. The parameters of the same source type were studied by Green et al. [155] with auto-regressive neural network flows achieving spectacular results.

Another rapidly expanding area of ML research in GW astronomy is detector characterization and data quality. Many works focus on the classification of noise transients (glitches), based on both simulated and real data using CNN, such as Razzano and Cuoco [156], Powell et al. [157] and George et al. [158]. Worth mentioning is also usage of citizen projects in this context such as the Gravity Spy by Bahaadini et al. [159], useful because of robust preparation of the training data by human inspection. Aside from glitch classification, data denoising is another important area of research. The goal here is to remove excessive noise from data to extract the underlying GW signal, which can then be used for parameter estimation or sky localization. Shen et al. [160] applied AEs on the denoising of CBC GWs. Denoising of noise transients on the example of blip glitch with dictionary learning has been conducted by Torres-Forné et al. [161]. Finally, ML is used also in the GW signal generation e.g. GAN has been used to generate burst GWs by McGinn et al. [162].

Every year, the number of ML applications in GW astronomy grows, making it impossible to list them all. For a recent review of the state-of-art, Cuoco et al. [163] provide a broad overview of such methods.

The application of ML methods and ANN in particular in the GW detectors data is a relatively new field of research in astrophysics. However, as various groups of researchers around the world have shown, it has the potential to significantly improve currently existing pipelines of data analysis as well as help in the precise estimation of astrophysical

parameters of GW emission. Before that happens, however, we need a thorough understanding of the possibilities and limits of ML concerning data collected by GW detectors. In this context, my research is an essential part of a wide-ranging effort to understand how ML can improve our methods of learning about the world.

1.3 Motivation and thesis overview

The thesis was motivated by the desire to investigate ML as a novel method of analyzing data collected by GW detectors. In practice, the investigation focused on *how* ML can be used to analyze difficult and notoriously noisy GW detector data, as well as *what* scientific outcomes such applications provide for GW astronomy.

To achieve the aforementioned objectives, I researched various aspects of GW astronomy. In the first project, I investigated the already detected class of GW signals - theoretically well-understood waveforms emitted by the CBC systems - from a novel perspective, employing ML as an anomaly detection method (paper I). In subsequent projects, I looked for GWs in previously undetected systems such as CCSN explosions (paper II) and oscillating (or deformed) rotating isolated NSs (paper III). Both of these GW sources prove to be much more difficult to detect in comparison to CBC signals because they are either not well-understood theoretically and short in duration (the burst-like CCSN signals), or well-understood but with much smaller GW amplitude (CW signals emitted by isolated NSs). The fourth project investigated the use of ML to reconstruct physical parameters of NSs from a GW-EM multi-messenger perspective (paper IV). The thesis covers therefore main types of GW sources that are currently actively pursued by the astrophysical community. From the data analysis point of view, the algorithm that in my opinion has the most potential of all ML methods - ANN - served as a link between all projects.

A summary of each publication concerning the material described in previous sections is provided below.

Paper I: Anomaly Detection in Gravitational Waves data using Convolutional AutoEncoders ([Filip Morawski et al. 2021 Mach. Learn.: Sci. Technol. 2 045014](#))

In this paper, I looked at how to use a convolutional AE to search for anomalies in GW data. The anomalies were defined in terms of transient signals distinct from the IFO's "usual" non-stationary noise. I concentrated on the anomalies represented by GWs emitted by binary BH system mergers. In the reconstruction of injected signals, I demonstrated the capabilities of ML implementation. In addition, I investigated the impact of GW strength, as defined by matched filter SNR, on anomaly detection. Furthermore, I considered the use of the method for the temporal localization of anomalies in the studied time-series data. Finally, I validated the results of anomaly searches on real data containing confirmed GW detections, demonstrating my method's generalization ability in detecting GWs unknown to my ML models during training.

Paper II: Core-Collapse Supernova Gravitational-Wave Search and DeepLearning Classification ([Alberto Iess et al. 2020 Mach. Learn.: Sci. Technol. 1 025014](#))

In this paper, with co-authors, I developed a new method based on ML to search for the GWs emitted during CCSN explosions. Our study relied on five cutting-edge CCSN models derived from the most recent hydrodynamic simulations of neutrino-driven core-collapse processes. The CNN algorithm was applied to two types of data representations

in the paper: time series and spectrograms. Our findings demonstrated that spectrograms are significantly better representations than time series. Aside from the stochastic nature of the CCSN GW in the time domain, the evolution of signal frequencies contains important features that help distinguish the GWs from noise. In addition, we demonstrated the results of future CCSN GW searches using the planned ET IFO. The results show that when the ET is used, the robustness of CCSN search using ML improves significantly when compared to currently operating interferometers (on the example of the Advanced Virgo), both in terms of detection efficiency and source distance. Finally, we demonstrated that CNNs were particularly effective at distinguishing CCSN GWs from glitches.

Paper III: Convolutional neural network classifier for the output of the time-domain \mathcal{F} -statistic all-sky search for continuous gravitational waves ([Filip Morawski et al. 2020 Mach. Learn.: Sci. Technol. 1 025016](#))

In this paper, I investigated the use of a CNN in the classification of GW signal candidates related to the continuous signal emission by deformed rotating NS. The proposed approach was novel in that it used CNN as a follow-up (post-processing) method to the \mathcal{F} -statistic, increasing the significance of possible astrophysical signal detection by studying the distribution of the \mathcal{F} -statistic rather than individual high-SNR candidates. I demonstrated a significant reduction in signal candidate processing time while maintaining the method's high accuracy. The CNN detected GWs based on a distribution from a single time segment across a wide range of GW frequencies and SNRs. Furthermore, the CNN's were able to differentiate GWs from the detectors' persistent spectral artifacts, known as stationary lines.

Paper IV: Neural network reconstruction of the dense matter equation of state derived from the parameters of neutron stars ([Filip Morawski & Michał Bejger 2020 A&A 642 A78](#))

In this paper, I investigated the use of AEs in the reconstruction of the dense matter equation of state (EOS) based on both EM (masses, radii) and GW (component masses, tidal deformability during the system's inspiral) observations. The ANN presented in the paper was flexible enough to generalize the mapping of the microscopic (EOS) parameters and global measurable parameters. Another notable finding was that a realistic NS mass function distribution based on known observations was insufficient for EOS reconstruction in the high-density regime. A large number of observations of massive stars (around $2 M_{\odot}$) would be required to overcome this problem. In addition, I demonstrated how ANN could be used to infer the radius of the NS from GW observations, which is a non-trivial relation to deduce.

Part II

To Expect The Unexpected
(Searches for irregularities in the
data)

Chapter 2

Paper I: Anomaly detection in gravitational waves data using convolutional autoencoders



PAPER

OPEN ACCESS

RECEIVED

28 November 2020

REVISED

1 March 2021

ACCEPTED FOR PUBLICATION

31 March 2021

PUBLISHED

19 July 2021

Original Content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](#).

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.



Anomaly detection in gravitational waves data using convolutional autoencoders

Filip Morawski^{1,*} , Michał Bejger¹ , Elena Cuoco^{2,3,4}  and Luigia Petre⁵ ¹ Nicolaus Copernicus Astronomical Center, Polish Academy of Sciences, Bartycka 18, 00-716 Warsaw, Poland² European Gravitational Observatory (EGO), I-56021 Cascina, Pisa, Italy³ Scuola Normale Superiore, Piazza dei Cavalieri 7, I-56126 Pisa, Italy⁴ INFN, Sezione di Pisa, Largo Bruno Pontecorvo, 3, I-56127 Pisa, Italy⁵ Department of Computer Science, Faculty of Science and Engineering, Åbo Akademi University, Tuomiokirkontori 3, 20500 Turku, Finland

* Author to whom any correspondence should be addressed.

E-mail: fmorawski@camk.edu.pl**Keywords:** gravitational waves, machine learning, autoencoders, convolutional neural networks

Abstract

As of this moment, 50 gravitational wave (GW) detections have been announced, thanks to the observational efforts of the LIGO-Virgo collaboration, working with the Advanced LIGO and the Advanced Virgo interferometers. The detection of signals is complicated by the noise-dominated nature of the data. Conventional approaches in GW detection procedures require either precise knowledge of the GW waveform in the context of matched filtering searches or coincident analysis of data from multiple detectors. Furthermore, the analysis is prone to contamination by instrumental or environmental artifacts called glitches which either mimic astrophysical signals or reduce the overall quality of data. In this paper, we propose an alternative generic method of studying GW data based on detecting anomalies. The anomalies we study are transient signals, different from the slow non-stationary noise of the detector. The anomalies presented in the manuscript are mostly based on the GW emitted by the mergers of binary black hole systems. However, the presented study of anomalies is not limited only to GW alone, but also includes glitches occurring in the real LIGO/Virgo dataset available at the Gravitational Waves Open Science Center. To search for anomalies we employ deep learning algorithms, namely convolutional autoencoders, which are trained on both simulated and real detector data. We demonstrate the capabilities of our deep learning implementation in the reconstruction of injected signals. We study the influence of the GW strength, defined in terms of matched filter signal-to-noise ratio, on the detection of anomalies. Moreover, we present the application of our method for the localization in time of anomalies in the studied time-series data. We validate the results of anomaly searches on real data containing confirmed gravitational wave detections; we thus prove the generalization capabilities of our method, towards detecting GWs unknown to our deep learning models during training.

1. Introduction

The first gravitational wave (GW) detection on 14 September 2015 [1] inaugurated a new era in astrophysics. The joint observational effort of the LIGO and Virgo Collaborations, working with the Advanced LIGO (aLIGO) [2] and the Advanced Virgo (aVirgo) [3] interferometers in a global network has provided 50 GW candidate signal detections, until the suspension of LIGO-Virgo observing run O3 in Spring 2020 [4–7]. Such an impressive number of statistically-significant signal candidates allows for the verification of many theoretical models, describing various sources of GW radiation, like binary systems of black holes (BHs) and neutron stars (NSs), as well as the very nature of gravity [8]. As a result of continuous work to improve their sensitivity, the aLIGO and the aVirgo interferometers will soon probe a much larger volume of space and

expand the capability of discovering new GW sources. According to theoretical predictions, tens of BH mergers and a few NS mergers will be soon routinely registered every year [9]. Such a large number of events will deliver extraordinary information about the nature of those objects, phenomena, as well as space-time itself.

What can be observed through the GW window strongly depends on the fidelity of the data analysis methods. The GW data is always noise dominated: astrophysical signals are buried deep into the detectors' noise, caused by various sources. Some of them are associated with the environment [8], e.g. seismic and environmental activity; others originate from the detector itself [9], such as the thermal fluctuations of the mirrors and the laser beam photon shot noise [10]—time-dependent fluctuations in the laser interacting with the mirrors of the interferometer. In order to reveal the hidden GWs, the traditional approach is based on matched filtering algorithms [11]. The examples of existing pipelines utilising a matched filtering approach are PyCBC [12] and GstLAL [13]. Assuming the model gravitational waveform is known, which is not always the case for all astrophysical sources, the algorithm scans the data to match an optimal template using a template bank. Matched filtering is computationally expensive and in the case of large template banks requires a lot of computational resources. As an optimal method of filtering in stationary Gaussian noise, it is prone to contamination by non-stationary instrumental artifacts, the so-called glitches, which either mimic GW signals or reduce the quality of the collected data. Existing alternatives to matched filtering methods are based on unmodeled searches of GW burst signals. Such GWs are of short duration with an unknown or partially modelled waveform morphology related to complicated or unknown astrophysics. These signals are searched by measuring an excessive power in the time-frequency domain that occurs coherently between multiple detectors. An example of GW burst pipeline is coherent WaveBurst (cWB) [14, 15]. The sensitivity of this method is affected by short duration glitches that may occur coincident in time between multiple detectors. Versatile and rapid pipelines are needed to deal with non-stationary noise (and instrumental glitches) and to perform preliminary analysis of large amounts of data from multiple detectors.

Deep learning (DL) [16] fits this role perfectly. DL has commenced a new era of machine learning (ML), a field of computer science based on specially-designed algorithms that can generalize ('learn') from examples in order to solve problems and make predictions, without the need of being explicitly programmed [17]. DL algorithms, based on the concept of neural networks—models of neuron connectivity in the brain—are able to analyse different representations of data with varied dimensionality, like images (spectrograms, Q-transforms, Wavelet-transform) or time series. Moreover, they can quickly process large amounts of data—a requirement for the real-time (low latency) analysis for the GW interferometers.

The purpose of this work is to propose a generic, model-independent data analysis method that searches for anomalies in signals recorded by the GW detectors, based on performant DL models. The innovation of the proposed method is that our DL model 'learns' the features of the noise and detects if anomalies occur in the detected signal. We define an anomaly as an extraordinary, sparse transient data feature, outstanding with respect to the 'normal' background noise of the detector. The anomaly signal could therefore either be represented as a GW or an instrumental glitch. In particular, the GWs studied in the presented work are based on the signals emitted by the binary BH systems (BBH).

The term 'anomaly' is a well defined concept in statistics and data analysis (hence also in machine learning), sometimes also referred to as 'outlier'. The DL methods we apply are perfectly fitted for their detection and analysis role here, being especially suited to detect, compress, and reconstruct non-linearities in the input signal data. The analysis assessing the capabilities of the method is performed with the BBH signals, injected (added) to both simulated and real detector data. Once trained and tested on injections, the DL models are validated on the confirmed GW detections—real astrophysical signals registered by the detectors. Our results provide a proof-of-concept for the advantages of using DL methods in the GW data analysis, namely the processing speed and ability to capture complicated non-linear relationships in the data. These features enable our method to be potentially used as an event-trigger-generator (ETG). In this context, the advantage of our method is the computational speed—ML algorithms once trained are extremely efficient in the processing of data, in particular when used on computers equipped with graphics processing units (GPUs). By searching for anomalies in the gravitational waves' data via ML algorithms, our method could support the currently existing ETGs such as Omicron [18] and Q-transform based Omega [19].

In GW astronomy, while DL is being actively researched, it is still quite a novel method. Therefore this research fits very well in the early adoption scheme of the modern state-of-the-art development in the field, knowledge of which will become indispensable in the near future. In the following we mention a few interesting test cases. George and Huerta [20] developed the deep filtering method for signal processing, based on a system of two deep convolutional neural networks, designed to detect and estimate parameters of compact binary coalescence signal in highly noisy time-series data streams. The same authors have been involved in a group working on denoising gravitational waves with autoencoders [21]. Dreissigacker *et al* [22] have been using DL as a search method for continuous GWs emitted by spinning neutron stars. Similar

work has been conducted by Morawski *et al* [23] on the application of convolutional neural networks for classifying continuous GW signal candidates. Furthermore, DL has been used by Beheshtipour and Papa [24] for clustering continuous GW candidates. DL has also been successfully used in the classification of glitches by Razzano and Cuoco [25]. Finally DL has been used in searches for GW emitted by core-collapse supernova explosions by Iess *et al* [26].

Within GW astrophysics, application of DL in anomaly detection is a fairly new concept (see however a recent application in [27]). In different fields of astrophysics the ML anomaly detection has already been successfully implemented [28, 29] (see [30] for a recent general overview of ML in astronomy). Outside astronomy, anomaly detection has also been proposed in the search for signatures of new physics in the Large Hadron Collider data [31], where it was demonstrated to discover a specific class of highly-energetic particle jets, without the prior knowledge of their specific features.

The outline of this work is as follows. In section 2 we briefly discuss the DL architecture applied in our work and in section 3 we describe our data generation procedures. In section 4 we explain the results of the anomaly detection studies performed on both simulated and real data. Finally, in section 5 we further discuss our results and draw some conclusions.

2. Deep Learning algorithms

In this paper we employ a combination of two deep learning methods for distinguishing GWs from noise signals. We first briefly describe these learning methods and then we explain how we applied them to our problem.

2.1. The convolutional neural network and autoencoder

A *convolutional neural network* [16] (CNN) is a deep, feed-forward artificial neural network (processes the information one-way, from the input to the output), designed for processing structured arrays of data, e.g. for classifying images. The core feature of a CNN is the convolution operator, that differentiates them from regular (linear) neural networks employing simple matrix multiplication. The convolution operation envisions the input structured array (say, of dimension $m \times n$) as a sequence of overlapping elements of dimension, say, $p \times p$, often with $p < \min(m, n)$. The convolution operator inner-multiplies each such element with a so-called kernel (or filter), of the same dimension ($p \times p$), that ‘slides’ over each element of the original array. Each $p \times p$ element of the input array is thus replaced by the result of the convolution operation (a number), and we thus have dimensionality reduction. This somewhat simplified convolution concept is illustrated in figure 1.

Multiple layers are typically sequentially used, each with varying numbers and types of kernels. The kernels are chosen so that the network learns specific features in one convolution (e.g. edges, corners, etc). Convolution layers are alternated with other specialised layers, all further reducing the dimensionality, until a final activation function maps the last layer to a vector whose elements correspond to the desired options (classes) for classifying the input structured array. Each class in this vector is usually a probability, so that the class with the highest value is indicated as the recognised class.

The overlapping of the input array elements together with the sliding kernel, as well as the sequencing of applying convolution step-wise simulates the structure and operation of the visual human cortex that processes incoming images in a series of layers, by identifying progressively more complex features. CNNs have been shown to work extremely well at picking up patterns in the input structured arrays, for instance in images, and thus have rapidly become the state-of-the-art in image classification and computer vision.

In our approach we use all the advantages brought by CNNs, but we further embed them into an *autoencoder* (AE) architecture [32]. An AE [16] is a special type of deep artificial neural network that step-wise encodes and compresses the input and then it (re)constructs an output based only on the most compressed encoding, called the hidden layer, latent representation, or bottleneck. The main AE hypothesis is that some structure exists in the input data, for instance some form of correlation between the input features; such a structure can be and is learnt by an AE and consequently leveraged when forcing the input through the latent layer. (If the input features are independent of each other, then the compression and subsequent reconstruction are very difficult.)

An ideal AE is sensitive enough to the inputs to accurately build a reconstruction and insensitive enough to the same inputs so that the model does not memorize/overfit the training data. Differently formulated, this forces the model to maintain in the latent representation only the variations in data needed for reconstruction, without holding on to redundancies within the input. There are different types of AE, but in our work we use the undercomplete AE, where the dimension of the latent representation is strictly smaller than the input dimension. In this way we take care of avoiding overfitting, since the model will not be able to copy the input to the output.

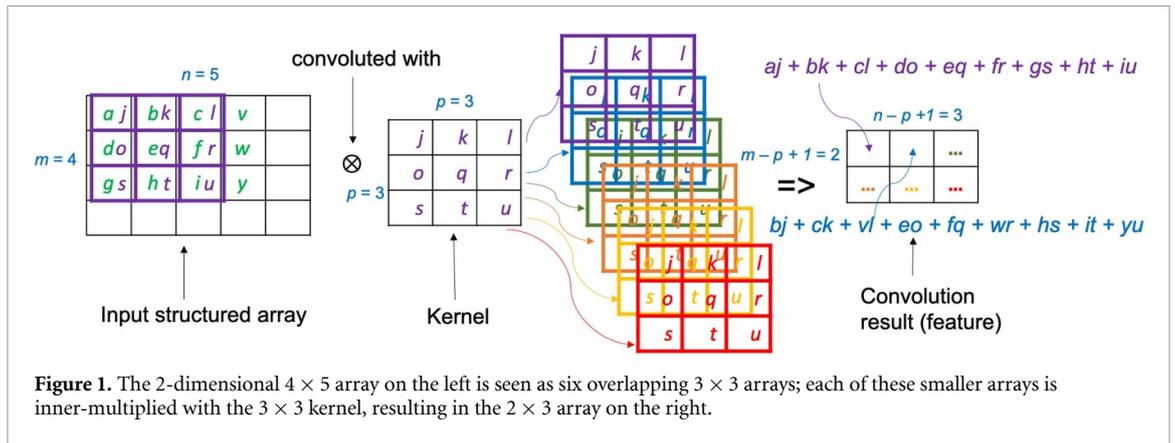


Figure 1. The 2-dimensional 4×5 array on the left is seen as six overlapping 3×3 arrays; each of these smaller arrays is inner-multiplied with the 3×3 kernel, resulting in the 2×3 array on the right.

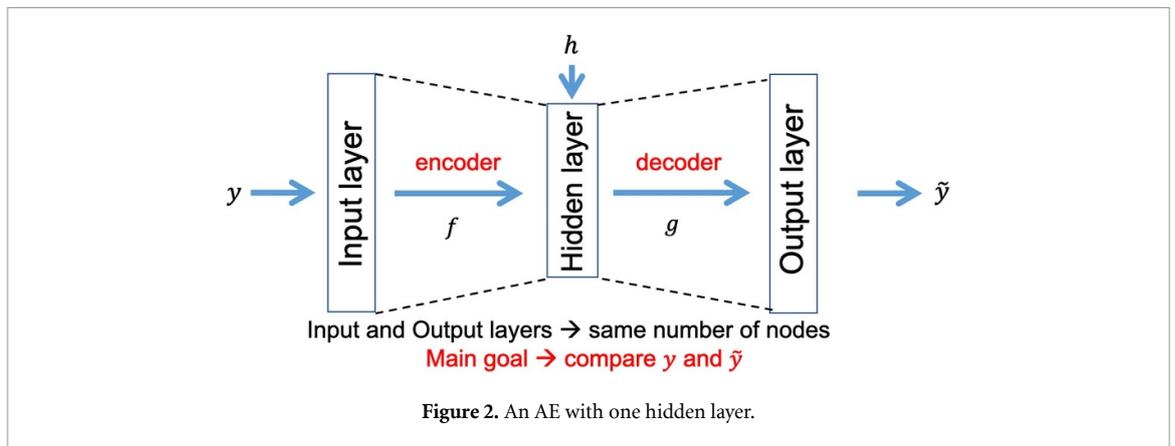


Figure 2. An AE with one hidden layer.

The simple AE in figure 2 (with only one hidden layer) works by first encoding the input $y \in \mathbb{R}^d$ to the element $h \in \mathbb{R}^p$, where $p < d$. Assuming we have the encoder activation function $f: \mathbb{R}^d \rightarrow \mathbb{R}^p$, then $h = f(Wy + b)$, where f can be sigmoid, ReLU, or something else, W is a weight matrix and b a bias factor (initialised randomly and updated iteratively during training). The decoder activation function $g: \mathbb{R}^p \rightarrow \mathbb{R}^d$ will map h to $\tilde{y} = g(\tilde{W}h + \tilde{b})$, where the activation function g , the weight matrix \tilde{W} and the bias factor \tilde{b} may be unrelated to f , W , and b . If only ReLU is used for activation and we have only one hidden layer, then we have a linear AE; if we have more hidden layers, or non-linear activation(s), then the AE becomes non-linear, which is better at detecting abstract features. Thus, in general, the encoder and decoder are proper neural networks in themselves, not simply activation functions.

The training of the AE works by attempting to minimize the reconstruction loss, most often using the mean squared error (MSE) formula [16]

$$\mathcal{L}(y, \tilde{y}) = \|y - \tilde{y}\|^2 = \left\| y - g(\tilde{W}f(Wy + b) + \tilde{b}) \right\|^2. \tag{1}$$

The training of the network works by updating the parameters $W, b, \tilde{W}, \tilde{b}$ until $\mathcal{L}(y, \tilde{y})$ is sufficiently small and further training does not decrease it anymore—in that case the network has converged. There are numerous algorithms for updating the parameters, but here we use the ADAM algorithm (adaptive moment estimation) [33], as it adapts the learning rate during training and has been empirically shown superior to other methods for large datasets, large number of parameters, as well as non-stationary input. The learning rate (training ‘step’ of updating) together with the batch size (number of training examples that use the same learning rate), the update method (also called as optimizer), the number and size of the layers and the loss function are the hyperparameters of the AE.

Since the AE is reducing dimensionality for encoding, it has often been used for only that—for instance, for feature learning. However, when compared to other dimensionality reduction techniques, such as principal component analysis (PCA), AE appears as a powerful generalization, since it is able to learn non-linear relationships in input data. While PCA attempts to discover a lower dimensional hyperplane describing the original data, AE is capable of learning non-linear manifolds of the least possible size, as illustrated in figure 3. Essentially, the AE learns a vector field for mapping input data towards lower

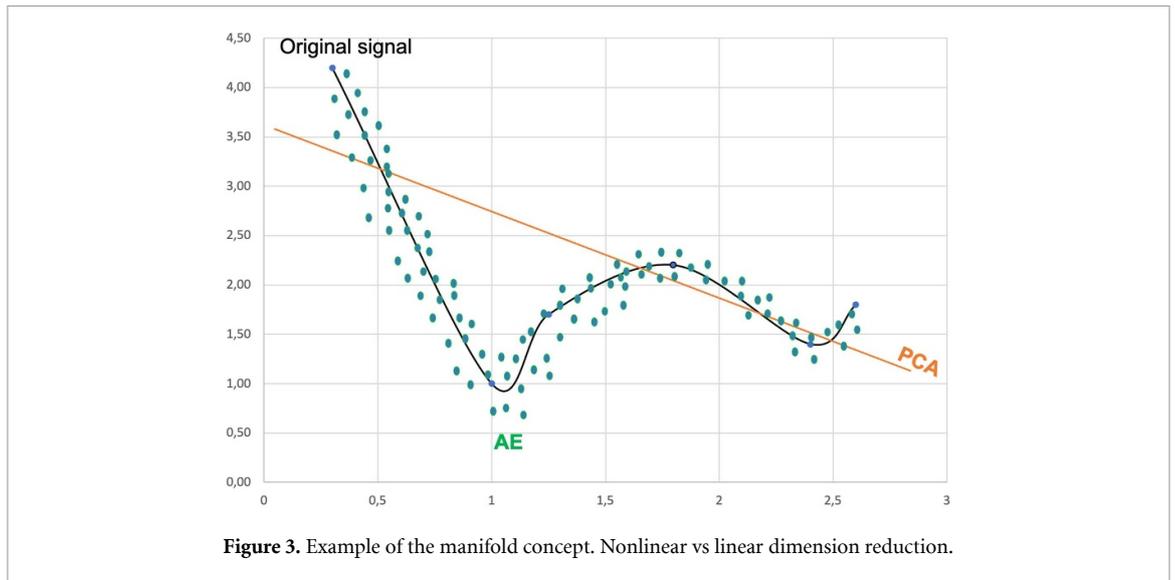


Figure 3. Example of the manifold concept. Nonlinear vs linear dimension reduction.

dimensional manifolds, that describe the high density region where input data concentrates. If the manifold accurately describes input data, then the AE has effectively learnt the input data.

2.2. Applying the CNN AE

Although CNNs were designed for the analysis of 2D data (i.e. images) [16], here we apply them for a simpler, 1D implementation, as our analysed signals are time series. The CNN AE is designed to learn two things. In case of input data instances containing only detector background noise (no anomalies), the AE is trained to reconstruct the noise as closely as possible. However, in case of instances containing an anomalous signal (GW or glitch in case of real data), the AE is trained to disregard the anomaly and reconstruct the data as if the signal was not present. By comparing the input to the output reconstructed by the AE, the anomaly present in the instance of data time series is recovered and further studied. Using the AE network is thus instrumental, since it will reconstruct only the signals it has been trained for, in our case the detector noise, and disregard anything else in its reconstruction, in our case the GW or the glitch, as noise.

Depending on the presence of anomalies in the input data, the loss value $\mathcal{L}(y, \tilde{y})$ is expected to vary. The $\mathcal{L}(y, \tilde{y})$ computed for an ‘anomalous’ input reaches higher values than in case of ‘anomaly-free’ input, since the difference between y (noise or noise with anomaly) and \tilde{y} (only noise) is larger. The difference between these signals is proportional to the amplitude of the anomaly. As a result, the AE trained on data containing stronger anomalies is expected to converge during training towards higher $\mathcal{L}(y, \tilde{y})$.

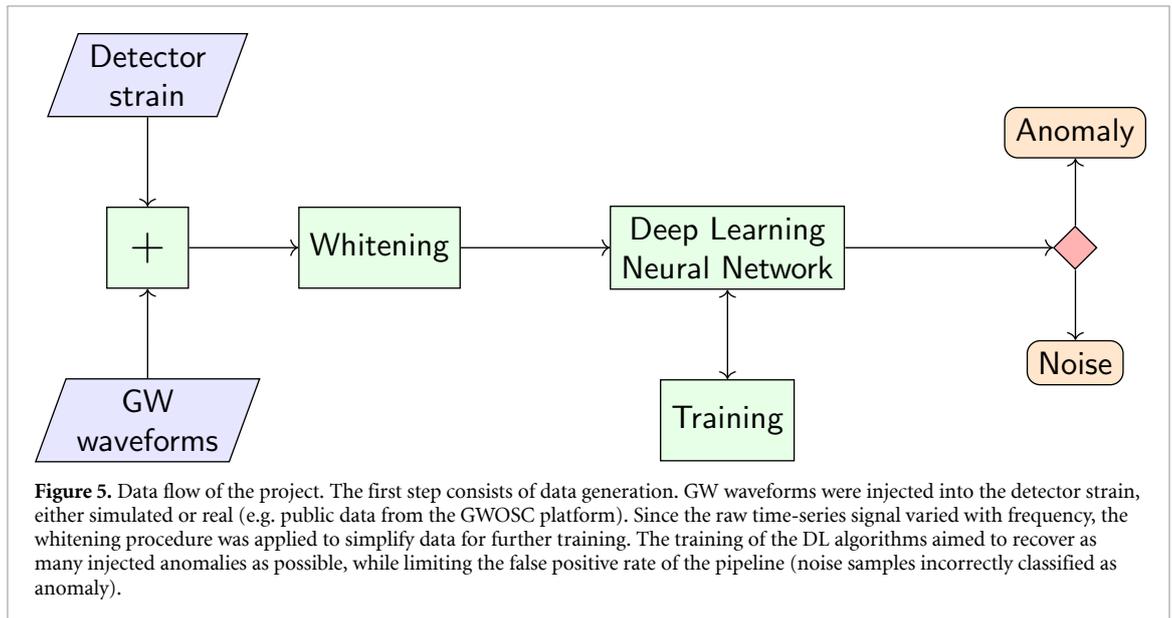
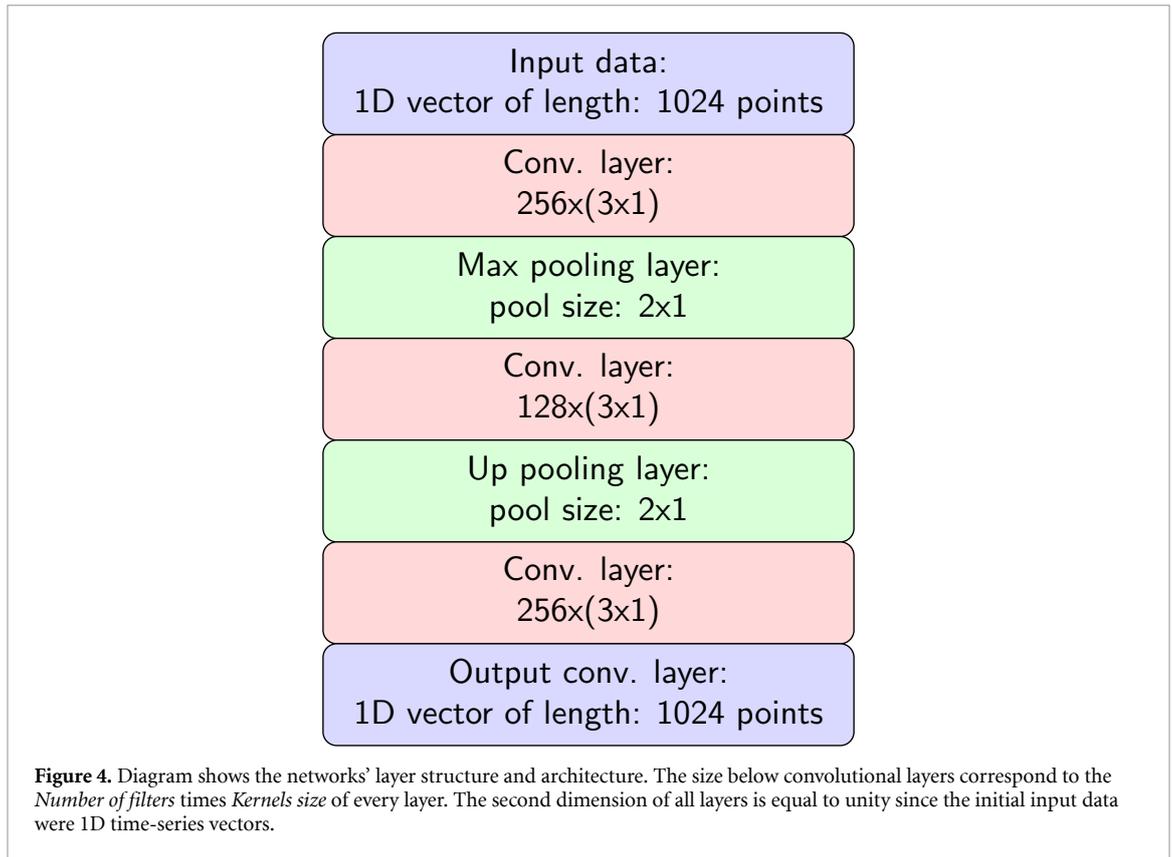
The final architecture⁶ used in section 4 was chosen based on empirical tests on the data. We tested architectures ranging from one to eight hidden layers. The final layout of the architecture is presented in figure 4. The chosen architecture reaching the minimum value for $\mathcal{L}(y, \tilde{y})$ for the fixed training data set was the network containing three hidden convolutional layers: an encoding layer, a decoding layer, and a latent representation layer in between them, with 256, 128, and 256 neurons, respectively. The kernel size was fixed for all layers to 3×1 . All but the final output layer use the ReLU as the activation function, whereas the final layer reconstructing the initial signal uses a sigmoid activation function. The other hyperparameters used for training were the ADAM optimizer [33] with learning rate of 0.0005 and batch size of 32.

In the following section, we detail our datasets used for training, validation and testing.

3. Training data sets and data flow

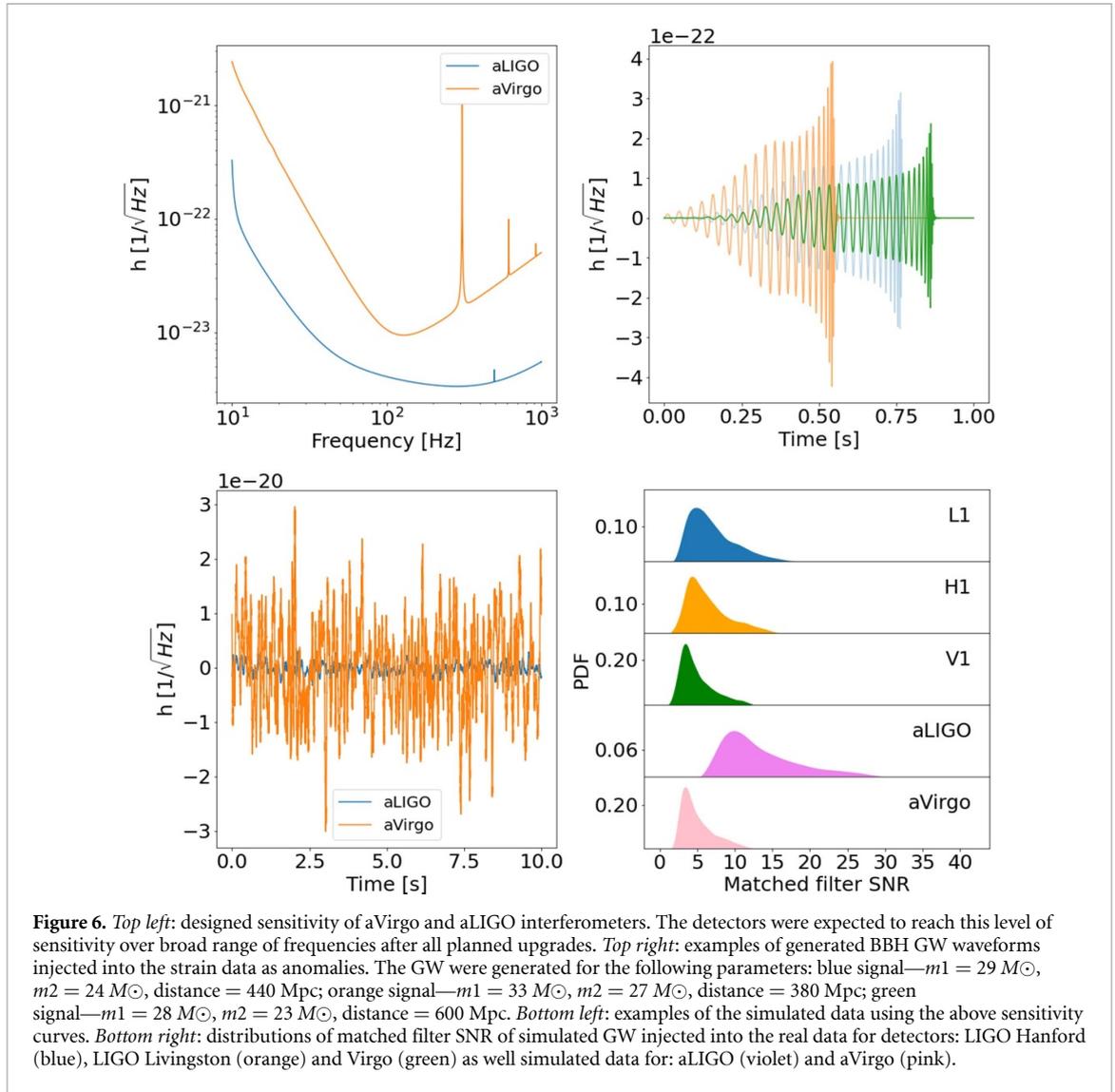
We prepared two kinds of training datasets: a simplified one by means of simulated detector strain time series (based on the colored normal distribution of noise), denoted DataSet 1 (DS1), and a realistic one based on the real LIGO-Virgo O2 observing run [39], publicly available at the Gravitational Waves Open Science Center (GWOSC) [40] and denoted DataSet 2 (DS2). In both cases we use the same general data flow presented in figure 5.

⁶ We implement the algorithm in Python [34] using the Keras/TensorFlow library [35, 36] with the GPU support. The development was performed on the NVidia Quadro P6000 (sponsorship via the NVidia GPU seeding grant). The production runs were deployed on the Prometheus cluster (Academic Computer Centre CYFRONET AGH) equipped with Tesla K40 GPU nodes, running CUDA 10.0 [37] and the cuDNN 7.3.0 [38].



The whitening mentioned in the workflow diagram removes the contribution of the stationary detector noise and re-weights the sensitivity at different frequencies [41]. As a result, the amplitude spectral density of the data becomes uniform and GW signals buried in the data are easier to search for and compare with each other. The whitening filter was re-computed separately for DS1 and DS2 as well as for every interferometer to take into account the differences in the sensitivity. The whitening procedure used for the analysis of both studied datasets was conducted in the frequency domain following pyCBC Python library modules [42].

To simulate an astrophysical GW signal emitted by a BBH we used the IMRPhenomv4 waveform model [43], which includes the binary inspiral, merger of the components and the final BH ringdown. The component BH masses m_1 and m_2 of the waveform model were chosen to be compatible with the first detected GW150914 [1]. We selected m_1, m_2 based on the initial mass function (IMF) in ranges associated with uncertainties of mass estimation for GW150914: $m_1 : 32.5\text{--}40.3 M_\odot, m_2 : 26.2\text{--}33.6 M_\odot$. For the index



of the IMF power law, we chose the value of $\alpha = -2.35$ [44]. The luminosity distances were chosen uniformly from the range between 200 and 800 Mpc, to cover a realistic range of the matched filter signal-to-noise ratio (SNR)—from 4 to 40 varying for different interferometers as shown in the bottom right plot in figure 6. The position in the sky was chosen to be optimal for every detector in a given moment of time. The examples of a few simulated GW signals are presented on the top right plot in figure 6.

The DS1 dataset was created for the two assumed sensitivity curves of the GW detectors. Each curve described the level of the detector sensitivity with respect to the frequency in such a way that the generated strain was mimicking the realistic time series output. In our analysis we used the designed sensitivity for the aVirgo from O3 run (version without squeezing) [45, 46] and the aLIGO [47] interferometers. ‘Designed’ means that the interferometers were expected to reach this level of sensitivity after all the planned upgrades. Band-pass filtering was then applied to the generated noise to remove high frequency (above 1 kHz) and low frequency (below 30 Hz, corresponding to the seismic noise) components from the data, as current interferometers are not sensitive enough to detect GWs outside that frequency range. The data was then resampled from 4096 Hz to 1024 Hz. An example of the output time series from the DS1 is shown on the bottom left plot in figure 6. Prepared in advance GW signals were injected into the generated strain and subjected to the procedure of whitening.

The DS2 (realistic) dataset was created based on the publicly available LIGO-Virgo O2 observing run, using the data stored at the GWOSC platform [40] for three interferometers: LIGO Hanford, LIGO Livingston and Virgo described in the text with the respective abbreviations $H1$, $L1$ and $V1$. For each detector, we chose six hours of data to train the DL models. For $L1$, we took segments between 1187 270 656 and 1187 295 232 (in GPS time units); for $H1$, between 1174 958 080 and 1174 982 656; whereas for $V1$, between 1187 672 064 and 1187 696 640. We injected into the strain the same GW signals as for the simulated

data. As a result, we obtained (for the same set of injections) three different distributions of matched filter SNR, since every detector had a different sensitivity. The distributions are illustrated in the bottom right plot in figure 6. For the comparison, we included additional SNR distributions of the simulated datasets. The obtained real data strain with injected anomalies was then subjected to the procedure of whitening and resampled from 4096 Hz to 1024 Hz.

In total, we generated five datasets (two simulated for aVirgo and aLIGO, and three based on real LIGO Livingston, LIGO Hanford and Virgo O2 data) which were further split into one second segments and divided for the training, validation and testing datasets (65%, 10% and 25% respectively). An additional test set, containing confirmed GW detections, has been created by using one hour of data around GPS time for each confirmed GW, whitened and resampled as described above. As confirmed GWs, we chose three BBH detections with the highest network SNR from O2 run: GW150914, GW170608 and GW170814 [39].

4. Results

The results presented below are split into subsections. The first presents the results of the anomaly searches on the simulated dataset containing injected GW. The second subsection covers searches of anomalies in the real data of Virgo and LIGO interferometers. The last subsection presents capabilities of anomaly searches on the real data containing confirmed detections of GW.

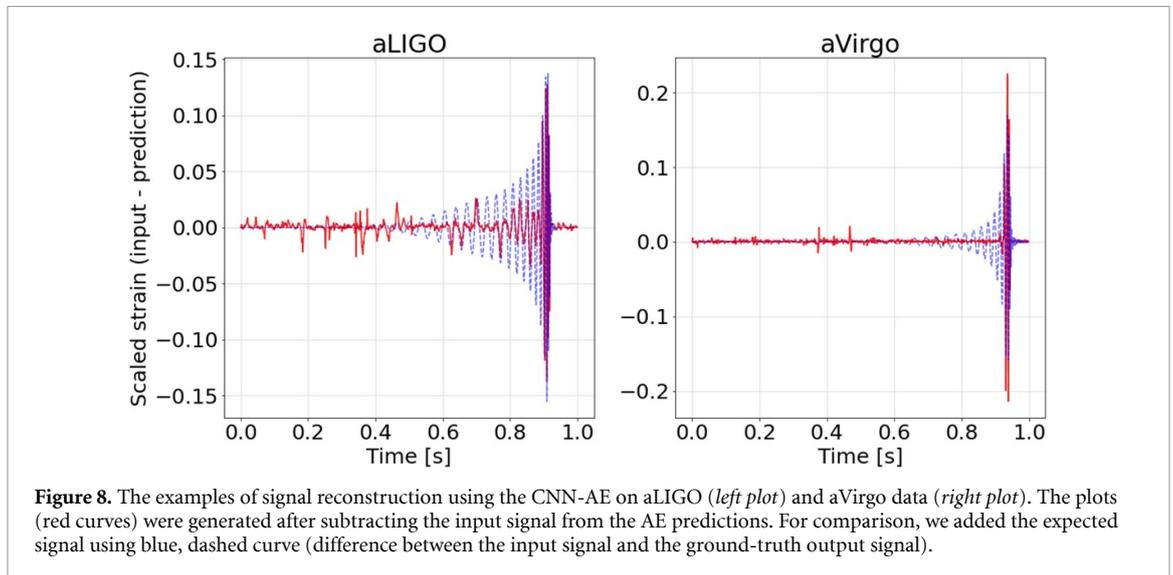
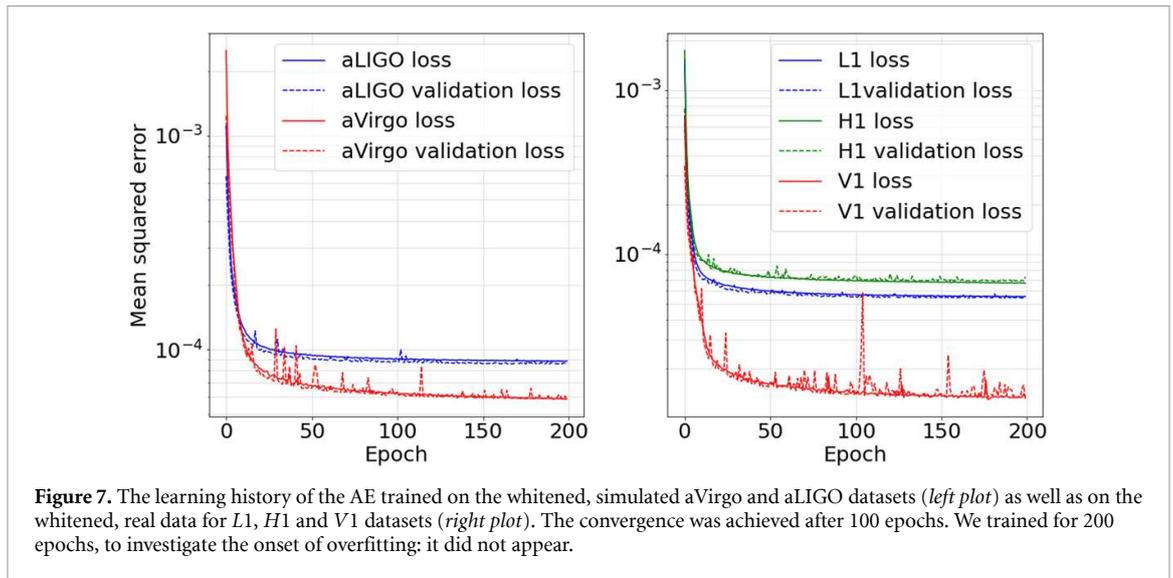
4.1. Anomaly searches on simulated data

The CNN-AE described in section 2 was first trained on the whitened, simulated data containing GWs. The convergence of the model was achieved after 100 epochs, during which the MSE loss function reached a value of around 6×10^{-5} for the aVirgo data and 10^{-4} for the aLIGO data. We extended the training for another 100 epochs to investigate the onset of overfitting. However, the overfitting did not appear and MSE fluctuated around the values mentioned above. The learning history of the AE, trained on both simulated datasets, with results for both the training and the validation sets is shown on the left plot in figure 7. For aVirgo, the same set of gravitational waveforms covered a SNR range of smaller values than for aLIGO, as a result of the worse detector sensitivity (see the top left plot in figure 6 for comparison). This, in turn, resulted in the convergence towards a lower MSE for the aVirgo dataset, since the difference between the ‘anomalous’ input and ‘anomalous-free’ reconstruction were smaller than in the case of aLIGO data (see section 2.2 for more details).

Furthermore, as described in section 2.2, the correctly trained CNN-AE reconstructed the pure detectors’ noise, regardless of the anomalies presence in the input data. Then, by subtracting the input from the reconstructed output we aimed to recover the underlying signal. We computed the mentioned differences between the input initial data and the output AE reconstruction. Examples of the results are presented in figure 8, where blue, dashed lines in the background correspond to the expected results, whereas red lines correspond to values obtained with the AE. In the majority of cases, we were able to correctly reconstruct the GW waveforms for aLIGO data. The anomaly was significantly different from the surrounding noise, with its recovered part mostly associated with the merger part of the gravitational waveforms, together with a small contribution from the inspiral part. In contrast, for the aVirgo dataset, the reconstruction was significantly worse, often dominated by the surrounding noise. In rare cases, as presented on the right plot in figure 8, the merger part was reconstructed. In the appendix A we present the summary of the match between the injected and reconstructed waveforms using $\langle x_1 | x_2 \rangle$ metric performed in the time domain.

Since the AE was able to correctly reconstruct the majority of anomalies buried in the data (at least for the aLIGO dataset), we aimed then to define a metric allowing automatic anomaly detection. We chose the MSE as such a metric, computed between the input data and the AE output as detailed in section 2.2. Figure 9 shows histograms of MSE for the two different signal types present in the studied data: noise and injected GW. As expected, values of MSE for the noise were much smaller than for the latter case and close to zero. There was a range of MSE in which histograms of both types overlapped (denoted with the burgundy colour in figure 9). Nevertheless, the majority of noise with injected GW instances in the aLIGO dataset and almost half of these instances in the aVirgo dataset had values larger than the noise. Moreover, we added the detection threshold (defined in the following paragraph) to the histograms to stress out how many of initially injected GW were correctly detected as anomalies (hatched area in figure 9).

We defined the threshold for the anomaly detection using the relation between the false positive rate (FPR) and MSE. Then, by fixing FPR at a particular value, we set the detection threshold (DT) on the corresponding MSE. In the presented analysis, we fixed FPR at 5%, resulting in the following thresholds: $DT_{\text{simV}} = 1.6 \times 10^{-5}$ for aVirgo and $DT_{\text{simL}} = 3.1 \times 10^{-5}$ for aLIGO. The results of the anomaly searches at $FPR = 5\%$ are shown in table 1 in the form of confusion matrix. Additionally, the comparison in the anomaly detection efficiency for both interferometers is shown on the left plot in the figure 10 in the form of



receiver operating characteristic (ROC) curves. Over all ranges of FPR, the aLIGO detector achieved a significantly higher detection efficiency, or true positive rate (TPR).

Presented in table 1 values quantitatively represent the results from both panels in figure 9. Anomaly row relates to the hatched area from those panels. In the case of aLIGO dataset 96% of detected anomalies are correctly related to the injected GW with minor contamination of noise samples. In case of aVirgo 59% of samples characterized by low SNR (around 10 and less) samples did not exceed the DT and contributed to the non-anomalous group. Details presenting the relation between the SNR of the injected GW and MSE of the reconstructed waveform can be found in the appendix B.

4.2. Anomaly searches on real data

Later on, the AEs were trained on whitened, real data from the O2 observational run, collected for three existing interferometers: *V1*, *L1* and *H1* with injected BBH gravitational waveforms. The right plot in figure 7 presents the learning history of the AE trained on the dataset for every detector. As in the case of the simulated data, AE reached convergence after around 100 epochs. We again prolonged the training to investigate the onset of overfitting, which did not appear. Since the difference between the ‘anomalous’ input and the ‘anomalous-free’ reconstruction for *V1* was the smallest among the considered datasets (as a result of the SNR smallest range), the AE during training converged towards the lowest MSE. Adequately, a smaller range of SNRs for the *H1* dataset with respect to the *L1* dataset resulted in the difference between the respective values of MSE (for *L1* being higher, and for *H1* being lower).

The ability of the AE to reconstruct the detectors’ noise was investigated as previously in the case of simulated data. The differences between the input strain and AE reconstructions were compared with the

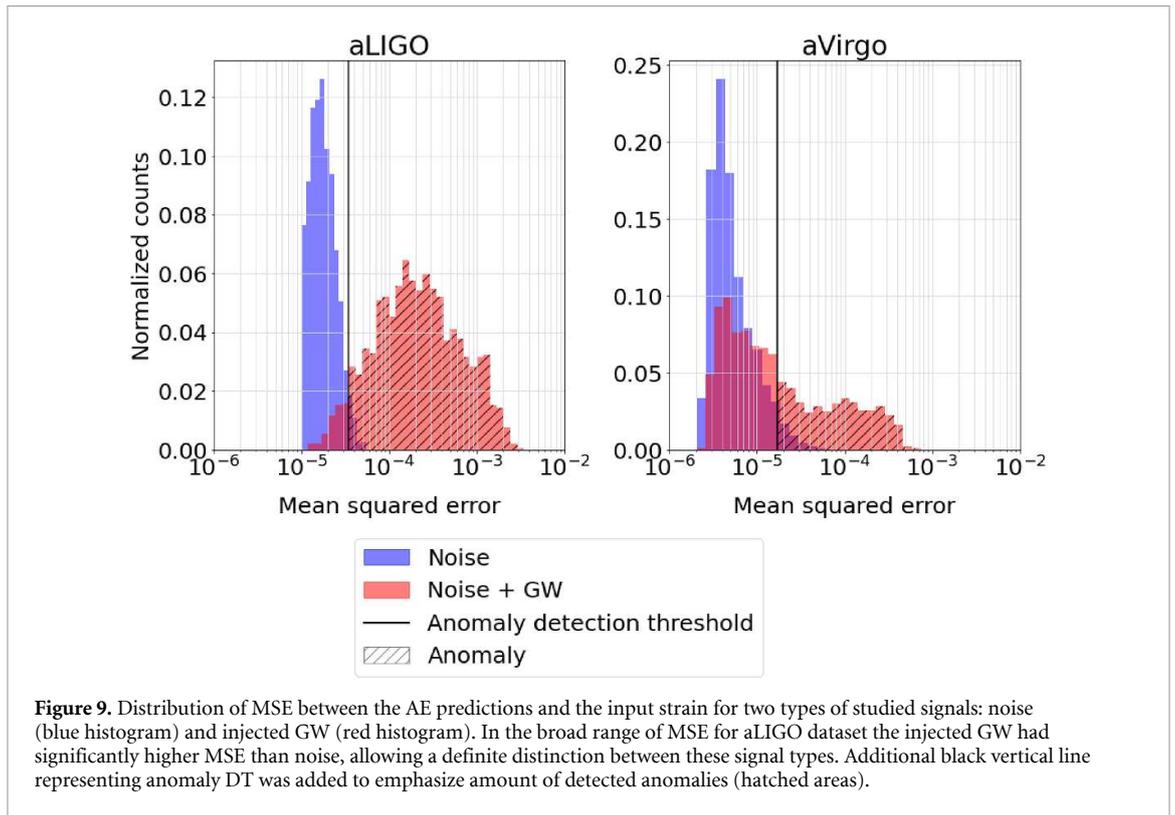


Figure 9. Distribution of MSE between the AE predictions and the input strain for two types of studied signals: noise (blue histogram) and injected GW (red histogram). In the broad range of MSE for aLIGO dataset the injected GW had significantly higher MSE than noise, allowing a definite distinction between these signal types. Additional black vertical line representing anomaly DT was added to emphasize amount of detected anomalies (hatched areas).

Table 1. Results of anomaly detection of CNN-AE at FPR = 5% for aLIGO and aVirgo dataset in the form of confusion matrix. Columns relate to the ground-truth values whereas rows to the predictions. For aLIGO dataset significant majority of detected anomalies corresponded to the data samples with injected GW. However in case of aVirgo more than a half of data samples with injected GW did not exceed the DT_{simV} as a result of low SNR (see figure 6 for comparison between aLIGO and aVirgo GW SNR distributions).

	aLIGO		aVirgo	
	Injected GW	Noise	Injected GW	Noise
Anomaly	96%	5%	41%	5%
Non-anomaly	4%	95%	59%	95%

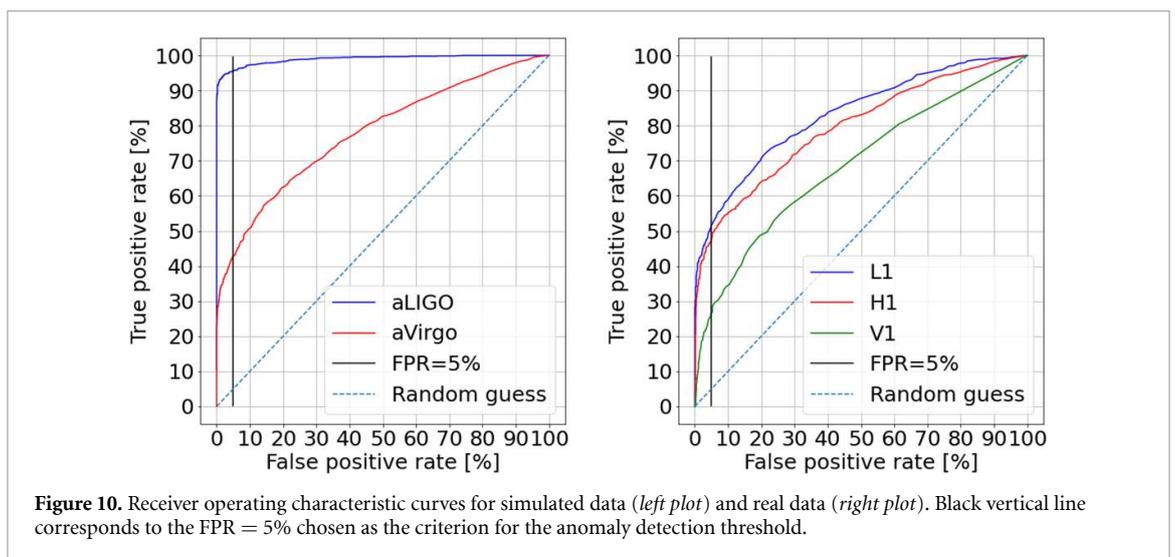
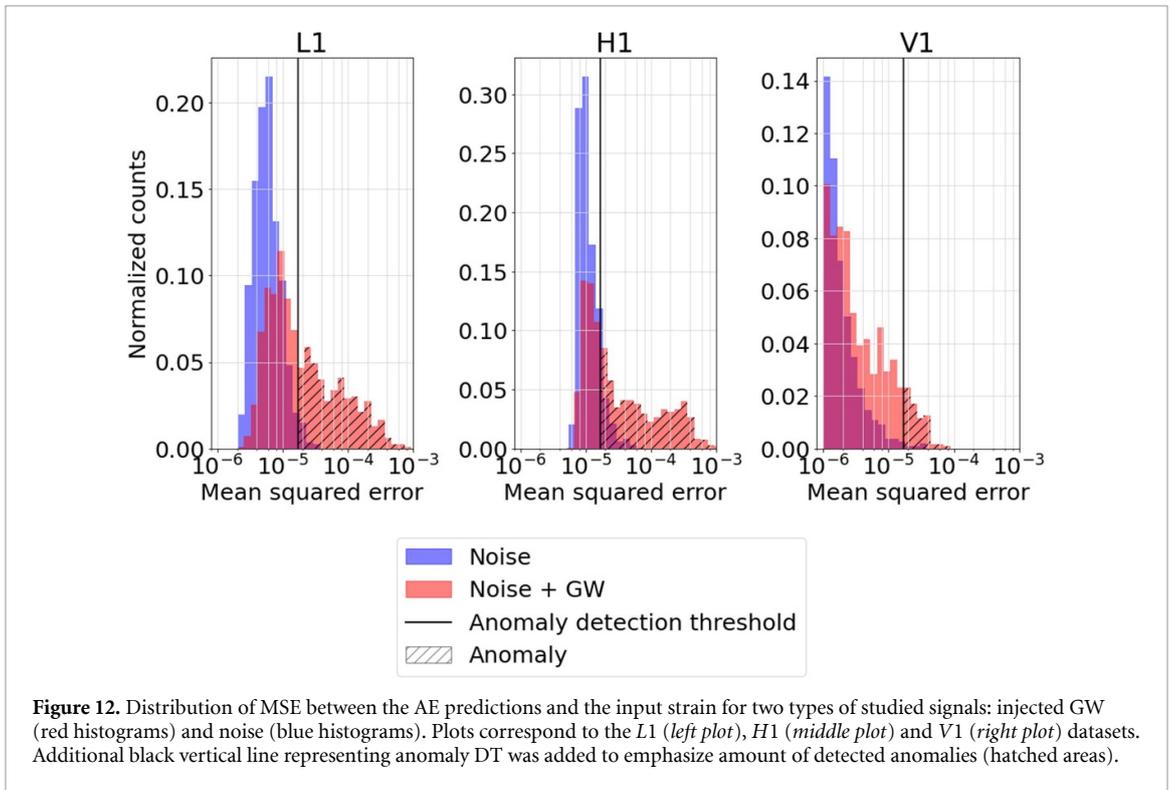
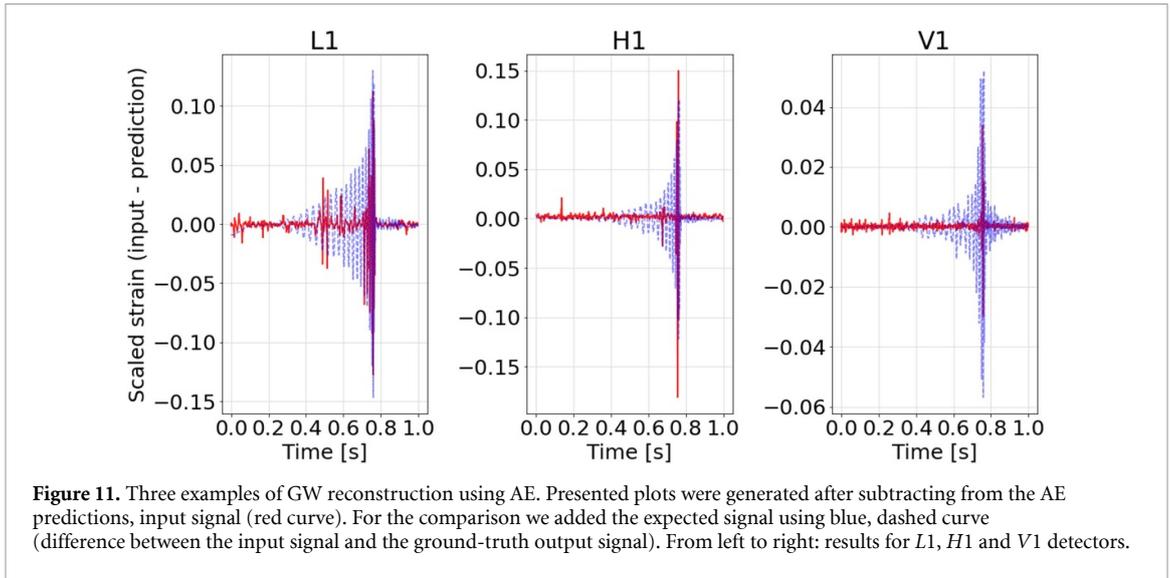


Figure 10. Receiver operating characteristic curves for simulated data (left plot) and real data (right plot). Black vertical line corresponds to the FPR = 5% chosen as the criterion for the anomaly detection threshold.

expected values. Examples of this comparison are presented in figure 11. The summary of the match between the injected and the reconstructed waveforms is presented in appendix A. The AE trained on L1 data achieved the best results, since the GW injected into the strain were extracted to the greatest extent in the merger part and partially in the inspiral part. On the other hand, the AE trained on the other datasets

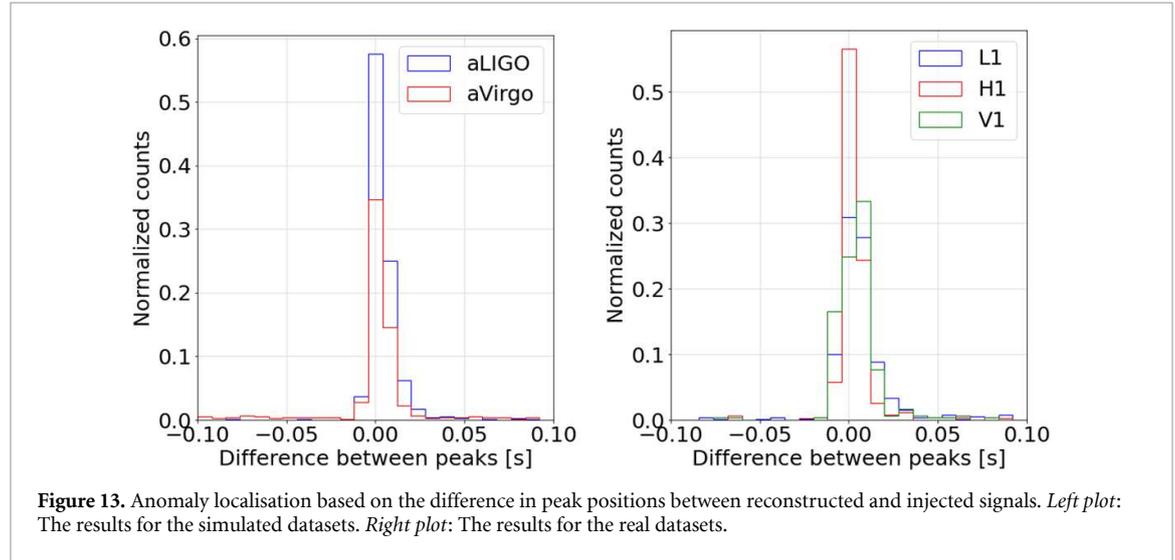


reconstructed mainly the merger part. Overall, the AE seemed to fail to reconstruct lower amplitudes and frequencies of the GW signal (the inspiral and ringdown part).

To compute the threshold for the anomaly detection, we generated histograms of the MSE for every detectors' dataset and compared its values with the FPR. The results are shown in figure 12. The anomalies covered a greater range of MSE values than the noise, with an overlapping range varying for different datasets (coloured in burgundy in figure 12). In the case of $L1$ data, this region was the smallest whereas for $V1$, it was the largest. As in the case of simulated data, we defined a detection threshold for anomalies assuming $FPR = 5\%$, which resulted in the following thresholds: $DT_{L1} = 1.3 \times 10^{-5}$ for $L1$, $DT_{H1} = 2.2 \times 10^{-5}$ for $H1$ and $DT_{V1} = 4.3 \times 10^{-6}$ for $V1$. The results of the anomaly searches on real datasets are present in table 2 in the form of confusion matrix. Anomaly row relates to the hatched area on panels in figure 12. In the case of $L1$ and $H1$ datasets, around half of detected anomalies are correctly related to the injected GW. Samples that did not exceed corresponding DT had low SNR. That was also the case of the $V1$ dataset where only 27% of samples exceeded DT_{V1} . Details presenting the relation between the SNR of the injected GW and MSE of the reconstructed waveform for the real datasets can be found in the appendix B.

Table 2. Results of anomaly detection of CNN-AE at FPR = 5% for $L1$, $H1$ and $V1$ datasets in the form of confusion matrix. Columns relate to the ground-truth values whereas rows to the predictions. For all datasets significant majority of detected anomalies correctly corresponded to the data instances with injected GW. However, more than a third part of non-anomalous class (samples that did not exceed DT for a given detector) related to the low SNR injected GW (see figure 6 for comparison between $L1$, $H1$ and $V1$ GW SNR distributions).

	$L1$		$H1$		$V1$	
	Inj. GW	Noise	Inj. GW	Noise	Inj. GW	Noise
Anomaly	52%	5%	50%	5%	27%	5%
Non-anomaly	48%	95%	50%	95%	73%	95%



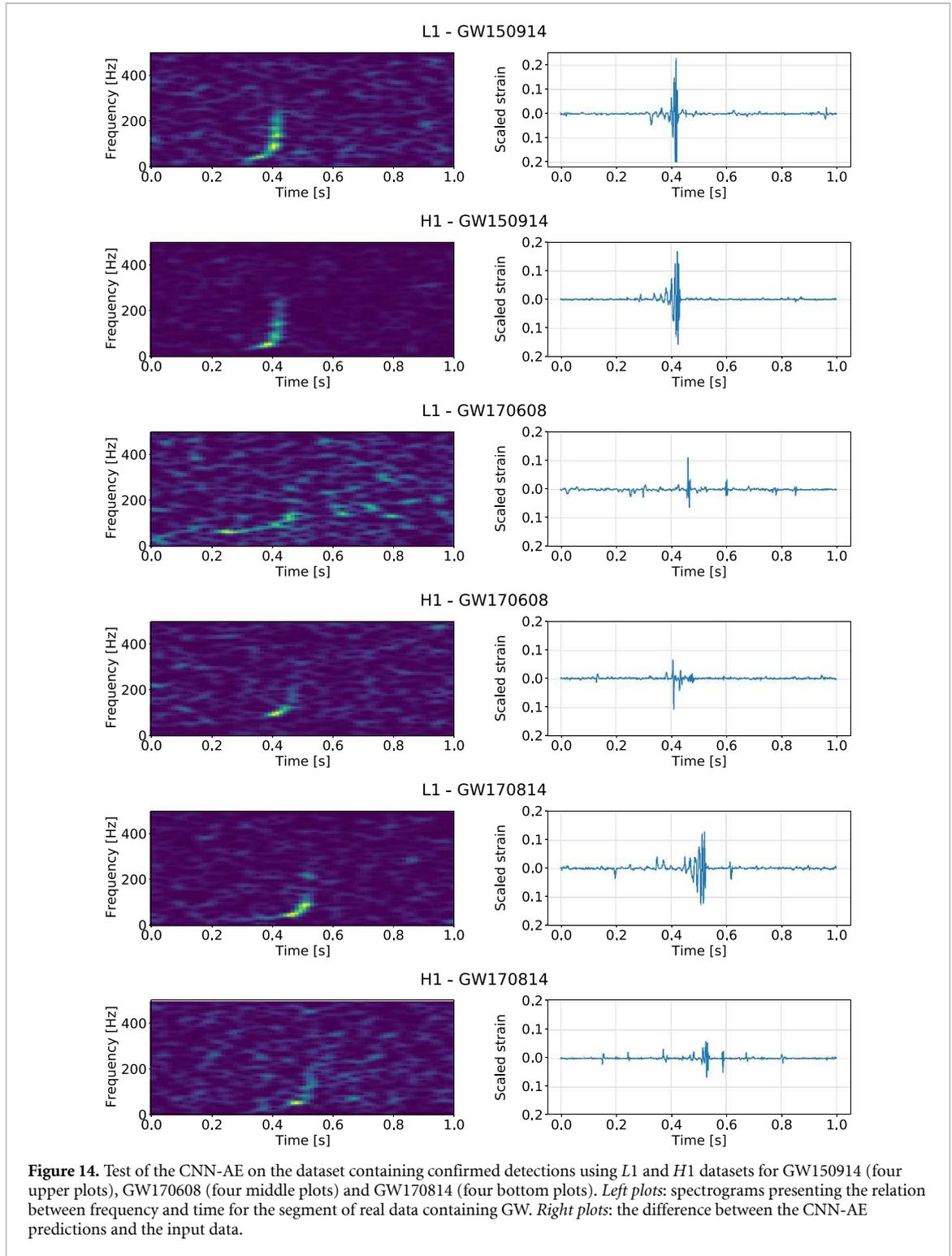
Additionally, we performed tests on the localisation in time of anomalies detected by our method. We compared the known times of the GW injection into the data with the time of the reconstructed signal. Specifically, we subtracted the times corresponding to the maximum amplitude peaks of both signals, and plotted histograms of the resulting differences for all the anomalies that exceeded computed previously DT. For comparison, we performed this procedure also for the simulated datasets. The results are shown in figure 13. In all the studied cases, around 95% of detected anomalies were localised within 0.05 s intervals around the injection times. We conclude that this feature may be useful not only for detection and reconstruction, but also potentially for other applications, such as the localisation of signals in the data from many detectors, and subsequent sky localisation of the sources.

4.3. Anomaly searches on confirmed GW detections

Using a selection of real GW detections provided by the LIGO-Virgo collaboration on the GWOSC platform [40], we tested the AE on three relatively strong signals from the GWTC-1 O1-O2 catalog [39]: GW150914 [1], GW170608 [48] and GW170814 [49]. The reported network SNR (square root of a sum of squares of SNRs from individual detectors ρ_i) $\rho_{\text{net}} = \sqrt{\sum_i \rho_i^2}$ is $\simeq 24$ for GW150914, $\simeq 15$ for GW170608 and $\simeq 16$ for GW170814 [40]. Assuming two equally sensitive detectors, each of them would measure SNR of $\simeq 17$ for GW150914, $\simeq 10$ for GW170608 and $\simeq 11$ for GW170814; note that the GW170814 was a three-detector event, with the single-detector SNRs in $H1$, $L1$ and $V1$ equal to 7.3, 13.7, and 4.4, respectively [49]. In reality, due to differences in sensitivity the detectors registered the signals with different SNRs, which were nevertheless near the single-detector SNR detection threshold, established by the FPR = 5% condition.

After whitening, test data were fed into the AE. Then, the reconstructed values were subtracted from the input data. The results of subtraction for GW150914 are shown on the two upper, right plots in figure 14 for the both LIGO detectors. The presented signals resulted in the $\text{MSE}_{L1} = 3.2 \times 10^{-4}$ and $\text{MSE}_{H1} = 1.0 \times 10^{-3}$. For both detectors, MSE had significantly higher values than the corresponding detection thresholds of FPR = 5%.

In case of GW170608, the AE detected the event; however, the reconstructed signal was limited to the merger part for both the LIGO detectors as shown on the two middle, right plots in figure 14. A potential explanation for this weaker reconstruction of this particular GW is related to the different mass ranges of GW170608 and the GWs used for the training of the AE. GW170608 had BBH component masses



corresponding to around $m_1 = 11.0$ and $m_2 = 7.6 M_\odot$ which were substantially smaller than the masses used in the data generation (see section 3 for more details). We note that the H1 detector was nominally outside of observing mode at the time of this event. This data release includes H1 data around the time of this event, by using a modified segment list, as was done for the published analysis [39, 40, 48]. Nevertheless, the AE detected GW170608, proving its generalisation capabilities towards recognising gravitational waveforms it was not trained for. This is a sure advantage of our proposed method, when compared to the matched filtering method. Furthermore, the MSE values for both detectors had slightly higher values above the detection thresholds at FPR = 5%: $MSE_{L1} = 5.3 \times 10^{-5}$ and $MSE_{H1} = 4.1 \times 10^{-5}$.

The last studied test case, GW170814, was detected in both LIGO detectors with a substantial part of the waveform being recovered, as shown on the two bottom, right plots in figure 14. For the V1 data, AE failed to detect the event evidently due to low reported SNR of $\simeq 4$ [49]. The MSE for the L1 and H1 datasets were above the data threshold, at 5% and equal to $MSE_{L1} = 2.2 \times 10^{-4}$, $MSE_{H1} = 2.2 \times 10^{-4}$, whereas for V1 the MSE was below the threshold: $MSE_{V1} = 1.8 \times 10^{-6}$.

These results confirm that our proposed method of using CNN-AE for anomaly searches is able to detect real GWs, even though the deep learning model was trained on relatively uncomplicated datasets, based on the information related to a particular GW waveform models, varying with respect to the limited range of masses and distances.

5. Summary

In this paper, we proved that AEs are a potentially powerful method for anomaly searches in GW data. A relatively simple AE, consisting of only three hidden layers, was capable of detecting anomalies, defined in terms of transient BBH GWs (as well glitches in real data), and trained on either simulated or real data. Moreover, our proposed method was able to detect as anomalies all three confirmed GWs used as a test case and even partially reconstruct the waveforms of the underlying signals.

In the proposed method we introduced a metric allowing for the automatic detection of anomalies. The chosen metric was MSE. Using this metric, we defined the threshold for the anomaly detection by associating MSE with FPR. The results presented in the manuscript referred to $FPR = 5\%$. At such a threshold we were able to detect from the LIGO data set, almost all of the injected GW into the simulated dataset, as well as around 50% for real detector data. In the case of Virgo, for simulated data around half of injected signals were detected and quarter for real data. The reason for worse results on real data taken from the O1 to O2 observational run, was the substantial difference in the sensitivity with respect to the simulated counterpart. The sensitivity of real detectors were overall worse than the designed one. However, the real sensitivity significantly improved for the O3 observational run after detector upgrades. Once the data from O3 becomes public, the anomaly searches of the proposed CNN-AE method are expected to improve.

Our method also proved to be useful in the precise time localisation of anomalies. Anomalies exceeding the detection threshold for all the studied datasets (real and simulated) turned out to be localised with accuracy below 0.05 s. Localisation of the peak amplitude may be particularly useful for the localisation of GW sources on the sky in multi-detector analysis. However, such an application of our method requires a separate study.

Finally, the successful detection of GW170608 proved the generalisation capabilities of our AE, towards the detection of GWs with parameters that are different than those of the GWs used for the AE training and in the data nominally outside of the observing mode.

Among the future projects we are considering are applications of recurrent neural networks instead of CNN in the anomaly searches as an alternative suited for the time-series data as well as anomaly searches of different GW types such as core collapse supernova signals.

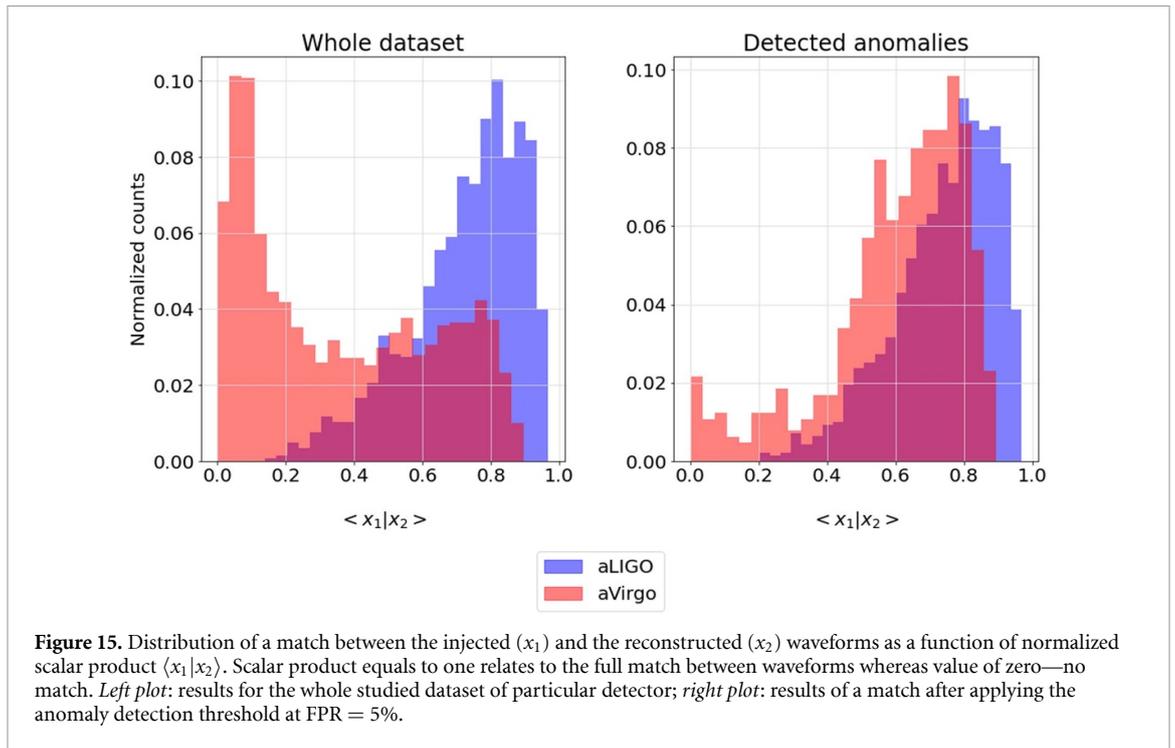
Data availability statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments

The work was partially supported by the Polish NCN Grant Nos. 2016/22/E/ST9/00037, 2017/26/M/ST9/00978 and 2020/37/N/ST9/02151, as well as the European Cooperation in Science and Technology COST action G2Net (CA17137). The Quadro P6000 used in this research was donated by the NVIDIA Corporation. The production computations were supported in part by PL-Grid Infrastructure and by the MNiSW grant for expansion of the strategic IT infrastructure for science.

This research has made use of data, software and/or web tools obtained from the Gravitational Wave Open Science Center (www.gw-openscience.org), a service of LIGO Laboratory, the LIGO Scientific Collaboration and the Virgo Collaboration. LIGO Laboratory and Advanced LIGO are funded by the United States National Science Foundation (NSF) as well as the Science and Technology Facilities Council (STFC) of the United Kingdom, the Max-Planck-Society (MPS), and the State of Niedersachsen/Germany for support of the construction of Advanced LIGO and construction and operation of the GEO600 detector. Additional



support for Advanced LIGO was provided by the Australian Research Council. Virgo is funded, through the European Gravitational Observatory (EGO), by the French Centre National de Recherche Scientifique (CNRS), the Italian Istituto Nazionale di Fisica Nucleare (INFN) and the Dutch Nikhef, with contributions by institutions from Belgium, Germany, Greece, Hungary, Ireland, Japan, Monaco, Poland, Portugal, Spain.

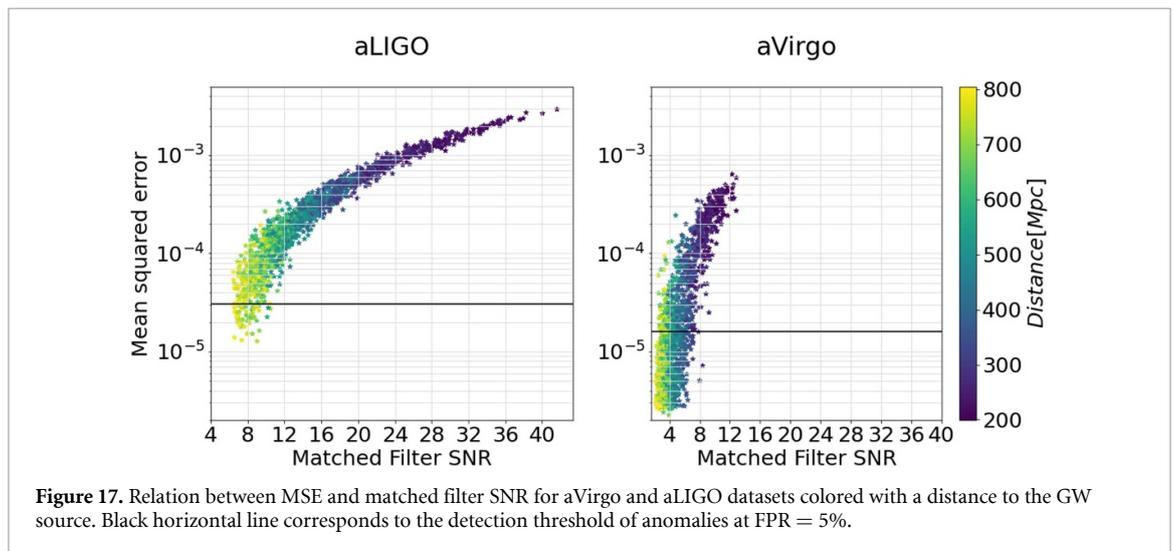
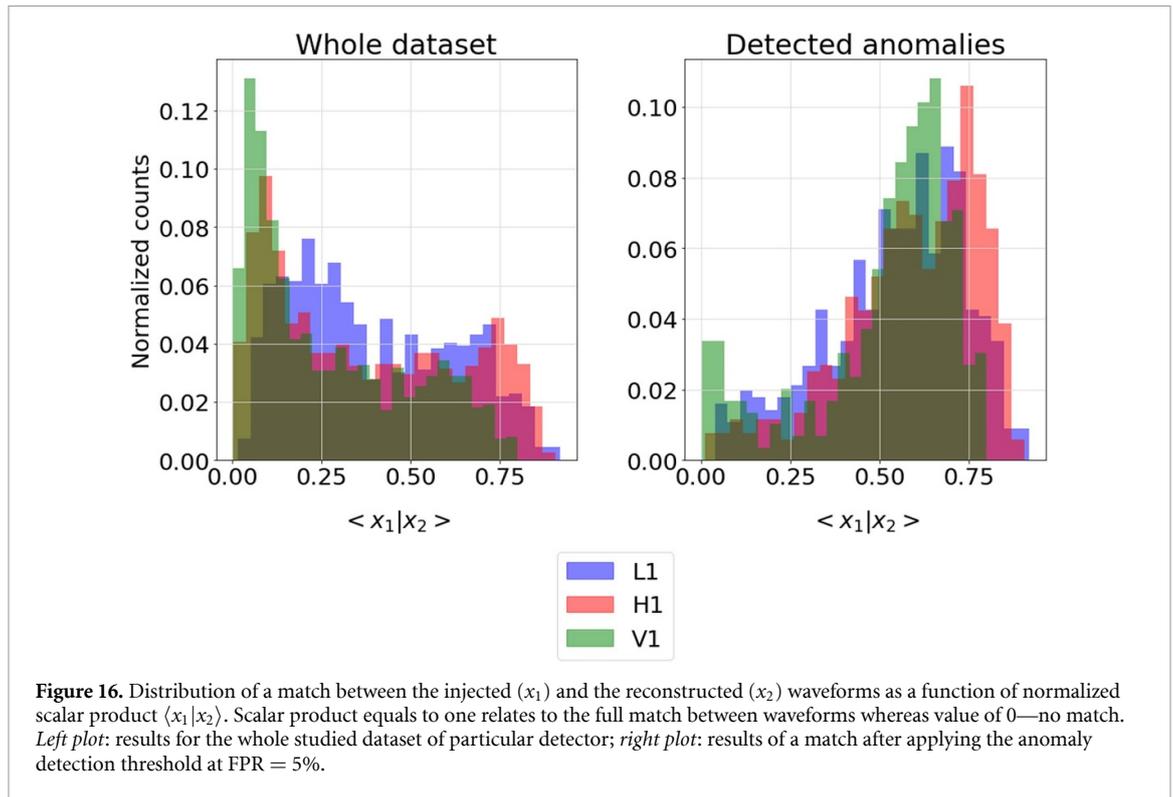
Appendix A. Match/overlap between the injected and reconstructed waveforms

A.1. Simulated dataset

To measure the match between the injected and the reconstructed waveforms we used the normalized scalar product $\langle x_1 | x_2 \rangle$ in the time domain resulting in values in a range (0, 1). Zero related to no match, whereas one to full match between the waveforms. Presented results in figure 15 corresponds to the whole studied dataset for aLIGO and aVirgo detectors (left panel) as well as samples exceeding the anomaly detection thresholds (right panel). Applying respective DT allowed to substantially reduce number of samples with no match between the waveforms as a result of low SNR of injected GW. Samples exceeding DT were reconstructed to a greater extent which resulted in $\langle x_1 | x_2 \rangle$ closer to one.

A.2. Real dataset

The same metric as in case of simulated dataset was used to study the match between the injected and the reconstructed waveforms for the real datasets. Presented in figure 16 results show the similarities in the match between consecutive datasets ($L1$, $H1$ and $V1$) as well as the effect of applying the anomaly detection threshold. Samples with $\langle x_1 | x_2 \rangle$ close to zero had low SNR. As a result they were poorly reconstructed which in turn translated into low value of MSE. Samples exceeding respective DT were reconstructed to a greater extent as in the case of simulated data. However, overall match was worse—the mean values of $\langle x_1 | x_2 \rangle$ for real datasets were around 0.6, whereas for simulated data 0.8.

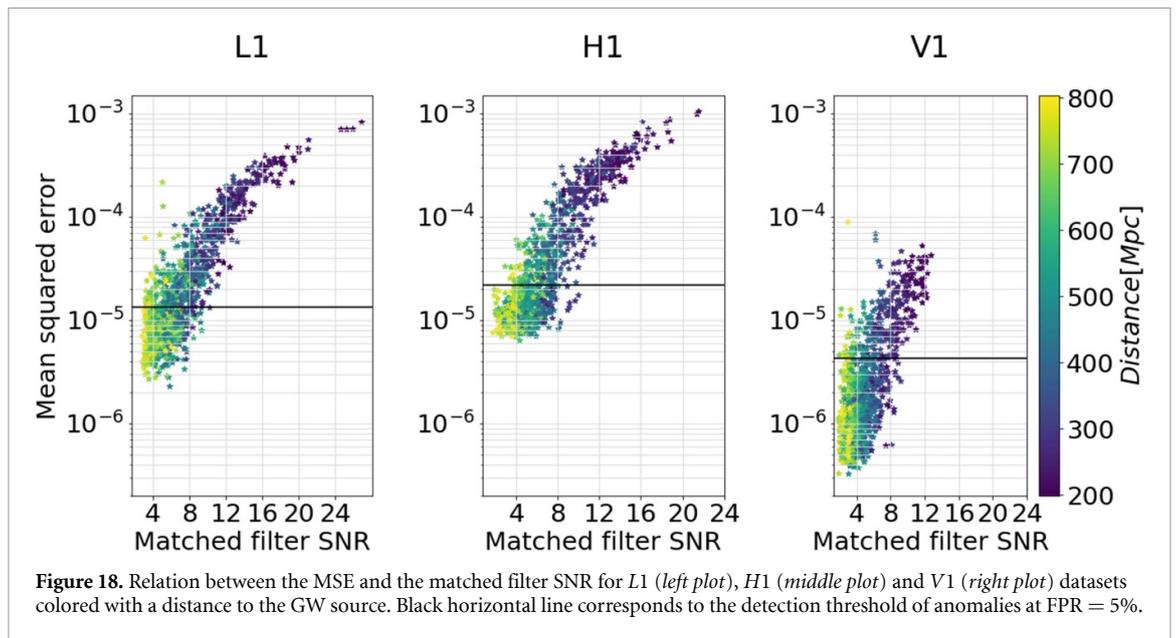


Appendix B. Signal-to-noise ratio vs mean squared error

B.1. Simulated dataset

For the aLIGO dataset above $\text{SNR} = 20$, the MSE-SNR relation was almost linear, with a small spread of individual data instances along MSE, as shown in figure 17. Whereas with the decline of SNR, the spread of MSE significantly increased, characterized by the non-linearity in the MSE-SNR relation. Anomalies around the same SNR for values below ten varied up to an order of magnitude in MSE. Manual inspection of data samples containing anomalies of low SNR provided an explanation of this behaviour. In the analysed samples, only the merger part of the gravitational waveform was recovered. For lower SNRs (below ten) the recovery was partial and dependent on the variability of the noise. If the amplitude of the noise in a given data segment was small comparing to the injected GW signal, the resulting MSE was higher than in the case of noise samples with larger amplitudes. Overall, for the aLIGO dataset, the susceptibility of AE to the local variability of the noise was inversely proportional to the SNR of the injected anomaly.

In case of the aVirgo dataset, the mentioned susceptibility was more significant. Matched filter SNRs for all injected GWs covered a smaller range of values than for aLIGO (compare SNR ranges on the bottom right



plot in figure 6 for aLIGO and aVirgo datasets). In the majority of studied cases, the recovery of the anomaly was partial and limited to the merger part.

B.2. Real dataset

The relation between SNR and MSE presented similar features as for the simulated data discussed above. The variability of MSE for samples of similar SNR was largest among the weakest anomalies as shown in figure 18. The increase of the matched filter SNR led to the decrease of the spread in MSE, and thus their relation became more linear. This was the case for the AEs trained on the *L1* and *H1* datasets. In contrast, the AE trained on the *V1* dataset was more susceptible to the local variability of the noise. Since the injected anomalies had a small SNR for *V1* (majority of injected GW had SNR below ten), their amplitude was significantly lower than the detectors noise. As a result, the anomaly detection depended on the variability of the noise, which had a random character.

ORCID iDs

Filip Morawski  <https://orcid.org/0000-0002-6194-8239>

Michał Bejger  <https://orcid.org/0000-0002-4991-8213>

Elena Cuoco  <https://orcid.org/0000-0002-6528-3449>

Luigia Petre  <https://orcid.org/0000-0002-0648-3301>

References

- [1] Abbott B P et al 2016 *Phys. Rev. Lett.* **116** 061102
- [2] Aasi J et al 2015 *Class. Quantum Grav.* **32** 074001
- [3] Acernese F et al 2015 *Class. Quantum Grav.* **32** 024001
- [4] LIGO Scientific Collaboration 2020 O3 summary (available at: www.ligo.caltech.edu/WA/news/ligo20200326)
- [5] Virgo Collaboration 2020 Virgo status (available at: www.virgo-gw.eu/status.html)
- [6] LIGO Scientific Collaboration and Virgo Collaboration 2020 Gravitational-wave candidate event database (available at: <https://gracedb.ligo.org/superevents/public/O3/>)
- [7] Abbott R et al 2020 GWTC-2: compact binary coalescences observed by LIGO and virgo during the first half of the third observing run (arXiv:2010.14527)
- [8] Abbott B P et al 2019 *Astrophys. J. Lett.* **882** L24
- [9] Acernese F et al 2016 *Phys. Rev. D* **93** 122003
- [10] Kumar A, Saminadayar L, Glattli D C, Jin Y and Etienne B 1996 *Phys. Rev. Lett.* **76** 2778–81
- [11] Owen B and Sathyaprakash B 1999 *Phys. Rev. D* **60** 022002
- [12] Usman S A et al 2016 *Class. Quantum Grav.* **33** 215004
- [13] Sachdev S et al 2019 (arXiv:1901.08580)
- [14] Klimentenko S et al 2016 *Phys. Rev. D* **93** 042004
- [15] Klimentenko S et al 2008 *Class. Quantum Grav.* **25** 114029
- [16] Goodfellow I, Bengio Y and Courville A 2016 *Deep Learning* (Cambridge, MA: MIT Press)
- [17] Samuel A L 1959 *IBM J. Res. Dev.* **3** 210–29
- [18] Robinet F, Arnaud N, Leroy N, Lundgren A, Macleod D and McIver J 2020 *SoftwareX* **12** 100620

- [19] Chatterji S, Blackburn L, Martin G and Katsavounidis E 2004 *Class. Quantum Grav.* **21** S1809–18
- [20] George D and Huerta E 2018 *Phys. Lett. B* **778** 64–70
- [21] Shen H, George D, Huerta E A and Zhao Z 2019 Denoising gravitational waves with enhanced deep recurrent denoising auto-encoders *ICASSP 2019—2019 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)* pp 3237–41
- [22] Dreissigacker C, Sharma R, Messenger C, Zhao R and Prix R 2019 *Phys. Rev. D* **100** 044009
- [23] Morawski F, Bejger M and Ciecielag P 2020 *Machine Learning: Science and Technology* **1** 025016
- [24] Beheshtipour B and Papa M A 2020 *Phys. Rev. D* **101** 064009
- [25] Razzano M and Cuoco E 2018 *Class. Quantum Grav.* **35** 095016
- [26] Iess A, Cuoco E, Morawski F and Powell J 2020 *Machine Learning: Science and Technology* **1** 025014
- [27] Corizzo R, Ceci M, Zdravetski E and Japkowicz N 2020 *Expert Syst. Appl.* **151** 113378
- [28] Giles D and Walkowicz L 2019 *MNRAS* **484** 834–49
- [29] D’Addona M, Riccio G, Cavuoti S, Tortora C and Brescia M 2020 Anomaly detection in astrophysics: a comparison between unsupervised deep and machine learning on kids data (arXiv:2006.08235)
- [30] Baron D 2019 (arxiv:1904.07248)
- [31] Farina M, Nakai Y and Shih D 2020 *Phys. Rev. D* **101** 075021
- [32] Baldi P 2011 Autoencoders, unsupervised learning and deep architectures *Proc. 2011 Int. Conf. on Unsupervised and Transfer Learning Workshop (UTLW’11)* vol 27 (JMLR.org) pp 37–50
- [33] Kingma D P and Ba J 2014 (arxiv:1412.6980)
- [34] Van Rossum G and Drake F L 2009 *Python 3 Reference Manual* (Scotts Valley, CA: CreateSpace)
- [35] Chollet F et al 2015 Keras (available at: <https://keras.io>)
- [36] Abadi M et al 2015 TensorFlow: large-scale machine learning on heterogeneous systems software available from tensorflow.org (available at: www.tensorflow.org)
- [37] Nickolls J, Buck I, Garland M and Skadron K 2008 *Queue* **6** 40–53
- [38] Chetlur S, Woolley C, Vandermersch P, Cohen J, Tran J, Catanzaro B and Shelhamer E 2014 (arxiv:1410.0759)
- [39] Abbott B P et al 2019 *Phys. Rev. X* **9** 031040
- [40] Abbott R et al 2019 arxiv:1912.11716
- [41] Cuoco E, Calamai G, Fabbri L, Losurdo G, Mazzoni M, Stanga R and Vetrano F 2001 *Class. Quantum Grav.* **18** 1727–51
- [42] Nitz A et al 2020 gwastro/pycbc: Pycbc release v1.16.10 (available at: <https://doi.org/10.5281/zenodo.4063644>)
- [43] Hannam M, Schmidt P, Bohé A, Haegel L, Husa S, Ohme F, Pratten G and Pürrer M 2014 *Phys. Rev. Lett.* **113** 151101
- [44] Salpeter E E 1955 *Astrophys. J.* **121** 161
- [45] The Virgo Collaboration 2009 *Advanced Virgo Baseline Design VIR-027A-09* (available at: https://tds.virgo-gw.eu/?call_file=VIR-0027A-09.pdf)
- [46] Virgo Collaboration 2020 Virgo interferometer monitoring webpage (available at: <https://vim-online.virgo-gw.eu>)
- [47] LIGO Scientific Collaboration 2020 Updated advanced LIGO sensitivity design curve (available at: <https://dcc.ligo.org/LIGO-T1800044/public>)
- [48] Abbott B P et al 2017 *Astrophys. J.* **851** L35
- [49] Abbott B P et al (LIGO Scientific Collaboration and Virgo Collaboration) 2017 *Phys. Rev. Lett.* **119** 141101

Part III

Trouble Is My Business (Classification of difficult signals)

Chapter 3

Paper II: Core-Collapse supernova gravitational-wave search and deep learning classification

Core-Collapse supernova gravitational-wave search and deep learning classification

**OPEN ACCESS**

RECEIVED

3 January 2020

REVISED

7 February 2020

ACCEPTED FOR PUBLICATION

4 March 2020

PUBLISHED

27 May 2020

Alberto Iess^{1,2} , Elena Cuoco^{3,4} , Filip Morawski⁵ and Jade Powell⁶ ¹ Università di Roma Tor Vergata, I-00133 Roma, Italy² INFN, Sezione di Roma Tor Vergata, I-00133 Roma, Italy³ European Gravitational Observatory (EGO), I-56021 Cascina, Pisa, Italy⁴ Scuola Normale Superiore, Piazza dei Cavalieri 7, I-56126 Pisa, Italy⁵ Nicolaus Copernicus Astronomical Center, Polish Academy of Sciences, Bartycka 18, 00-716, Warsaw, Poland⁶ OzGrav, Centre for Astrophysics and Supercomputing, Swinburne University of Technology, Hawthorn, VIC 3122, Australia.E-mail: alberto.iess@roma2.infn.it

Original Content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](#).

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

**Abstract**

We describe a search and classification procedure for gravitational waves emitted by core-collapse supernova (CCSN) explosions, using a convolutional neural network (CNN) combined with an event trigger generator known as a Wavelet Detection Filter (WDF). We employ both a 1D CNN classification using time series gravitational-wave data as input, and a 2D CNN classification with time-frequency representation of the data as input. To test the accuracies of our 1D and 2D CNN classification, we add CCSN waveforms from the most recent hydrodynamical simulations of neutrino-driven core-collapse to simulated Gaussian colored noise with the Virgo interferometer and the planned Einstein Telescope sensitivity curve. We find classification accuracies, for a single detector, of over $\sim 95\%$ for both 1D and 2D CNN pipelines. For the first time in machine learning CCSN studies, we add short duration detector noise transients to our data to test the robustness of our method against false alarms created by detector noise artifacts. Further to this, we show that the CNN can distinguish between different types of CCSN waveform models.

1. Introduction

Following upgrades in recent years, the sensitivity of the Advanced LIGO [1, 2] and Advanced Virgo [3] gravitational-wave (GW) detectors has greatly increased. This led to the first direct observation of GWs emitted from a binary black hole merger in 2015 [4] and the first multimessenger observations of a binary neutron star merger [5]. Compact binaries are the most common source for ground based GW detectors [6] due to their high GW amplitudes in the frequency range of 10–1000 Hz, however as the observing time and sensitivity increases, other lower amplitude sources of GWs may be detected. Core-collapse supernovae (CCSNe) are a promising potential future GW source, which have yet to be observed [7, 8]. Their rates for our Galaxy are estimated to be ~ 1 per 100 years [9–11].

Searches for compact binary GW signals use a technique known as matched filtering [12]. This involves having a large template bank of waveforms that represent the entire parameter space. Producing such a template bank for CCSN signals is currently not possible as the time series waveforms for CCSNe are stochastic, and there are only ~ 30 CCSNe GW signals currently available from the most state of the art 3D supernova simulations, which do not cover the entire CCSNe parameter space [13–16]. Therefore, current searches for CCSN GW signals employ a coherent multi-detector network time-frequency analysis in the wavelet domain with minimal assumptions about the signal morphology [17–19].

Although the time series of CCSN waveforms are stochastic, common features have been found in time-frequency representations of CCSN GW signals. This includes low frequency (below ~ 200 Hz) GW emission due to oscillations of the shock wave known as the standing accretion shock instability (SASI) [20–22], and g-modes that occur at higher frequencies and increase in frequency over time in the spectrograms [13–16, 23, 24]. Incorporating some of our knowledge of the time frequency shape of CCSNe into our search methods may increase the sensitivity of our searches in comparison to search methods that make minimal assumptions on signal morphology.

One way to achieve this may be through the application of machine learning techniques. Machine learning techniques have already been developed for compact binary searches [25–27]. These searches use a template bank of waveforms for training similar to those used in a matched filter approach. Other studies have applied machine learning techniques to the reduction of short duration transient detector noise artifacts, known as *glitches* [28–30]. For a machine learning CCSN search, a different approach to prepare a training set is needed. Astone *et al* [31] apply a convolutional neural network (CNN) to search for CCSNe in multiple GW detectors. They use for their training set a phenomenological model that represents g-mode emission in CCSN GW signals. However, they only test their search on one of their phenomenological waveforms and not one of the actual waveforms from 3D hydrodynamical CCSN simulations. Cavaglia *et al* [32] perform a single interferometer search for CCSNe by applying machine learning techniques to reduce the single detector noise background. Chan *et al* [33] in parallel and independent to this work, apply a CNN directly to time series data to search for CCSNe. However, Chan *et al* use only the stochastic time series, and therefore do not benefit from the common features expected in the CCSN signals frequency content. They also do not account for the short duration detector noise transients, known as *glitches*, that frequently occur in GW detector data. Cavaglia *et al* [32] perform their classification task with a genetic programming algorithm using a small set of computed parameters as input features, including burst duration, central frequency and bandwidth.

Therefore, in this study we aim to determine the full potential of machine learning for CCSN searches by including spectrograms of CCSNe models, to take full advantage of their common features in frequency space, and to also account for detector glitches. We probe current and future interferometric detectors capabilities of observing GW signals from CCSNe using a combination of an event trigger generator called Wavelet Detection Filter (WDF) [34] and a 1D or 2D CNN [35, 36]. CNNs are a class of deep neural networks characterised by layers in which the output is computed as a correlation between a filter and the input data to the layer. CNNs have achieved significant performances in image classification tasks and are widely used in the field of computer vision. An in depth description of the subject can be found in Goodfellow *et al* [37]. We apply our method to simulated data for the Advanced Virgo detector, as an example of a current ground based detector, and also the future GW detector Einstein Telescope (ET) [38], which has better sensitivity in the CCSN frequency range. To test the robustness of our method, we add glitches to our simulated data and test our algorithm on CCSN waveforms that are not included in the training set.

The paper is structured as follows: section 2 describes the CCSN waveforms used in this study. Section 3, outlines the method for producing the simulated GW detector noise and glitches. Section 4 gives an overview of the WDF trigger generator. Section 5 describes the machine learning algorithms applied in this study. The results are shown in section 6, and a discussion and conclusion is given in section 7.

2. Supernova waveforms

In this section, we give a brief description of the CCSN waveforms used in this study. We use the most up to date waveforms available from 3D hydrodynamical simulations of neutrino-driven explosions. Their time series waveforms are shown in figure 1, while figure 2 shows examples of their time frequency features.

The neutrino-driven explosion mechanism is thought to occur in most CCSNe (see [39] for a review). Stars with zero age main sequence (ZAMS) masses above $\sim 8M_{\odot}$ form electron-degenerate iron cores. When the cores reach their effective Chandrasekhar masses [40], they become gravitationally unstable and collapse continues until the core reaches nuclear densities. At this stage the core rebounds and a shock wave is launched outwards. The shock wave loses energy as it moves outwards and begins to stall. It must gain more energy for the shock to be revived and power the explosion. In the neutrino-driven mechanism, the energy required to power the explosion comes from a reabsorption of some of the neutrinos.

2.1. Model s18

Model s18 is the $18M_{\odot}$ ZAMS progenitor from Powell & Müller [13] simulated with the general relativistic neutrino hydrodynamics code CoCoNuT-FMT [41]. The simulation end time is 0.9 s, at which time the GW emission has reached very low amplitudes. This model shows a clear g-mode signal in the spectrogram. This model explodes at ~ 300 ms after core bounce. The GW frequency peaks at ~ 850 Hz.

2.2. Model he3.5

Model he3.5 is the $3.5M_{\odot}$ ultra-stripped helium star from Powell & Müller [13] simulated with the general relativistic neutrino hydrodynamics code CoCoNuT-FMT. An ultra-stripped star is a star in a binary system that has been stripped of its outer layers due to mass transfer to the binary companion star [42]. The simulation ends at 0.7 s after core-bounce time, well after the peak GW emission phase. This model shows a

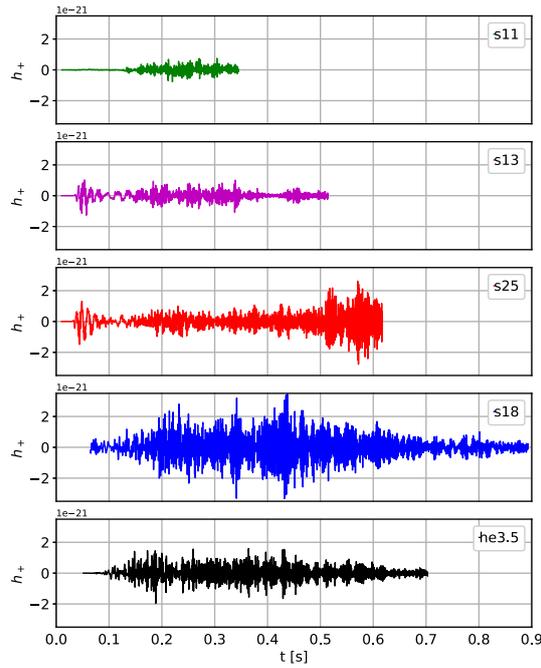


Figure 1. The h_+ gravitational-wave signals for all the CCSN models used in this analysis at a distance of 1 kpc, as computed starting from the time of core bounce $t = 0$ s.

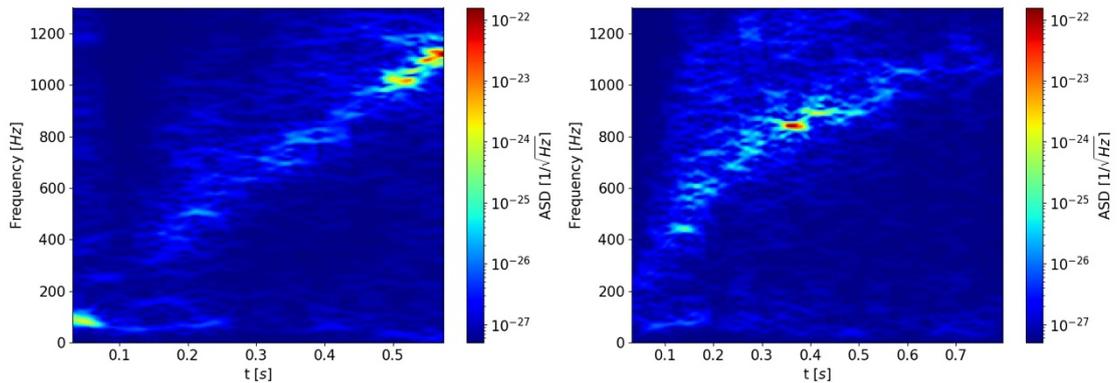


Figure 2. Time-frequency representation of the h_+ polarization of model s25 (*left*) and model s18 (*right*) for a source at a distance of 1 kpc. Both models show high frequency g-modes that peak at different frequencies. Model s25 also has a strong low frequency SASI mode.

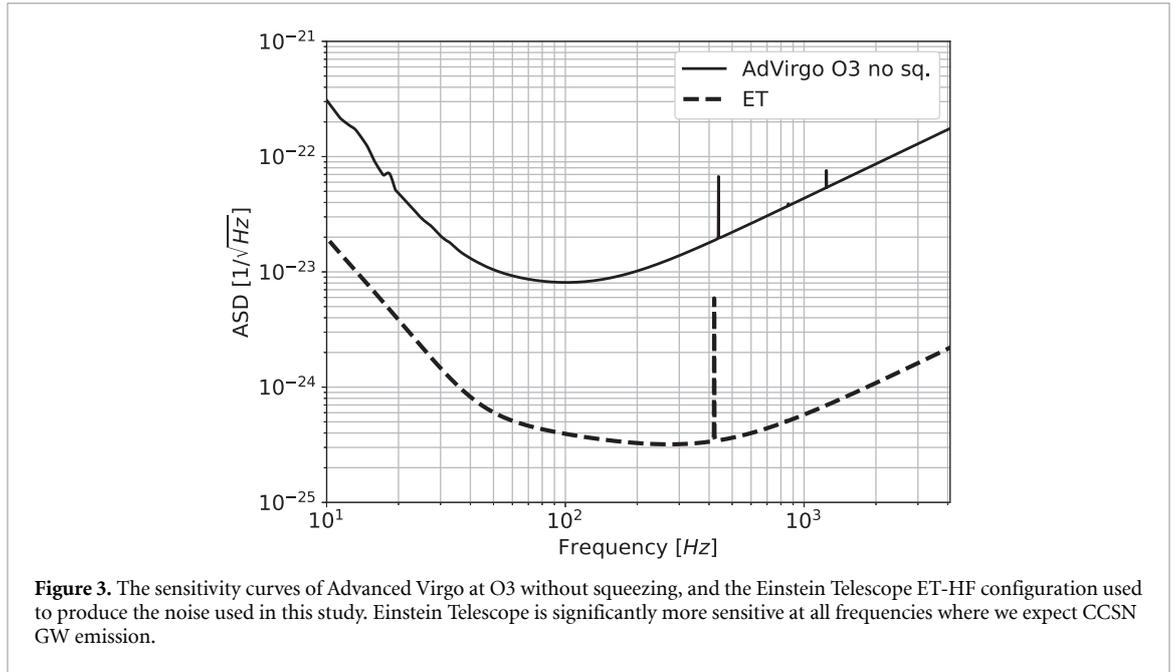
clear g-mode in the spectrogram. The amplitude of the GW signal is strongest at ~ 900 Hz. This model explodes at ~ 0.4 s after core bounce.

2.3. Model s25

Model s25 is a $25 M_{\odot}$ ZAMS model simulated by Radice *et al* [14] using the Eulerian radiation-hydrodynamics code FORNAX [43]. This model explodes very late at 0.5 s after core bounce. The simulation ends at 0.62 s after core bounce time when the GW emission is still high. This progenitor shows a clear signature of the SASI at low-frequency as well as high frequency g-modes. Its GW emission peaks at ~ 1000 Hz.

2.4. Model s13

Model s13 is a $13 M_{\odot}$ ZAMS model simulated by Radice *et al* [14] using the Eulerian radiation-hydrodynamics code FORNAX. This model does not explode, and shows GW emission associated with g-modes. This model ends at 0.78 s after core bounce. Due to the lack of shock revival, this model has lower GW amplitude and peaks at a frequency of ~ 800 Hz. This model shows SASI activity at last times, but this does not show as a strong feature in the GW emission.



2.5. Model s11

Model s11 is the $11 M_{\odot}$ ZAMS model simulated by Andresen *et al* [23] with the PROMETHEUS-VERTEX code [44], which employs Newtonian gravity. The simulation ends 0.35 s after the core bounce time. This model does not explode. However, it still has a large shock radius which prevents growth of the SASI in this model. This model has the smallest GW amplitude of all the GW signals considered in this study. Its peak frequency occurs at ~ 600 Hz.

3. Data sets

To test our method, we build different data sets using independent realisations of simulated Gaussian noise for Virgo O3 (VO3) without squeezing [45, 46] and the Einstein Telescope (ET) tuned for high frequencies ET-HF from the ET-D configuration described in [38], as examples of current and future GW detectors. The sensitivity curves are shown in figure 3. While ET is designed to be built in a triangle-shaped xylophone configuration, the sensitivities reported in [38] refer to a single pair of low (ET-LF) and high-frequency (ET-HF) interferometers of 10 km arm length and an opening angle of 90° . Compared to VO3, ET-HF has improved sensitivity at all frequencies, but especially at the higher frequencies where the peak amplitude of most CCSN models occurs.

As well as the CCSN waveforms described in section 2, in order to test the robustness of our search method we also add to our data ad hoc waveforms representative of noise transients observed in real detector data, glitches, characterised in [47, 48]. These include sine Gaussians and waveforms that are a good representation of scattered light glitches, which are a common problem in GW detectors. We generated 11 000 glitches in each of the two classes. The simulated glitches are produced in the same method as [49], and are given by,

$$\begin{aligned} h_{SG}(t) &= h_0 \sin(2\pi f_0(t - t_0)) e^{-\frac{(t-t_0)^2}{2\tau^2}} \\ h_{SL}(t) &= h_0 \sin(\phi_{SL}) e^{-\frac{(t-t_0)^2}{2\tau^2}}, \end{aligned} \quad (1)$$

where $\phi_{SL} = 2\pi f_0(t - t_0)[1 - K(t - t_0)^2]$, and τ is the time parameter which describes signal width. The parameter K defines the curvature of the scattered light arches in the time-frequency domain. We used values for τ and K such that the glitches are contained in time windows comparable to those of the CCSN signals. This is compatible with the values in [49] based on comparison with real detector glitches. We define the quality factor of sine Gaussians as $Q = 2\pi f_0\tau$ and its values are determined by the central glitch frequency f_0 and the exponential decay time constant τ . In interferometric data, glitches which can be modelled by sine Gaussians generally have $Q < 100$. We also included glitches with $Q > 100$ so that they will have longer durations similar to the CCSNe models. The central frequency range varies from 30 to 1000 Hz. The second equation in equation (1) is representative of scattered light glitches and their harmonics at lower frequencies

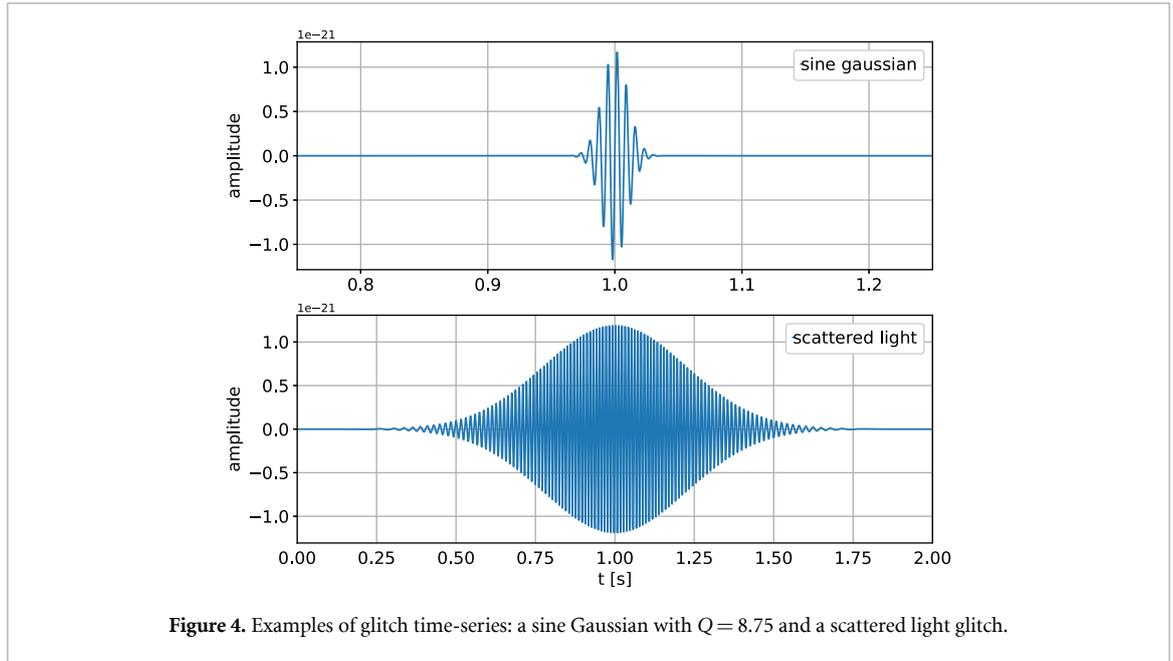


Figure 4. Examples of glitch time-series: a sine Gaussian with $Q = 8.75$ and a scattered light glitch.

and values of $K \sim 0.5$. The equation for h_{SL} can also describe a type of glitch caused by radio frequency beat notes, known as whistle glitches, which occur at higher frequencies and K . We picked the amplitude value h_0 in terms of background noise standard deviations from a log-normal distribution with mean and sigma for the underlying normal distribution of $\mu_g = -1$ and $\sigma_g = 2$. This choice results in h_0 typically below one noise standard deviation and signal-to-noise ratios (SNRs) between 1 and 1000 for both kinds of glitches. However, the peak of the SNR distribution for the scattered light glitches falls at higher values of ~ 50 than the peak for the sine Gaussian glitches at ~ 30 .

We define the square of the injection SNR in the frequency domain as in [50, 51]

$$\left(\frac{S}{N}\right)^2 = 4 \int_0^{f_{\max}} \frac{\tilde{h}(f)\tilde{h}(f)^*}{S_n(f)} df, \quad (2)$$

where $\tilde{h}(f)$ is the Fourier transform of the injected waveform time-series $h(t)$, the asterisk denotes the complex conjugate and $S_n(f)$ is the one-sided noise power spectral density.

For each interferometer, we produced a total of 5 hours of detector noise over which we injected signals at a rate of 1000 per hour for each model, including the two noise glitch models. The simulated CCSN sources follow uniform distributions in sky angles and the distances r are log-normally distributed in order to cover different order of magnitudes and SNRs. Detector sensitivity curves influence our choice for the distance distributions used in the two datasets: for VO3 distances range from 0.01 kpc to 10 kpc with average $\bar{r} \approx 0.6$ kpc, while for ET the distances range from 0.1 kpc to 1000 kpc with $\bar{r} \approx 70$ kpc. The angles are taken in the coordinate system described by the interferometer, so that the signal waveform is given by the weighted sum of the two GW polarizations h_+ and h_\times as

$$h(t) = F_+ h_+(t) + F_\times h_\times(t), \quad (3)$$

where F_+ and F_\times are the antenna pattern functions given by

$$\begin{aligned} F_+ &= \frac{1}{2}(1 + \cos^2 \theta) \cos 2\phi \cos 2\psi - \cos \theta \sin 2\phi \sin 2\psi \\ F_\times &= \frac{1}{2}(1 + \cos^2 \theta) \cos 2\phi \sin 2\psi + \cos \theta \sin 2\phi \cos 2\psi, \end{aligned} \quad (4)$$

where ϕ, θ are the angles to the source measured from one of the interferometer arms and from the normal to the detector plane, respectively, and ψ is the polarization angle [52]. The use of the ET-HF noise curve allows us to adopt the single interferometer antenna pattern as an approximation also for ET. We will drop the subscript and refer to ET-HF as ET in the rest of the text. The waveforms are resampled to 4096 Hz, as most of their GW emission is below 1000 Hz.

4. Wavelet detection filter

Trigger event generation is provided by Wavelet Detection Filter (WDF) [34], which has been previously used in the context of glitch classification problems as described in Powell *et al* [53]. The detection algorithm is based on decomposition of the data into multiple time-frequency resolution maps, through the wavelet transform. A generic time series $s(t)$ is projected onto a family of mutually orthonormal wavelets as

$$\langle s | \psi_{a,b} \rangle = \int_{-\infty}^{+\infty} s(t) \frac{1}{\sqrt{b}} \psi^* \left(\frac{t-a}{b} \right) dt, \quad (5)$$

where ψ^* is the complex conjugate of the mother wavelet, parameter b sets the scale of the time-frequency map and a is the time-shifting parameter. For analysis of transients WDF implements the discrete wavelet transform using a bank or wavelet family including Daubechies, Haar and spline wavelets [54–56].

In this study, we whiten the data in the time domain as described in Cuoco *et al* [57]. This is achieved by fitting the noise power spectral density (PSD) using an Auto-Regressive (AR) model as described in [57]. In this study, the initial 300 seconds of each dataset are used to fit the noise PSD with an AR model of order 4000. After whitening, the wavelet coefficients are expected to contain features of the transient waveforms on different time and frequency scales. In particular, only relevant coefficients are retained using the Donoho and Johnstone method [58] which sets a lower threshold t on the absolute value of the wavelet coefficients $|w_i|$ as

$$t = \sqrt{2 \log N} \hat{\sigma}, \quad (6)$$

where N is the number of window data points and $\hat{\sigma}$ the estimation of the noise standard deviation. WDF describes the event triggers by means of a set of parameters: the trigger time (timestamp), the mean and maximum frequency content of the event, the wavelet coefficients containing the event information and a value proportional to the WDF SNR_w, not to be confused with the injection SNR, given by

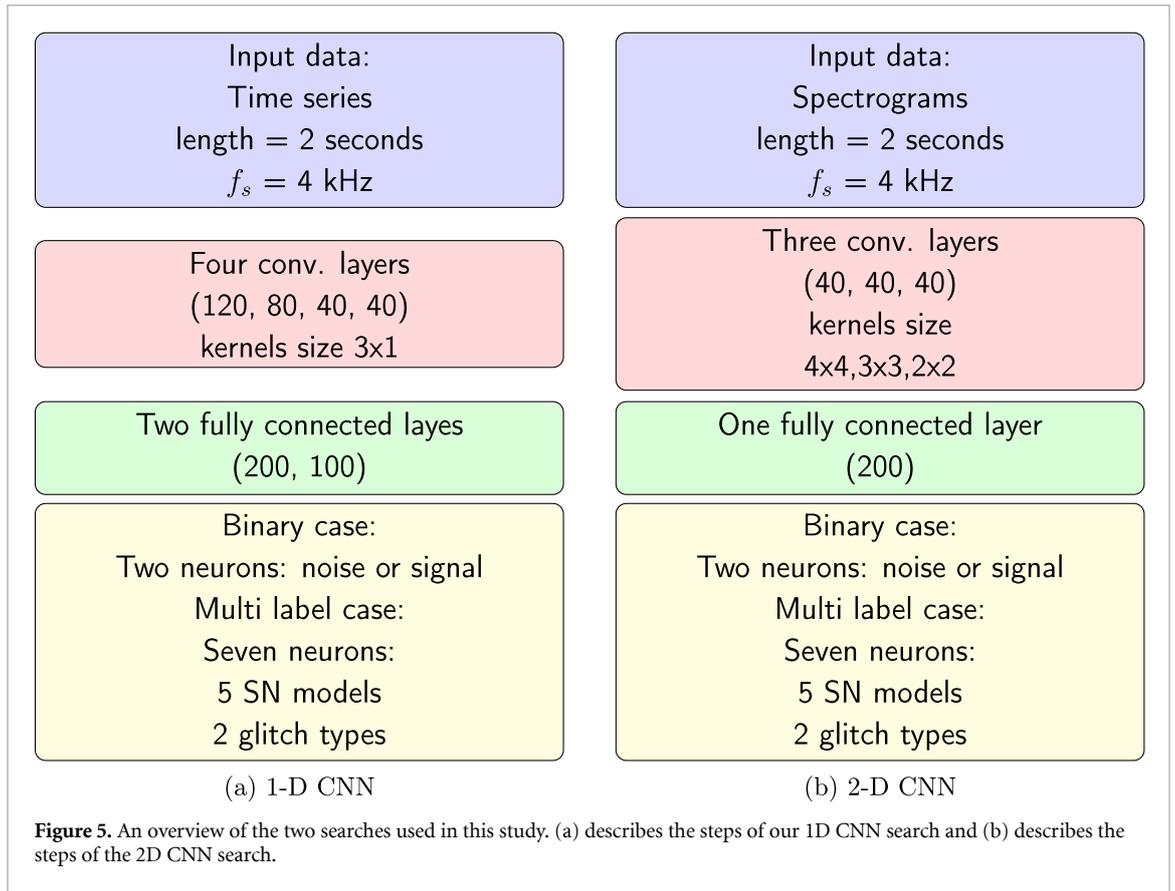
$$\text{SNR}_w = \frac{\sum_i w_i^2}{\hat{\sigma}}, \quad (7)$$

which is the indicator for the threshold of the WDF detected events. The WDF window size used in this analysis is 1024 points with an overlap of 256 between consecutive windows, which corresponds to a time-window of 0.25 s and 0.062 5 s, respectively, for a sampling frequency of 4096 Hz.

After WDF finds all the triggers in the data, the time-domain whitened strain data around the triggers are then fed to the CNN classifier. The advantages of combining a trigger generator with the CNN are that the trigger generator saves time by not needing to apply the CNN to all data, and as the CNN only produces a classification of events, information provided by the event trigger generator, such as the SNR, is required to produce an estimate of the significance of a real detection.

5. Deep learning pipelines

Each candidate event found by WDF [34] has a timestamp which is used to build the input sample data for the CNN [35, 36]. To mimic the real case scenario, among all trigger times we choose the one corresponding to the highest SNR_w value inside a time window coincident with the injection. We define the coincident window so that the WDF window overlaps the true signal by at least 3/4 of its length (i.e. 768 points). Note that we are assuming that the trigger timestamp produced by WDF occurs approximately at half of the injected signal duration, so that we can take a symmetric 1 s time window around it to cover the entire waveform. To take into account the fact that the peak SNR_w may not occur at such exact time, a random shift in time can be manually applied, although this has not been done to achieve the results presented in this analysis. Given the timestamp, the dataset can be tailored for the two cases of 1D and 2D convolutional neural networks. The unique antenna pattern weights and distance scaling associated with each signal, combined with the variability of the Gaussian noise background, ensure that all samples are different among each other. The signals recovered by WDF for all CCSNe and glitch models are then merged and shuffled to form the full dataset. The number of signals in each class depends on the characteristics of the model: for instance, the lowest emission model s11 contains roughly half of the signals compared to all the other models, because more s11 signals are at lower SNRs and are missed by WDF. The full datasets for the two detectors are divided in the following way: 60% for model training, 10% for model validation and 30% for testing. An exception to this is presented in subsection 6.1.2, where we use all the signals of a particular model as test set, while we keep training on 60% of the remaining waveforms and validating on 10%. The validation set was used for hyperparameter tuning and to stop training before overfitting.



5.1. CNN 1D

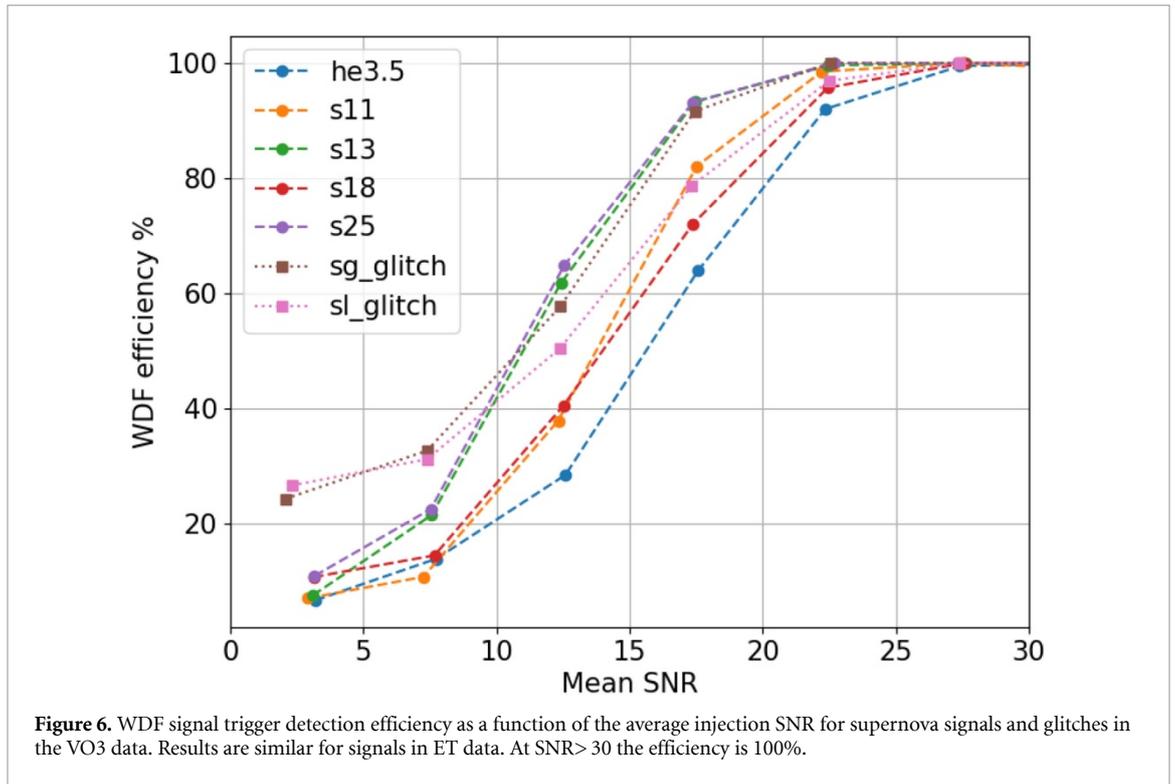
The input data consist of 1 s of whitened time-series sampled at 4096 Hz, symmetrically taken around the event trigger times. Longer segments lead to a decrease in accuracy, since we are inserting more noise background before and after the signals. The network is structured in four layers as showed in figure 5(a): each of them contains a convolutional layer with, respectively, 120, 80, 40 and 40 filters with 3x1 kernels and a rectified linear unit activation function (we recall that $\text{ReLU}(x) = \max(0, x)$). A 2x2 max pooling is then applied to reduce the input data to the next layer. Spatial dropout involving 40% of the layer's weights is applied as a regularization procedure. Two fully connected (FC) layers with 200 and 100 neurons follows. The final layer depends on the task given to the CNN. In the case of binary classification between signal and noise instances (or two different kinds of signal) there will be a 2-neuron FC layer with softmax activation function and the associated loss function will be the binary cross-entropy. In the case of multilabel classification this will be substituted by a n_l -neuron FC layer, where n_l is the number of distinct labels, and a categorical cross-entropy loss function. The optimizer used is Adam [59], with a learning rate $\alpha = 0.001$ and a batch size of 128 samples per training step. The number of epochs is never taken greater than 50, since at larger values the model starts to overfit.

5.2. CNN 2D

In the 2D case, inputs are spectrogram images built using a total of 2 s whitened data around the trigger times, although the window is tunable for shorter signals. By using spectrograms, information on the signal phase, which is dependent on the stochastic realization provided by simulations, is lost and replaced by the added value of the frequency morphology of the waveform. The architecture is similar to the 1D case, but with some relevant differences as sketched in figure 5(b), starting from three layers instead of four. The number of convolutional filters in each layer is 40. Moreover, the kernels are taken to have different sizes (4x4, 3x3, 2x2) and no spatial dropout is involved. Finally the batch size is of 64 samples while the Adam learning rate is $\alpha = 0.001$, with an associated learning decay rate of 0.066 667.

6. Search and classification results

First we apply WDF to the data to find the trigger times at which the signals and glitches occur. Figure 6 shows the WDF detection efficiency as a function of the average injection SNR for the VO3 data, computed



on segments each containing 3000 signals of a specific class. All signals with SNR > 30 are detected by WDF. We recall that the SNRs of the injected signals depend on distance, antenna pattern and model GW amplitude. We use the same distance range for all CCSN models over the same detector background, while there is a lot of variation in the models amplitudes (as previously shown in figure 1). At fixed distances the low energy, non exploding models, such as model s11, have lower SNRs and therefore the SNR distributions are not the same for each models training and testing set. Models s13 and s25 have the highest WDF detection efficiency, which is due to their models having higher SNR waveforms in the training data. Model s18 is also a high amplitude model, but its detection efficiency may be worse than the other high amplitude models due to its longer duration. We see similar WDF results for the signals and glitches in ET noise.

6.1. Supernova vs glitch classification

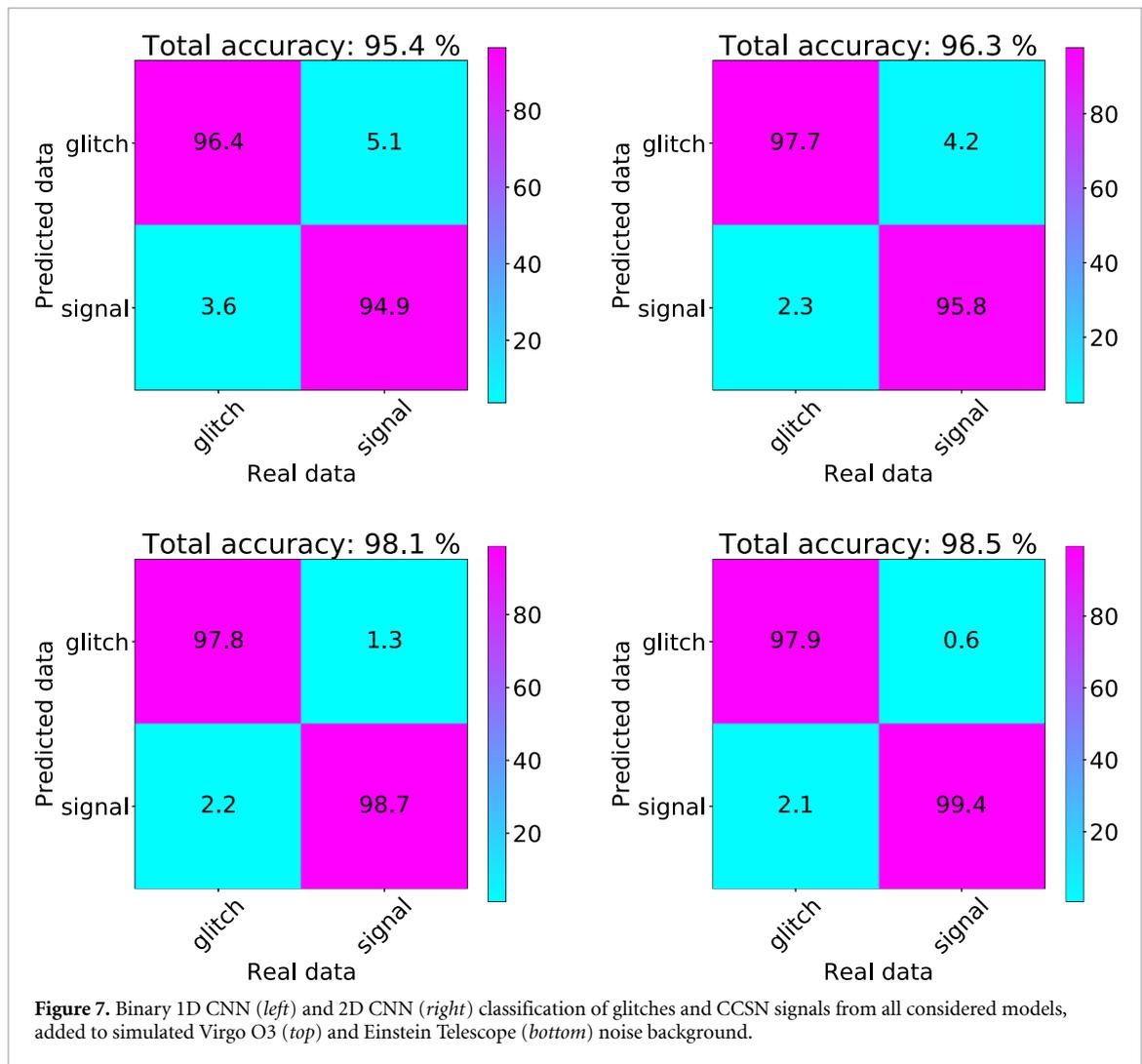
In this section, we perform a binary classification into two categories: signals and glitches. The signal class should include all CCSN models, and the same holds for the glitch class. In a real case scenario, training would be carried out on all available realistic CCSN waveforms. The true positive rate (TPR) for each class are found on the diagonal of the confusion matrix and are defined in the test set as the ratio between the correctly classified samples, true positives, and the total number of samples in a class computed as the sum of true positives (TP) and false negatives (FN) as

$$TPR = \frac{TP}{TP + FN}. \quad (8)$$

The CNN was trained on a Tesla k40 GPU. The 2D CNN takes longer to train than the 1D CNN, because for each training batch spectrograms have to be computed from the whitened time series. For binary classification, the training phase is of the order of 20 to 30 minutes depending on the choice between the 1D and 2D CNN.

6.1.1. Models included in training

First we test our method using the same waveform models in the test sets as we used in the training step. Therefore, the CNNs are trained on samples from all the glitch and CCSNe models and the trained networks are then used to classify new instances of the same models, characterized by different background noise, antenna pattern weights or glitch parameters. This is not a very realistic case, as we know that a real CCSN detection is unlikely to look exactly like one of our training waveform models. However, it does allow us to determine the maximum accuracy of our method, and we can compare that number to the case where different waveforms are used for training and testing to quantify how much accuracy is lost due to a lack of CCSNe waveforms.



The confusion matrices for this case, are presented in figure 7. We find very high accuracies (above $\sim 95\%$) for both the 1D and 2D CNN results. The results improve slightly for the 2D case as expected due to the common features in the CCSN spectrograms.

The misclassified waveforms correspond to low SNRs, indistinctive of the CCSN model. For the ET dataset, results are better both in the 1D the 2D case with accuracies in the signal class $\sim 99\%$, with $\sim 98\%$ of the total samples correctly classified when including glitches, as showed in figure 7 (*bottom*). The improved results for ET are due to the better high frequency sensitivity of the detector.

6.1.2. Models not included in training

As a real CCSN signal would not match exactly the waveforms we use for training, we test the ability of the CNN to generalize to different CCSN models by training the network on a subset of four signal models and then testing it on the one that was not included in the training step. We keep training on 60% and validating on 10% of the full shuffled dataset, but after removing all signals corresponding to the CCSNe test model. To avoid testing on a signal sample which is too small, we use 100% of the signals detected by WDF of that particular model in the test set, coupled with the usual 30% of the glitch classes.

The sensitivities achieved for the signal classes are reported in table 1 for the two detectors, VO3 and ET, and the two CNN implementations, along with the total accuracies. The *Total* accuracy is defined as the overall number of correctly classified samples in either the signal or noise class, over the total number of samples. The results are similar for both the 1D and 2D CNNs with total accuracies consistently above 90%. The accuracy is not as high for model s25 as it is for the other models. We suspect that this is due to model s25 being the only model that we consider that contains a strong low frequency SASI mode. Model s25 also has a higher peak GW frequency than the other models used for training. Therefore, we could improve on this result in the future by including more CCSN waveforms with SASI features in the training set. The

Table 1. Binary classification results on CCSN models not included in training. For the 1D and 2D CNN implementations and both detector datasets, we show the sensitivity for the *Signal* class. We also report the *Total* accuracy computed as the ratio of the number of correctly classified samples in one of the two classes and the total number of test samples. The waveform used in the test set for the signal class is shown in the first column.

VO3 Test set	1D CNN accuracy		2D CNN accuracy	
	<i>Signal</i>	<i>Total</i>	<i>Signal</i>	<i>Total</i>
s11	93.9	93.7	98.0	94.3
he3.5	96.2	95.5	95.2	97.6
s18	97.5	96.7	98.4	97.9
s13	94.5	94.4	94.4	96.9
s25	95.1	95.1	92.2	95.9

ET Test set	1D CNN accuracy		2D CNN accuracy	
	<i>Signal</i>	<i>Total</i>	<i>Signal</i>	<i>Total</i>
s11	94.5	96.7	95.5	97.2
he3.5	98.0	97.8	98.5	97.6
s18	92.1	94.2	92.4	96.2
s13	95.9	96.6	84.5	94.1
s25	73.3	83.2	89.6	95.5

results show that we do not lose a lot of accuracy when we perform a more realistic test with models not included in the training set.

6.2. Multi-model classification

In this section, we aim to distinguish between different CCSN waveform models. Distinguishing between different types of CCSN waveforms can allow us to determine properties of a CCSN GW source, such as the explosion mechanism, or the presence of features like the low frequency SASI [60, 61]. The models we use in this section all have the same neutrino-driven explosion mechanism, and are used just to demonstrate the method rather than to make any astrophysical statement by distinguishing between them with the CNN. In the future, we could apply the same technique to models with different explosion mechanisms when more modern 3D CCSN waveforms, for example for the magnetorotational explosion mechanism, become available.

CCSN waveform end times are usually determined by a lack of available computer time rather than some astrophysical reason. Therefore, when we attempt to distinguish between them we do not want the CNN to include information about the different lengths of the signals, as that information is not astrophysical. To avoid this issue, in the 2D CNN case we adjusted the window size to match the length of the shortest waveform considered in this analysis, s11, with duration 0.35 s. As mentioned in section 5.1, we made the assumption that the WDF trigger timestamp falls at half of the injected signal length, in order to take a symmetric window in that time interval. Therefore, by choosing the s11 window, we only keep the central part of the longer duration signal models such as s18.

The morphology of the time series waveforms are stochastic, and therefore cannot be related back to individual types of models. However, it is expected that their time series amplitudes may be related to the CCSN properties. In this section, to avoid problems related to the stochastic nature of the waveforms, we decide to ensemble [62] the results for both the 1D and 2D CNN to produce the multi-label classification results.

The confusion matrices for the ensemble multilabel classification are shown in figure 8. The sensitivities for each class are found on the diagonal. While total accuracies decrease compared to the binary problem, they still reach 87.9% for VO3 and 89.6% for ET. In both cases, higher emission models are classified more accurately. Sine Gaussians with longer duration τ and shorter scattered light glitches are responsible for a large part of the misclassified samples among the two noise classes. The trained model is able to distinguish among the different CCSN models even when using a short time window to build the sample spectrogram and time-series.

In figure 9, we show the combined VO3 and ET SNR distribution of the CCSNe signals in the test set up to an SNR value of 100. As expected misclassified signals are distributed at lower SNRs, while the correctly classified signals cover the full range of the initial test set distribution. Both distributions decrease at SNRs lower than 10, since WDF picks very few signals in that region. Overall, multilabel classification requires training times of ~ 1 hour to achieve the reported accuracies.

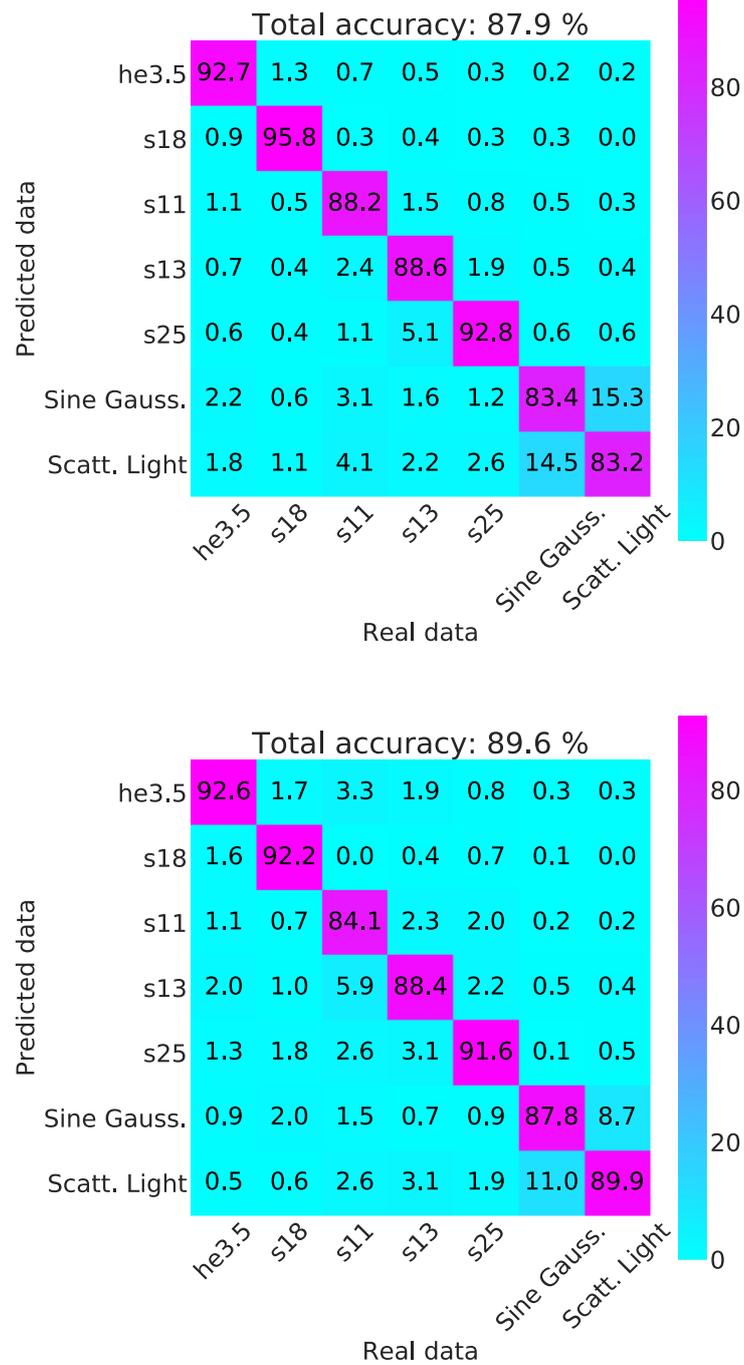
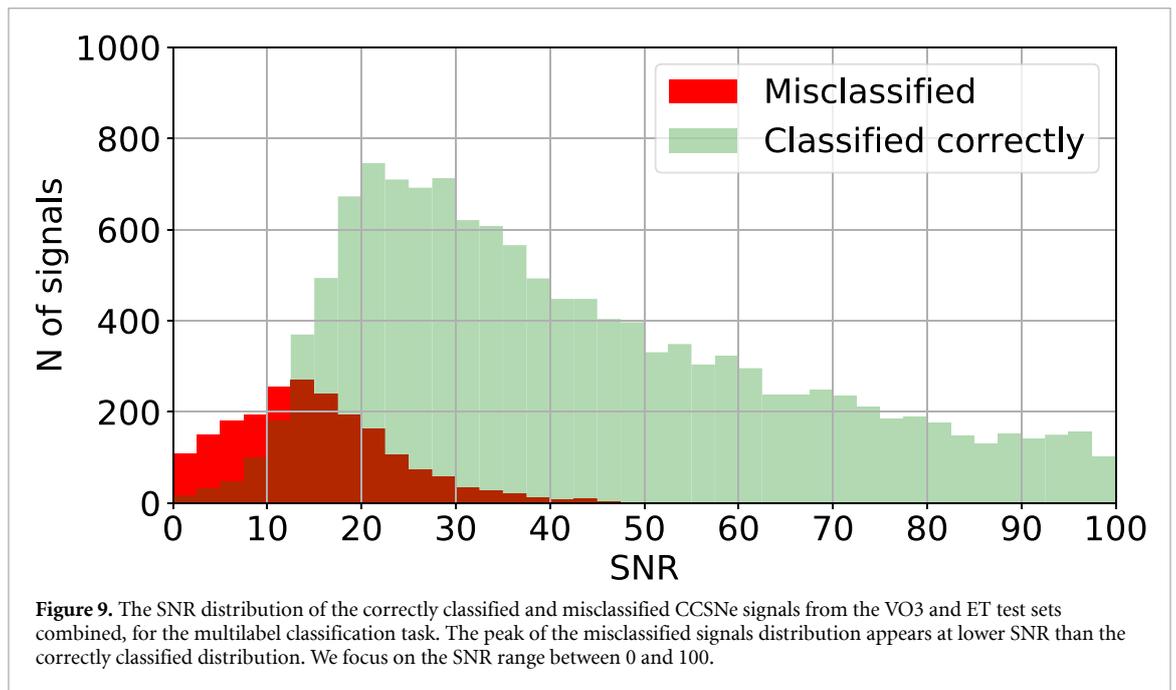


Figure 8. Confusion matrices for multilabel classification for the VO3 (*top*) and ET (*bottom*) datasets with all the CCSNe and the two glitch models.

7. Discussion

Current searches for GWs from CCSNe make no assumptions on the signal morphology. The time series waveforms for CCSN GW signals are stochastic, and so they can never be used in a matched filter search. However, the frequency content of CCSN signal predictions is not stochastic and can be directly related to the properties of the explosion and the proto neutron star. A matched filter search in the frequency domain still cannot be performed for CCSN signals because there are not enough 3D waveforms currently available, and they do not fully cover the CCSN parameter space. Further to this, many simulations are ended before the peak GW emission time or have missing input physics. However, in the waveforms we do have available, we see similar features in their spectrograms that we would expect to see in a real GW detection, and incorporating this information into our GW searches may increase our sensitivity to these sources.



We aimed to achieve this by performing a new search using a combination of a trigger generator called WDF and a CNN. A trigger generator finds all excess power events above a threshold, and provides some information about each event. The events could be real GW signals or background detector noise glitches. Using a trigger generator allowed us to save search time because we then only feed time around the WDF triggers to the CNN. As long as a whitening procedure is implemented, it is possible to directly feed the data to a CNN without a trigger generator. However, while we do not provide a direct comparison with the computational cost of a CNN-only pipeline, we predict a larger training set would be needed to compensate for the unknown localization of the signal in time, since the actual signal could fall anywhere in the data stream. Using a trigger generator also enables us to estimate how significant each GW detection is, which cannot be done with a CNN alone. We do not include the significance of events in our study, but it will be an important step for future work using real GW detector data. Moreover, the wavelet coefficient information produced by WDF could be used to reconstruct the waveform of a detected event, and this will be included in future work.

Compared to Chan *et al* [33] and Astone *et al* [31], we performed only a single detector search, using one current and one future third generation GW detector. In the future, our search would need to be extended to a multi detector analysis to fully compare to existing pipelines. In this case, we would run WDF on data from multiple detectors and search for coincidence between the different detectors triggers before passing the data to the CNN. We included in our study 5 of the most modern 3D simulations of CCSNe that we used to train the CNN. We also trained the CNN on two types of simulated glitches to show how we can distinguish real events from transient detector noise events. In real GW detector data, there are a larger number of different glitch types, and the data needed to train machine learning models for the real glitches can be obtained from the citizen science project Gravity Spy [30]. Astone *et al* do not use simulated waveforms, but phenomenological models to train their CNN. Cavaglia *et al* include older waveforms from 2D simulations which are not believed to be state-of-the-art. Finally, Chan *et al* use a mixture of the older and newer models.

We ran both a 1D CNN on the time series data, and a 2D CNN using spectrograms of the data. The 1D method is less robust due to the stochastic nature of the waveforms. Although the time series features of the waveforms we used here cannot be related back to the source parameters, they are still very different to glitches, and we show that we can distinguish them from noise with a 95% accuracy. It is likely that the 1D search would also find GW signals that are not CCSNe, as the time series could represent any burst of GWs. However, in the 2D case, the spectrograms used for training are directly related to emission features we only expect from a CCSN signal, and should be able to distinguish between a CCSN and some other kind of GW burst. The common features in the spectrograms improve our detection accuracy slightly. However, we must be careful during the training to include all the possible features, as we can see that the accuracy for model s25 decreases when we do not include other signals with a strong low frequency SASI component in the training. Therefore, we see the benefit in including both a 1D and a 2D search, as the 1D search is more likely to detect the GWs if our predictions of the CCSN emission are incorrect, and if they are correct then the

features in the 2D spectrograms result in increased search accuracy. For this reason, we applied for the first time in this field the ensemble method for the 2 different classification model in the multilabel case.

We showed that the CNN can also distinguish between different CCSN waveform models. We do not have any important astrophysical differences in these models that we used here, but in future studies we could use this to distinguish between waveforms from different explosion mechanisms, or models with and without rotation.

Acknowledgment

This article/publication is based upon work from COST Action CA17137, supported by COST (European Cooperation in Science and Technology). JP is supported by the Australian Research Council (ARC) Centre of Excellence for Gravitational Wave Discovery (OzGrav), through project number CE170 100 004. FM is supported by the Polish National Science Centre Grant No. 2016/22/E/ST9/00 037. The computations were carried out on the PL-GRID Prometheus cluster infrastructure.

Data availability statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID iDs

Alberto Iess  <https://orcid.org/0000-0001-9658-6752>

Elena Cuoco  <https://orcid.org/0000-0002-6528-3449>

Filip Morawski  <https://orcid.org/0000-0002-6194-8239>

Jade Powell  <https://orcid.org/0000-0002-1357-4164>

References

- [1] LIGO Scientific Collaboration Harry G M *et al* 2010 *Class. Quant. Grav.* **27** 084006
- [2] LIGO Scientific Collaboration Aasi J *et al* 2015 *Class. Quant. Grav.* **32** 074001
- [3] Virgo Collaboration Acernese F *et al* 2015 *Class. Quant. Grav.* **32** 024001
- [4] LIGO Scientific Collaboration and Virgo Collaboration Abbott B P *et al* 2016 *Phys. Rev. Lett.* **116** 061102
- [5] LIGO Scientific Collaboration and Virgo Collaboration Abbott B P *et al* 2017 *Phys. Rev. Lett.* **119** 161101
- [6] LIGO Scientific Collaboration and Virgo Collaboration Abbott B P *et al* 2019 *Phys. Rev. X* **9** 031040
- [7] LIGO Scientific Collaboration and Virgo Collaboration Abbott B P *et al* 2016 *Phys. Rev. D* **94** 102001
- [8] LIGO Scientific Collaboration and Virgo Collaboration Abbott B P *et al* 2020 *Phys. Rev. D* **101** 084002
- [9] van den Bergh S and Tammann G A 1991 *Annu. Rev. Astron. Astrophys.* **29** 363–407
- [10] Cappellaro E, Turatto M, Benetti S, Tsvetkov D Y, Bartunov O S and Makarova I N 1993 *Astron. Astrophys.* **273** 383
- [11] Alexeyev E N and Alexeyeva L N 2002 *J. Exp. Theor. Phys.* **95** 5–10
- [12] Usman S A *et al* 2016 *textitClass. Quant. Grav.* **33** 215004
- [13] Powell J and Müller B 2019 *Mon. Not. R. Astron. Soc.* **487** 1178–90
- [14] Radice D, Morozova V, Burrows A, Vartanyan D and Nagakura H 2019 *Astrophys J. Lett.* **876** L9
- [15] Andresen H, Müller E, Janka H T, Summa A, Gill K and Zanolin M 2019 *Mon. Not. R. Astron. Soc.* **486** 2238–53
- [16] Kuroda T, Kotake K, Hayama K and Takiwaki T 2017 *Astrophys. J* **851** 62
- [17] Klimentko S *et al* 2016 *Phys. Rev. D* **93** 042004
- [18] Klimentko S *et al* 2008 *Class. Quant. Grav.* **25** 114029
- [19] Gossan S E, Sutton P, Stuver A, Zanolin M, Gill K and Ott C D 2016 *Phys. Rev. D* **93** 042002
- [20] Blondin J M, Mezzacappa A and DeMarino C 2003 *Astrophys. J* **584** 971
- [21] Blondin J M and Mezzacappa A 2006 *Astrophys. J* **642** 401–9
- [22] Foglizzo T, Galletti P, Scheck L and Janka H T 2007 *Astrophys. J* **654** 1006–21
- [23] Andresen H, Müller B, Müller E and Janka H T 2017 *Mon. Not. R. Astron. Soc.* **468** 2032–51
- [24] Yakunin K N *et al* 2017 arXiv:1701.07325
- [25] Gebhard T D, Kilbertus N, Harry I and Schölkopf B 2019 *Phys. Rev. D* **100** 063015
- [26] George D and Huerta E A 2018 *Phys. Rev. D* **97** 044039
- [27] Gabbard H, Williams M, Hayes F and Messenger C 2018 *Phys. Rev. Lett.* **120** 141103
- [28] Powell J, Torres-Forné A, Lynch R, Trifirò D, Cuoco E, Cavaglià M, Heng I S and Font J A 2017 *Class. Quant. Grav.* **34** 034002
- [29] Razzano M and Cuoco E 2018 *Class. Quant. Grav.* **35** 095016
- [30] Zevin M *et al* 2017 *Class. Quant. Grav.* **34** 064003
- [31] Astone P, Cerdá-Durán P, Di Palma I, Drago M, Muciaccia F, Palomba C and Ricci F 2018 *Phys. Rev. D* **98** 122002
- [32] Cavaglià M, Gaudio S, Hansen T, Staats K, Szczepańczyk M and Zanolin M 2020 *Mach. Learn.: Sci. Technol.* **1** 015005
- [33] Chan M L, Heng I S and Messenger C 2019 arXiv:1912.13517
- [34] Cuoco E, Razzano M and Utina A 2018 *26th European Signal Conf. (EUSIPCO), 2018* pp 2648–52
- [35] Yamashita R, Nishio M, Do R K G and Togashi K 2018 *Insights into Imaging* **9** 611–29
- [36] Russakovsky O *et al* 2015 *Int. J. Comput. Vis.* **115** 211–52
- [37] Goodfellow I, Bengio Y and Courville A 2016 *Deep Learning* (Cambridge, MA: MIT Press)

- [38] Hild S et al 2011 *Class. Quant. Grav.* **28** 094013
- [39] Janka H T 2017 Neutrino driven explosions *Handbook of Supernovae* ed A Alsabti, P Murdin, (Berlin: Springer) pp 1095
- [40] Bethe H A 1990 *Rev. Mod. Phys.* **62** 801–66
- [41] Müller B, Janka H T and Dimmelmeier H 2010 *Astrophys. J. Supp. S* **189** 104–33
- [42] Tauris T M, Langer N and Podsiadlowski P 2015 *Mon. Not. R. Astron. Soc.* **451** 2123–44
- [43] Skinner M A, Dolence J C, Burrows A, Radice D and Vartanyan D 2019 *Astrophys. J. Supp. S* **241** 7
- [44] Rampp M and Janka H T 2002 *Astron. Astrophys.* **396** 361–92
- [45] The Virgo Collaboration 2009 *Advanced Virgo Baseline Design, note VIR-027A-09* (https://tds.virgo-gw.eu/?call_file=VIR-0027A-09.pdf)
- [46] Virgo Interferometer Monitoring webpage (<https://vim-online.virgo-gw.eu>)
- [47] Ligo Scientific Collaboration Abbott B P et al 2016 *Class. Quant. Grav.* **33** 134001
- [48] Ligo Scientific Collaboration Aasi J et al 2015 *Class. Quant. Grav.* **32** 115012
- [49] Razzano M and Cuoco E 2018 *Class. Quant. Grav.* **35** 095016
- [50] Cutler C and Flanagan E E 1994 *Phys. Rev. D* **49** 2658–97
- [51] Maggiore M 2008 *Gravitational Waves* (Oxford: Oxford Univ. Press) (<https://doi.org/10.1093/acprof:oso/9780198570745.001.0001>)
- [52] Schutz B F 2011 *Class. Quant. Grav.* **28** 125023
- [53] Powell J, Trifirò D, Cuoco E, Heng I S and Cavaglia M 2015 *Class. Quant. Grav.* **32** 215012
- [54] Daubechies I 1992 *Ten lectures on wavelets* (Philadelphia, PA: Society for Industrial and Applied Mathematics) **61**
- [55] Mallat S 1999 *A Wavelet Tour of Signal Processing* (Cambridge, MA: Academic Press) (<https://dog.org/10.1016/B978-0-12-466606-1.X5000-4>)
- [56] Unser M 1997 *Proc. SPIE* **3169** 422–31
- [57] Cuoco E, Calamai G, Fabbroni L, Losurdo G, Mazzoni M, Stanga R and Vetrano F 2001 *Class. Quant. Grav.* **18** 1727
- [58] Donoho D L and Johnstone I M 1994 *Biometrika* **81** 425–55
- [59] Kingma D P and Ba J 2015 *3rd Int. Conf. on Learning Representations, ICLR 2015* arXiv:1412.6980
- [60] Powell J, Gossan S E, Logue J and Heng I S 2016 *Phys. Rev. D* **94** 123012
- [61] Roma V, Powell J, Heng I S and Frey R 2019 *Phys. Rev. D* **99** 063018
- [62] Dietterich T G 2000 Ensemble methods in machine learning *Multiple Classifier Systems* ed J Kittler, F Roli (Berlin, Heidelberg: Springer) pp 1–15

Chapter 4

Paper III: Convolutional neural network classifier for the output of the time-domain \mathcal{F} -statistic all-sky search for continuous gravitational waves



PAPER

OPEN ACCESS

RECEIVED
4 November 2019REVISED
20 March 2020ACCEPTED FOR PUBLICATION
6 April 2020PUBLISHED
22 June 2020

Original Content from
this work may be used
under the terms of the
[Creative Commons
Attribution 4.0 licence](#).

Any further distribution
of this work must
maintain attribution to
the author(s) and the title
of the work, journal
citation and DOI.



Convolutional neural network classifier for the output of the time-domain \mathcal{F} -statistic all-sky search for continuous gravitational waves

Filip Morawski¹ , Michał Bejger¹  and Paweł Ciecieląg¹ 

Nicolaus Copernicus Astronomical Center, Polish Academy of Sciences, Bartycka 18, 00-716, Warsaw, Poland

¹ Author to whom any correspondence should be addressed.E-mail: fmorawski@camk.edu.pl**Keywords:** deep learning, continuous gravitational waves, convolutional neural networks, neutron stars

Abstract

Among the astrophysical sources in the Advanced Laser Interferometer Gravitational-Wave Observatory (LIGO) and Advanced Virgo detectors' frequency band are rotating non-axisymmetric neutron stars emitting long-lasting, almost-monochromatic gravitational waves. Searches for these continuous gravitational-wave signals are usually performed in long stretches of data in a matched-filter framework e.g. the \mathcal{F} -statistic method. In an all-sky search for *a priori* unknown sources, a large number of templates are matched against the data using a pre-defined grid of variables (the gravitational-wave frequency and its derivatives, sky coordinates), subsequently producing a collection of *candidate signals*, corresponding to the grid points at which the signal reaches a pre-defined signal-to-noise threshold. An astrophysical signature of the signal is encoded in the multi-dimensional vector *distribution* of the candidate signals. In the first work of this kind, we apply a deep learning approach to classify the distributions. We consider three basic classes: Gaussian noise, astrophysical gravitational-wave signal, and a constant-frequency detector artifact ('stationary line'), the two latter injected into the Gaussian noise. 1D and 2D versions of a convolutional neural network classifier are implemented, trained and tested on a broad range of signal frequencies. We demonstrate that these implementations correctly classify the instances of data at various signal-to-noise ratios and signal frequencies, while also showing concept generalization i.e. satisfactory performance at previously unseen frequencies. In addition we discuss the deficiencies, computational requirements and possible applications of these implementations.

1. Introduction

1.1. Gravitational wave searches

Gravitational waves (GWs) are distortions of the curvature of spacetime, propagating with the speed of light [1]. Direct experimental confirmation of their existence was recently provided by the Laser Interferometer Gravitational-Wave Observatory (LIGO) and Virgo collaborations [2, 3] in the form of observations of, to date, several binary black hole mergers [4–6], and one binary neutron star (NS) merger, the latter also being electromagnetically bright [7]; the first transient GW catalog [8] contains the summary of the LIGO and Virgo O1 and O2 runs.

In addition to merging binary systems, among other promising sources of GWs are non-axisymmetric supernova explosions, as well as long-lived, almost-monochromatic GW emission by rotating, non-axisymmetric NS, sometimes called 'GW pulsars'.

In this article we will focus on the latter type of signal. The departure from axisymmetry in the mass distribution of a rotating NS can be caused by dense-matter instabilities (e.g. phase transitions, *r*-modes), strong magnetic fields and/or elastic stresses in its interior (for a review see [9, 10]). The deformation and hence the amplitude of the GW signal depend on the largely unknown dense-matter equation of state, surrounding and history of the NS; therefore the time-varying mass quadrupole required by the GW

emission is not naturally guaranteed as in the case of binary system mergers. The LIGO and Virgo collaborations performed several searches for such signals, both targeted searches for NS sources of known spin frequency parameters and sky coordinates (pulsars, [11, 12] and references therein), as well as all-sky searches for *a priori* unknown sources with unknown parameters ([13, 14] and references therein).

1.2. All-sky searches for continuous GWs

All-sky searches for continuous GWs are ‘agnostic’ in terms of GW frequency f , its time derivatives (*spindown* \dot{f} , sometimes \ddot{f} and higher), and sky position of the source (e.g. δ and α in equatorial coordinates). The search consists of sweeping the parameter space to find the best-matching template by evaluating the signal-to-noise ratio (SNR). There are various algorithms (for a recent review of the methodology of continuous GW searches with the Advanced LIGO O1 and O2 data see [10, 15]), but in the core they rely on performing Fourier transforms of the detectors’ output time series.

Some currently used continuous GW searches implement the \mathcal{F} -statistic methodology [16]. In this work we will study the output produced by one of them, the all-sky time-domain \mathcal{F} -statistic search [17] implementation, called the TD-Fstat search [18] (see the documentation in [19]). This data analysis algorithm is based on matched filtering; the best-matching template is selected by evaluating the SNR through maximization of the likelihood function with respect to a set of above-mentioned frequency parameters f and \dot{f} , and sky coordinates δ and α . By design, the \mathcal{F} -statistic is a reduced likelihood function [16, 17]. The remaining parameters characterizing the template—the GW polarization, amplitude and phase of the signal—do not enter the search directly, but are recovered after the signal is found. Recent examples of the use of the TD-Fstat search include searches in the LIGO and Virgo data [20–22], as well as mock data challenge [23].

Assuming that the search does not take into account time derivatives higher than \dot{f} , it is performed by evaluating the \mathcal{F} -statistic on a pre-defined grid of f , \dot{f} , δ and α values in order to cover the parameter space optimally and not overlook the signal, for which the true values of $(f, \dot{f}, \delta, \alpha)$ may fall between the grid points. The grid is optimal in the sense that for any possible signal there exists a grid point in the parameter space such that the expected value of the \mathcal{F} -statistic for the parameters of this grid point is greater than a certain value; for a detailed explanation see [17, 24].

The number of sky coordinates’ grid points as well as \dot{f} grid points increases with frequency. Consequently the volume of the parameter space (number of evaluations of the \mathcal{F} -statistic) increases; see e.g. figure 4 in [25]. In addition, the total number of resulting *candidate GW signals* (crossings of the pre-defined SNR threshold) increases. For high frequencies, this type of search is particularly computationally demanding.

The SNR threshold should preferably be as low as possible, because the continuous GWs are very weak—currently only upper limits for their strength are set [13, 14, 20–22]. A natural way to improve the SNR is to analyze long stretches of data since the SNR, denoted here by ρ , increases as a square root of the data length T_0 : $\rho \propto \sqrt{T_0}$. In practice, coherent analysis of the many-months-long observations (the typical length of a LIGO/Virgo scientific run is about one year) is computationally prohibitive. Depending on the method, the adopted coherence time ranges from minutes to days, and then additional methods are used to combine the results incoherently. The TD-Fstat search uses few-days-long data segments for coherent analysis. In the second step of the pipeline the candidate signals obtained in the coherent analysis are checked for coincidences in a sequence of time segments to confirm the detection of GW [20]. Here we explore an alternative approach to these studies, using results of a single data segment to classify a distribution of candidate signals as potentially interesting. In addition, we note that the coincidences step can be memory-demanding since the number of candidates can be very large, especially in the presence of spectral artifacts. The following work therefore explores an additional classification/flagging step for noise disturbances which can vastly reduce the number of signal candidates from a single time segment for further coincidences.

1.3. Aim of this research

The aim of this work is to classify the output of TD-Fstat search, the multi-dimensional distributions of candidate GW signals. Specifically, we study the application of a convolutional neural network (CNN) on the distribution of candidate signals obtained by evaluating the TD-Fstat search algorithm on a pre-defined grid of parameters. The data contains either pure Gaussian noise, Gaussian noise with injected astrophysical-like signals, or Gaussian noise with injected purely monochromatic signals, simulating spectral artifacts local to the detector (so-called stationary lines).

1.4. Previous works

The CNN architecture [26] has already proven to be useful in the field of the GW physics, in particular in the domain of image processing. Razzano and Cuoco [27] used CNNs for classification of noise transients in the

GW detectors. Beheshtipour and Papa [28] studied the application of deep learning on the clustering of continuous GW candidates. George and Huerta [29] developed the *Deep Filtering* algorithm for signal processing, based on a system of two deep CNNs, designed to detect and estimate parameters of compact binary coalescence signals in noisy time-series data streams. Dreissigacker *et al* [30] used deep learning (DL) as a search method for CWs from rotating neutron stars over a broad range of frequencies, whereas Gebhard *et al* [31] studied the general limitations of CNNs as a tool to search for merging black holes.

The last three papers discuss the DL as an alternative to matched filtering. However, it seems that the DL has too many limitations for application in the classification of GWs based on raw data from the interferometer (see discussion in [31]). For this reason we have decided to study a different application of DL. We consider DL a tool complementary to matched filtering, which allows one to effectively classify a large number of signal candidates obtained with the matched filter method. Instead of studying only binary classification, we have covered multi-label classification assessing the case of artifacts resembling the CW signal. Finally our work compares two different types of convolutional neural networks implementations: one-dimensional (1D) and two-dimensional (2D).

1.5. Structure of the article

The article is organized as follows. In section 2 we introduce the DL algorithms with particular emphasis on CNNs and their application in astrophysics. Section 3 describes the data processing we used to develop an accurate model for the TD-Fstat search candidate classification. Section 4 summarizes our results, which are further discussed. A summary and a description of future plans are provided in section 5.

2. Deep learning

DL [32] has commenced a new area of machine learning, a field of computer science based on special algorithms that can learn from examples in order to solve problems and make predictions, without the need to be explicitly programmed [33]. DL stands out as a highly scalable method that can process raw data without any manual feature engineering. By stacking multiple layers of artificial neurons (called neural networks) combined with learning algorithms based on back-propagation and stochastic gradient descent ([26] and references therein), it is possible to build advanced models able to capture complicated non-linear relationships in the data by composing hierarchical internal representations. The deeper the algorithm is, the more abstract concepts it can learn from the data, based on the outputs of the previous layers.

The DL is commonly used in commercial applications associated with computer vision [34], image processing [35], speech recognition [36] and natural language processing [37]. What is more, it is also becoming more popular in science. DL algorithms for image analysis and recognition have been successfully tested in many fields of astrophysics like galaxy classification [38] and asteroseismology [39]. Among the various DL algorithms there is one that might be especially useful in the domain of the GW physics—CNNs.

2.1. Convolutional neural network

A CNN is a deep, feed-forward artificial neural network (network that processes the information only from the input to the output), the structure of which is inspired by studies of the visual cortex in mammals, the part of the brain that specializes in processing visual information. The crucial element of CNNs is called a convolution layer. It detects local conjunctions of features from the input data and maps their appearances to a feature map. As a result the input data is split into parts, creating local receptive fields and compressed into feature maps. The size of the receptive field corresponds to the scale of the details to be examined in the data.

CNNs are faster than typical fully connected [40], deep artificial neural networks because sharing weights significantly decreases the number of neurons required to analyze data. They are also less prone to overfitting (the model learning the data *by heart* and preventing correct generalization). The *pooling layers* (subsampling layers) coupled to the convolutional layers might be used to further reduce the computational cost. They constrain the size of the CNN and make it more resilient to noise and translations, which enhances their ability to handle new inputs.

3. Method

3.1. Generation of data

To obtain a sufficiently large, labeled training set, we generate a set of TD-Fstat search results (distributions of candidate signals) by injecting signals with known parameters. We define three different classes of signals resulting in the candidate signal distributions used subsequently in the classification: 1) a GW signal, modeled here by injecting an astrophysical-like signal that matches the \mathcal{F} -statistic filter, corresponding to a spinning triaxial NS ellipsoid [17]; 2) an injected, strictly monochromatic signal, similar

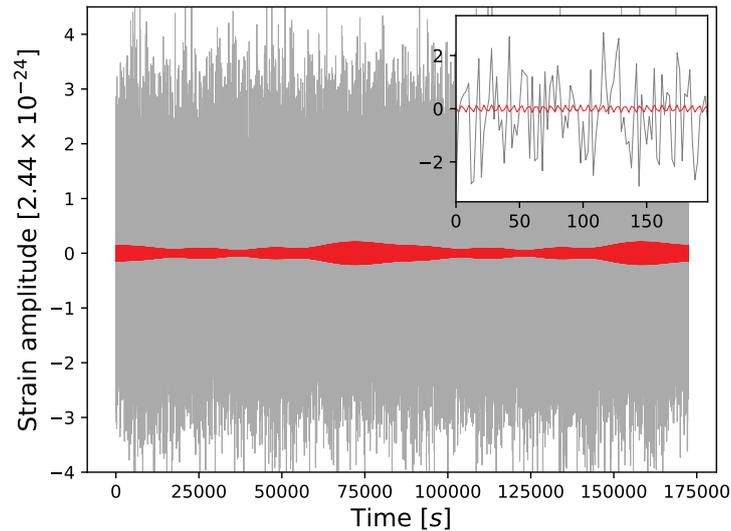


Figure 1. Example of continuous GW time-domain data input of TD-Fstat search. The grey time series of $T_0 = 2$ sidereal days length mimics the downsampled, narrow-banded data produced from the raw interferometer data [17, 20]. The data contains an almost-monochromatic astrophysical GW signal (red curve) of $\rho_{inj} = 10$, and the following parameters (see also table 1 for the parameters of the search and the text for more details): frequency $f = 2.16$ (in the units of the narrow band, between 0 and π), spindown $\dot{f} = -3.81 \times 10^{-8}$ (in dimensionless units of the pipeline, corresponding to $\dot{f}_{astro} = -3.03 \times 10^{-9} \text{ Hz s}^{-1}$; [17]), $\delta = 0.474$ (range between $-\pi/2$ and $\pi/2$) and $\alpha = 5.84$ (range between 0 and 2π). The reference frequency of the narrow band equals 100 Hz. Visible modulation is the result of the daily movement of the detector with respect to the astrophysical source, as well as of their relative positions, reflecting the quadrupolar nature of the detector's antenna pattern; in the case of a stationary line local to the detector such modulation is absent.

to realistic local artifacts of the detector (so-called stationary lines) [41], for which the \mathcal{F} -statistic is not an optimal filter; or 3) pure Gaussian noise, resembling the ‘clean’ noise output of the detector. These three classes are henceforth denoted by the **cgw** (continuous gravitational wave), **line** and **noise** labels, respectively.

To generate the candidate signals for the classification, the TD-Fstat search uses narrow-banded time series data as an input. In this work we focus on stationary white Gaussian time series, into which we inject astrophysical-like signals, or monochromatic ‘lines’ imitating the local detector’s disturbances. An example of such input data is presented in figure 1. It simulates the raw data taken from the detector, downsampled from the original sampling frequency (16 384 Hz in LIGO and 20 000 Hz in Virgo) to 0.5 Hz, and is divided into narrow frequency bands. Because the frequency of an astrophysical almost-periodic GW signal is not expected to vary substantially (only by the presence of \dot{f}), we use a bandwidth of 0.25 Hz, as in recent astrophysical searches [21, 22]. Each narrow frequency band is labeled by a reference frequency, related to the lower edge of the frequency band. Details of the input data are gathered in table 1. Additional TD-Fstat search inputs include the ephemeris of the detector (the position of the detector with respect to the Solar System Barycenter and the direction to the source of the signal, for each time of the input data), as well as the pre-defined grid parameter space of $(f, \dot{f}, \delta, \alpha)$ values, on which the search (\mathcal{F} -statistic evaluations) is performed [24].

In the signal-injection mode, the TD-Fstat search implementation adds an artificial signal to the narrow-band time domain data at some specific $(f, \dot{f}, \delta, \alpha)_{inj}$, with an assumed SNR ρ_{inj} . For long-duration, almost-monochromatic signals, which are the subject of this study, ρ_{inj} is proportional to the length of the time-domain segment T_0 and the amplitude of the signal h_0 (GW ‘strain’), and inversely proportional to the amplitude spectral density of the data S , $\rho_{inj} = h_0 \sqrt{T_0/S}$. The output SNR ρ for a candidate signal corresponding to $(f, \dot{f}, \delta, \alpha)_{inj}$ is a result of the evaluation of the \mathcal{F} -statistic on the Gaussian-noise time series with injected signal. The value of ρ at $(f, \dot{f}, \delta, \alpha)_{inj}$ is generally close to, but different from ρ_{inj} due to the random character of noise (ρ is related to the value of \mathcal{F} -statistic as $\rho = \sqrt{2(\mathcal{F} - 2)}$) (see [42] for detailed description). Furthermore, it is calculated on a discrete grid. This is the principal reason why we do not study individual signal candidates and their parameters, but the resulting ρ distributions in the $(f, \dot{f}, \delta, \alpha)$ parameter space (i.e. at the pre-defined grid of points), since the \mathcal{F} -statistic shape is complicated and has several local maxima, as shown e.g. in figure 1 of [43]. In the case of pure noise class, no additional signal is added to the original Gaussian data, but the data is evaluated in the pre-described range of $f, \dot{f}, \delta, \alpha$.

Subsequently, to produce instances of the three classes for further classification, the code performs a search around the randomly selected injection parameters $(f, \dot{f}, \delta, \alpha)_{inj}$, which in most cases fall in between the grid points, in the range of a few nearest grid points (± 5 grid points, see table 1). In the case of **cgw** all parameters are randomized, whereas for **line** we take $\dot{f} \equiv 0$. To be consistent in terms of the input data, e.g.

Table 1. Parameters of the input to the TD-Fstat search code (see e.g. [20]). Time series consist initially of random instances of white Gaussian noise, to which **cgws** or **lines** were added. Segment length T_0 is equal to 2 sidereal days with 2 s sampling time results in 86 164 data points. The \mathcal{F} -statistic (SNR) threshold is applied in order to select signal candidates above a certain SNR ratio, to exclude those that are most likely a result of random noise fluctuations.

Detector	LIGO Hanford
Reference band frequency	50, 100, 200, 300, 500, 1000 Hz (20, 250, 400, 700, 900 Hz for tests)
Segment length T_0	2 days
Bandwidth	0.25 Hz
Sampling time dt	2 s
Grid range	± 5 points
\mathcal{F} -statistic (SNR) threshold	14.5 (corresponding to $\rho = 5$)
Injected SNR ρ_{inj}	from 8 to 20 (from 4 to 20 for tests)

number of candidate signals, in the case of a stationary line, we also select a random sky position and perform a search in a range similar to the **cgw** case (this reflects the fact that spectral artifacts may also appear as clusters of candidate signal points in the sky). All the candidate signals crossing the pre-defined \mathcal{F} -statistic threshold (corresponding to the SNR ρ threshold) are recorded.

For each configuration of injected SNR ρ_{inj} and reference frequency of the narrow frequency band, we have produced 2500 signals per class (292 500 in total). For the **cgw** class we assumed the simplest distribution over ρ_{inj} , i.e. a uniform distribution, as the actual SNR distribution of astrophysical signals is currently unknown. We apply the same ‘agnostic’ procedure for the **line** class; their real distribution is difficult to define without a detailed analysis of weak lines in the detector data (our methodology allows us in principle to include such a realistic SNR distribution in the training set). To train the CNN, we put the lower limit of 8 on ρ_{inj} . Above this value, the peaks in the candidate signal ρ distributions for the **cgw** and **line** classes are still visible on the $\rho(f, \dot{f}, \delta, \alpha)$ plots (see figure 2 for the $\rho_{inj} = 10$ case). For $\rho_{inj} < 8$, the noise dominates the distributions, hindering the satisfactory identification of signal classes. Nevertheless, in the testing stage of the algorithm we extend the range of ρ_{inj} down to 4.

To summarize, each instance of the training classes is a result of the following input parameters: $(f, \dot{f}, \delta, \alpha)_{inj}$ and ρ_{inj} , and consist of the resulting distribution of the candidate signals: values of the SNR ρ evaluations of the TD-Fstat search at the grid points of the frequency f (in fiducial units of the narrow band, from 0 to π), spindown \dot{f} (in Hz s^{-1}), and two angles describing its sky position in equatorial coordinates, right ascension α (values from 0 to 2π) and declination δ (values from $-\pi/2$ to $\pi/2$); see figure 2 for an exemplary output distribution of the candidate signals.

The CNN required an input matrix of fixed size. However, the number of points on the distributions shown in figure 2 may vary for each simulation. Depending on the frequency (see table 1) it may increase a few times. To address this issue, we transformed point-based distributions into two different representations: a set of four 2D images (four distributions) and a set of five 1D vectors (five \mathcal{F} -statistic parameters).

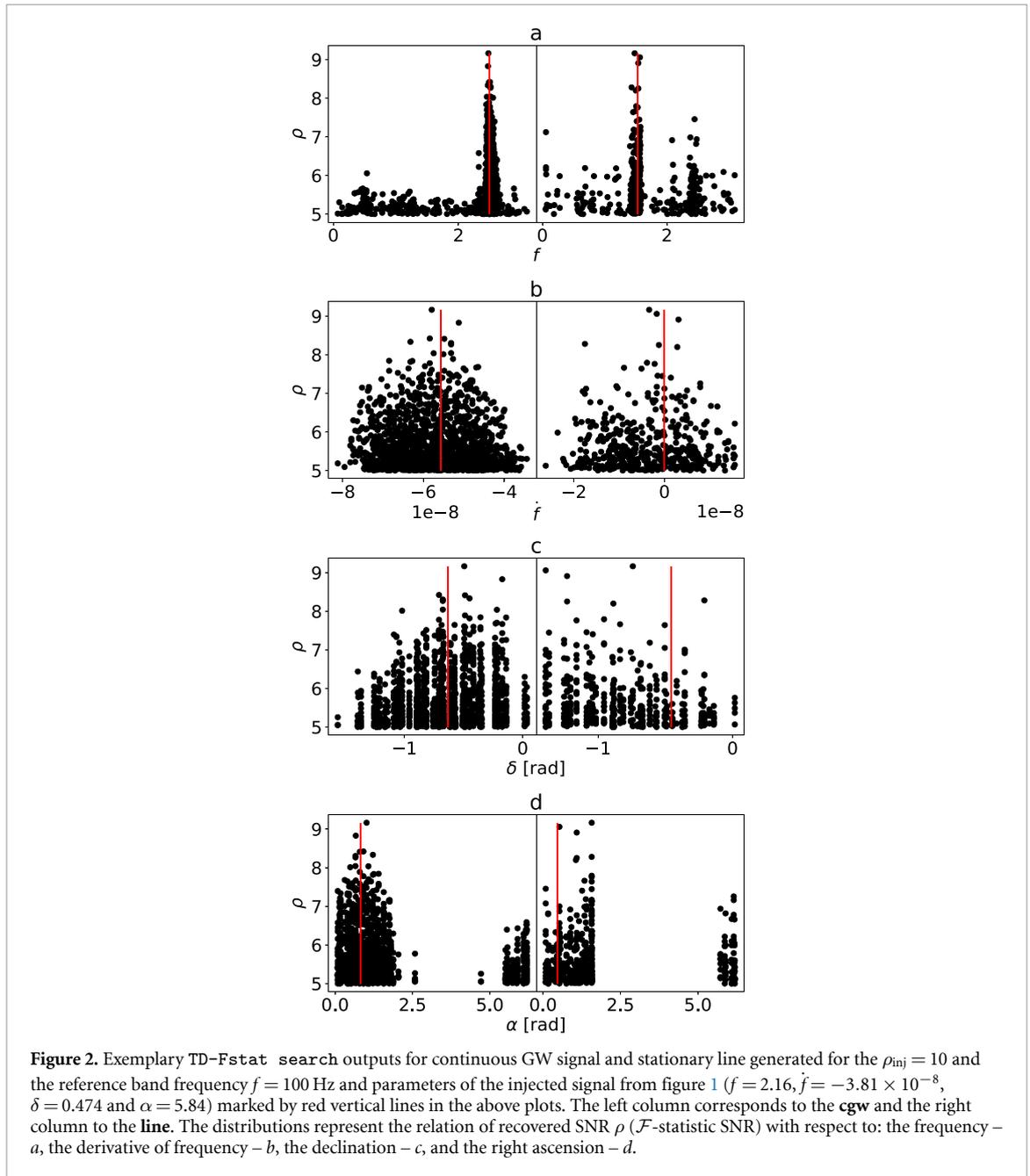
The image-based representation was created via conversion to a two-dimensional histogram (see figure 3) of the corresponding point-based distributions. Their sizes are 64×64 pixels. We chose this value empirically; smaller images lost some information after transformation, whereas bigger images led to significantly extended training time of the CNN we used.

The vector-based representation was created through selection of the 50 greatest values of the ρ distribution and their corresponding values from the other parameters (f, \dot{f}, δ and α). The length of the vector was chosen empirically. The main limitation was related to the density of the point-like distributions, which changed proportionally to the frequency. For the 50 Hz signal candidates, the noise class had sparse distributions of slightly more than 50 points. Furthermore, the vectors were sorted with respect to the ρ values (see figure 4); this step allowed slightly higher values of classification accuracy to be reached.

The created datasets were then split into three separate subsets: the training set (60% of signals from the total dataset), the validation set (20% of signals from the total dataset) and the testing set (20% of signals from the total dataset). The validation set was used during training to monitor the performance of the network (whether it overfits). The testing data was used after training to check how the CNN performs with unknown samples.

3.2. Neural network architecture

The generated datasets required two different implementations of the CNN. Overall we tested more than 50 architectures ranging from 2 – 6 convolutional layers and 1 – 4 fully connected layers for both models. The final layouts are shown in figures 5(a) and 5(b). The architectures that were finally chosen are based on a



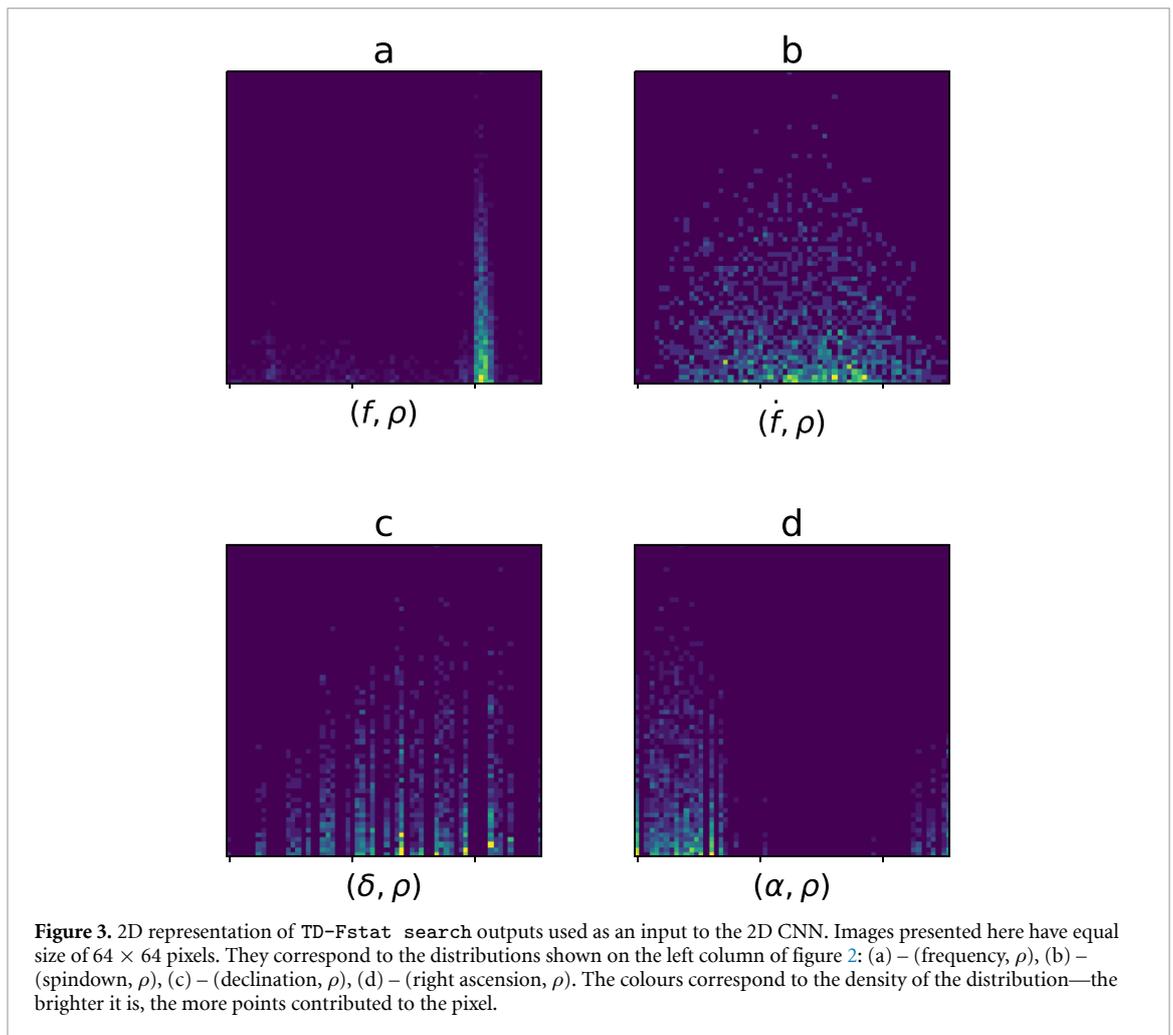
compromise between the model accuracy and the training time. Models larger than those specified in figures 5(a) and 5(b) achieved similar performance, but at the cost of significantly longer training time.

In the case of 1D CNN, the classifier containing three convolutional layers and two fully connected layers yielded the highest accuracy (more than 94% for the whole validation/test datasets). In contrast, the 2D CNN required four convolutional layers and two fully connected layers to reach the highest accuracy (85% over the whole validation/test datasets). The models were trained for 150 epochs which took 1 h for the 1D CNN and 15 h for the 2D CNN (on the same machine equipped with the Tesla K40 NVidia GPU).

To avoid overfitting we included dropout [44] in the architecture of both models. The final set of hyperparameters used for the training was as follows for both implementations (definitions of all parameters specified here can be found in [26]): ReLU as the activation function for hidden layers, softmax as the activation function for output layer, cross-entropy loss function, ADAM optimizer [45], batch size of 128, and 0.001 learning rate (see figures 5(a) and 5(b) for other details). The total number of parameters used in our models were the following: 52 503 for the 1D CNN, and 398 083 for the 2D CNN.

The CNN architectures were implemented using the Python Keras library [46] on top of the Tensorflow library [47], with support for the GPU. We developed the model on NVidia Quadro P6000² and performed

² Benefiting from the donation via the NVidia GPU seeding grant.



the production runs on the Cyfronet Prometheus cluster³ equipped with Tesla K40 GPUs, running CUDA 10.0 [48] and the cuDNN 7.3.0 [49].

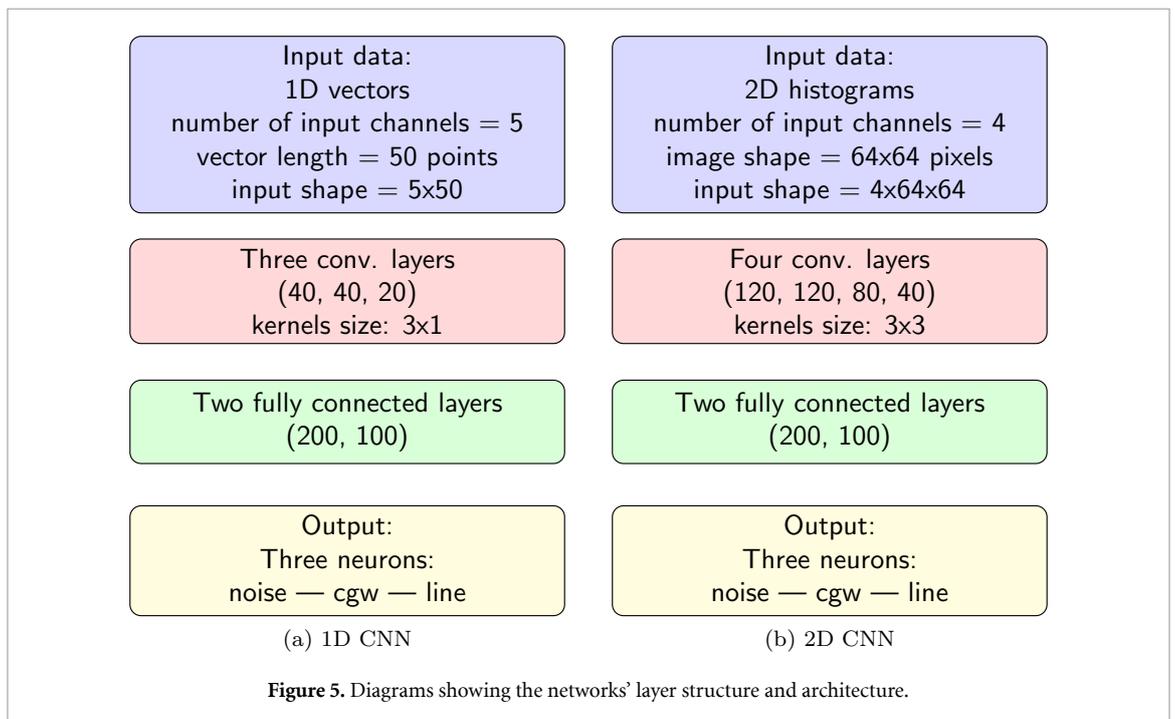
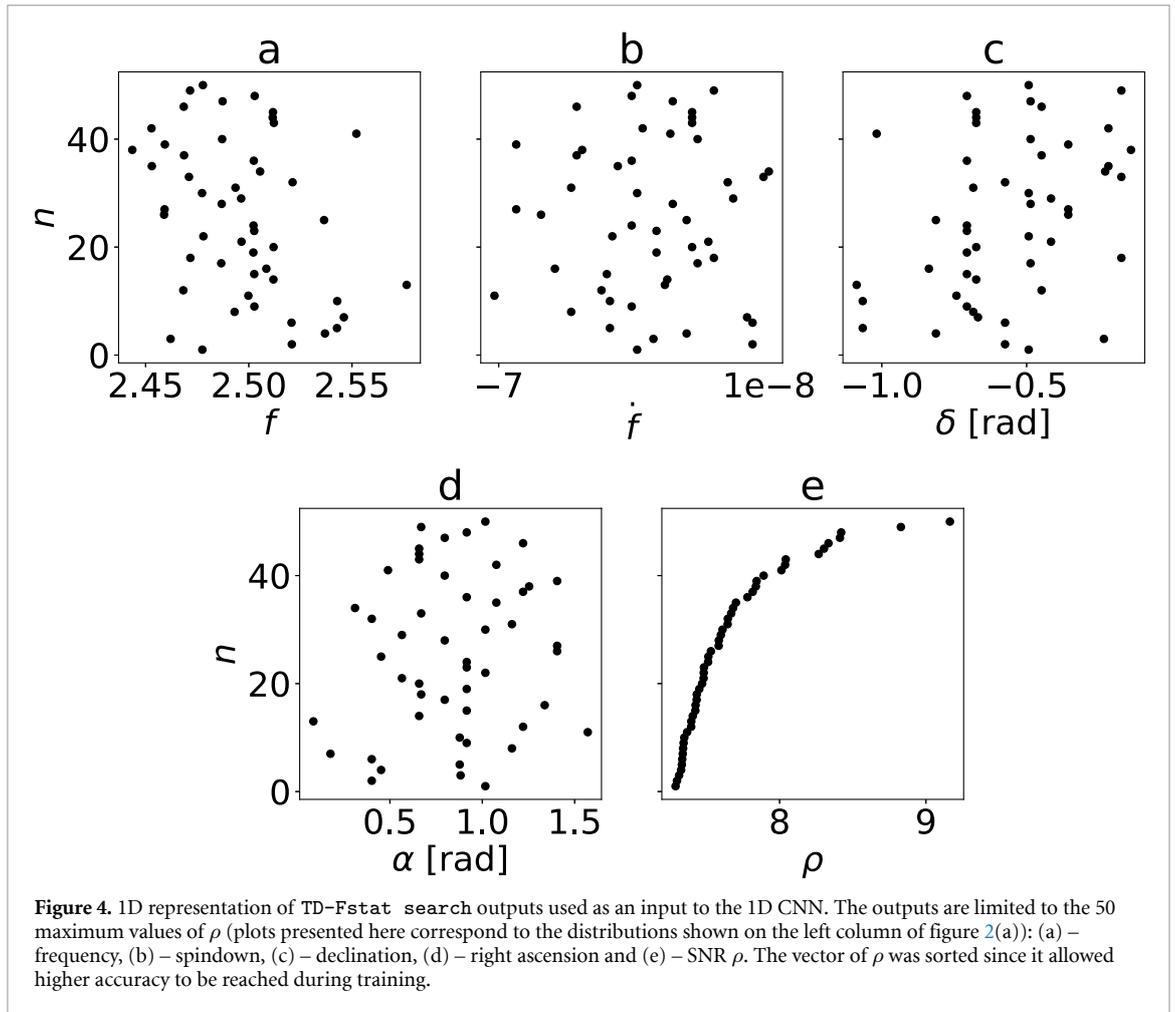
4. Results and discussion

Both CNNs described in section 3.2, figure 5(a) and figure 5(b) were trained on the generated datasets. During the training the model implementing 1D architecture was able to correctly classify 94% of all candidate signals, whereas the model implementing 2D architecture reached 85% accuracy (see the comparison between learning curves in figure 6). Accuracy is defined as the fraction of correctly predicted instances of data to total number of signal candidates. Since the very first epoch, the first model showed better ability to generalize candidate signals over a large range of frequencies and values of injected SNR ρ_{inj} .

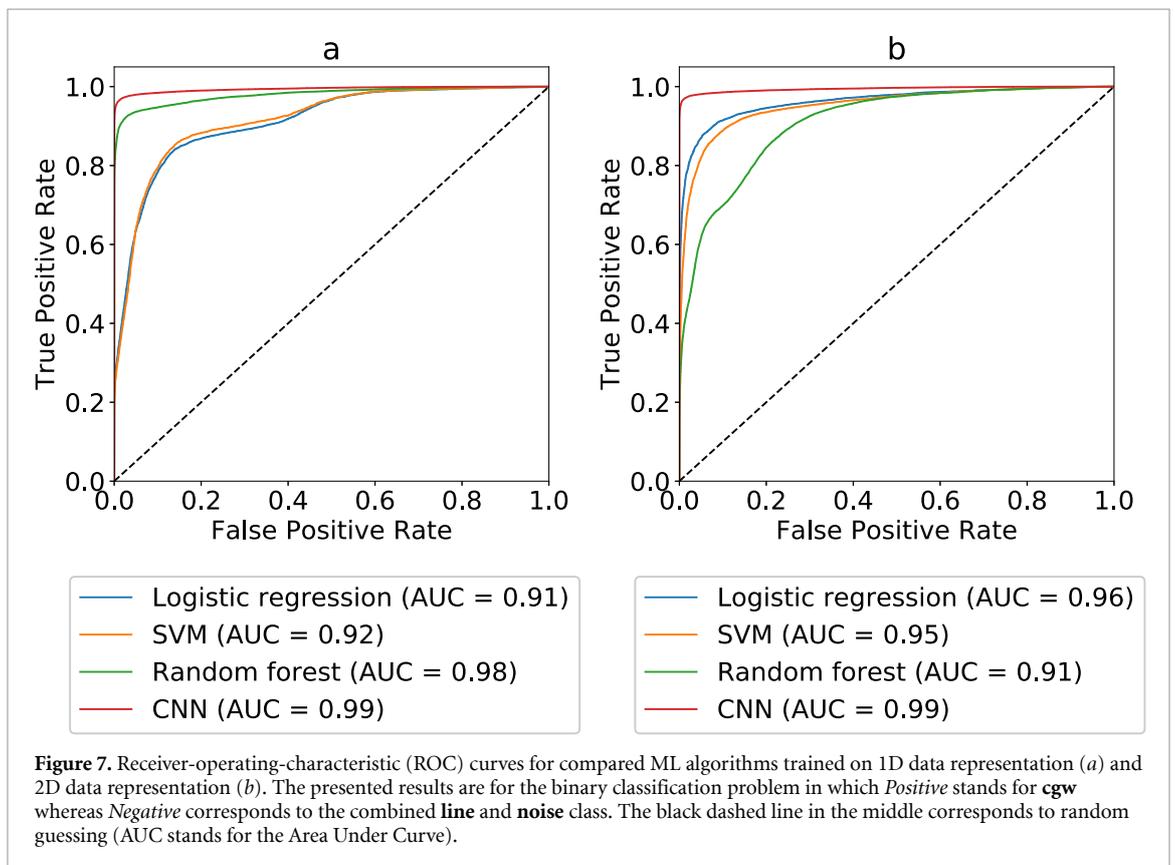
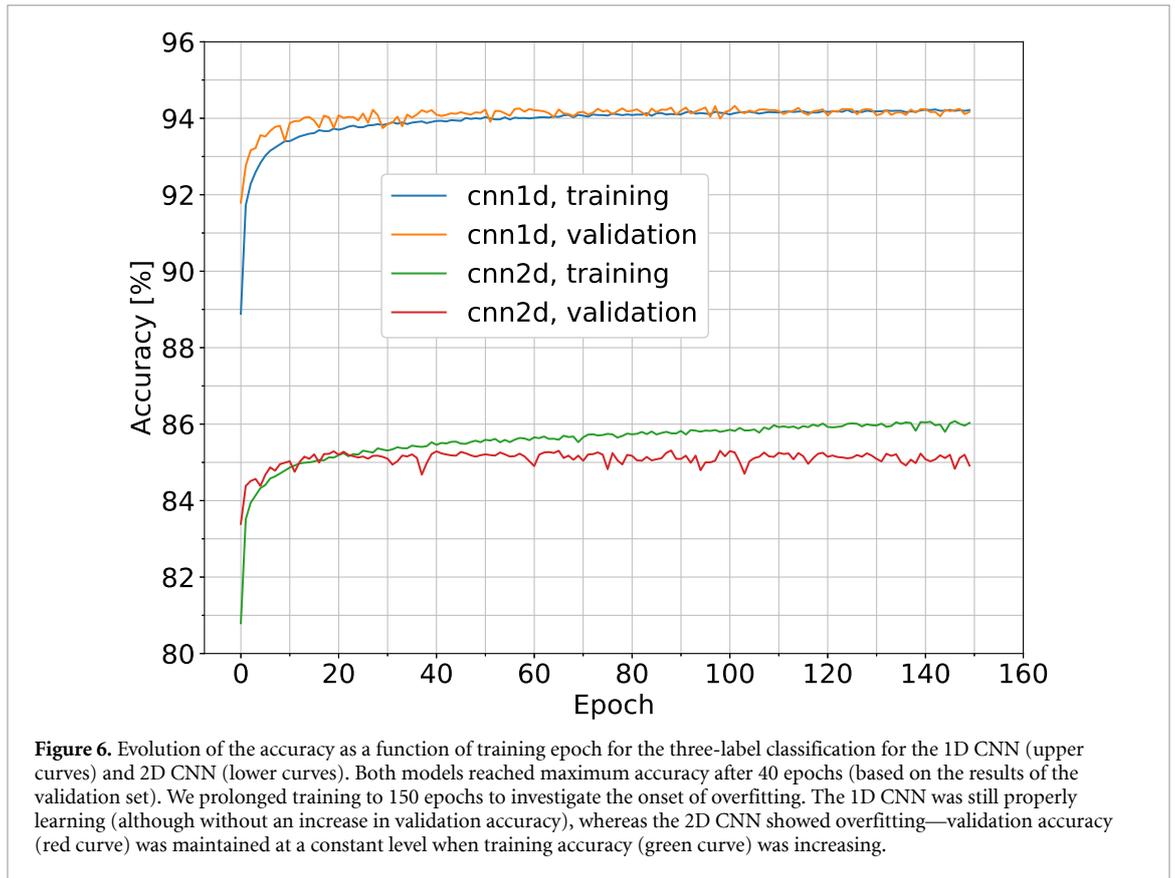
To justify the choice of a CNN as an algorithm suitable for the classification of signal candidates, we made a comparison test with different ML methods such as logistic regression, support vector machine (SVM) and random forest. For the test we modified the multi-label classification problem into a binary case to create receiver-operating-characteristic (ROC) curves. The classes of **line** and **noise** were combined into a single non-astrophysical class. The results of the comparison are shown in figures 7(a) and 7(b). The results shown in the left figure correspond to models trained and tested on 1D data representation, whereas the results shown in the right plot refer to 2D data representation. In both cases the CNNs outperformed other ML models. To further underline the differences, table 2 shows the detection probability (true positive rate, TPR) at a 1% of the false alarm rate (false positive rate or FPR).

CNNs achieved a similar level of detection probability, significantly outperforming the other algorithms. In the case of binary classification or detection of **cgw**, the 2D CNN seemed to be slightly better even with much lower accuracy as shown in figure 6. However the aim of our work was not only to classify GWs, but also to investigate their usefulness in the detection of stationary line artifacts. The data collected by the GW

³ Prometheus, Academic Computer Centre CYFRONET AGH, Kraków, Poland.



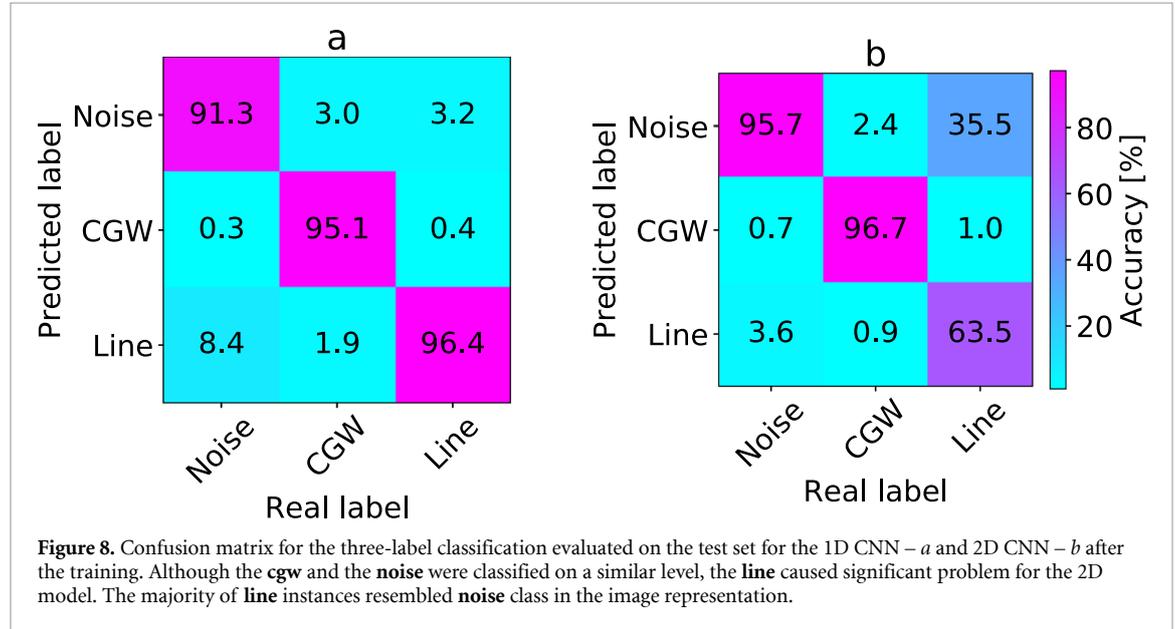
detectors is noise dominated and polluted by spectral artifacts in various frequency bands, which significantly impact the overall quality of data. Since the CNNs may potentially help in classification of lines to remove them from the science data, the analysis with respect to the multi-label problem is beneficial.



To decide which CNN architecture was more suitable to the multi-classification, our models were tested against unknown before samples (test dataset), after the training. The results are shown in figure 8 in the form of a confusion matrix. Both models were able to correctly classify the majority of **cgw** (95.1% for the 1D model and 96.7% for the 2D model) as well as the **noise** (91.3% and 95.7%, respectively). However, the

Table 2. Summary of detection probabilities for **cgw** at 1% false alarm rate for compared ML algorithms trained and tested on 1D and 2D data representations.

	Detection probability of cgw at 1% false alarm rate			
	<i>Logistic regression</i>	<i>SVM</i>	<i>Random forest</i>	<i>CNN</i>
1D data	33.8	31.9	89.2	96.3
2D data	72.8	57.6	38.4	96.8



difference in the classification of the **line** was significant. The 1D CNN was able to correctly classify 96.4% of line candidates, whereas for the 2D CNN it was only 63.5%. Although the 2D model seemed to be more suited for the binary classification task (detection of GW signal from the noise), the 1D CNN outperformed the 2D version in the multi-label classification.

Knowing the general capabilities of designed CNNs, we performed additional tests trying to understand the response of our models against signal candidates of specific parametrization. We generated additional datasets for particular values of SNR ρ_{inj} and frequency (see table 1). We expanded the ρ range down to the value of 4, which corresponds to the \mathcal{F} -statistic threshold for the signal candidate. This step allowed us to test the response of the CNN against unknown during training very weak signals that seemed to be indistinguishable from the noise.

The results are presented in figures 9(a) and 9(b) (for 1D and 2D CNNs, respectively). The 1D model presented significantly more stable behavior toward the candidates over the whole range of considered frequencies. It also maintained nearly stable accuracy for the data with the injected SNR $\rho_{inj} \geq 10$ (reaching a value of more than 90% for all of them). Interestingly, candidates with $\rho_{inj} < 8$ were correctly classified in 60–70% of samples for frequency ≥ 200 Hz. This was a relatively high value, taking into consideration their noise-like pattern (for **cgw** and **line** instances). This pattern had the biggest influence on the classification of the signal candidates generated for frequencies 50 and 100 Hz and $\rho_{inj} < 8$. The small number of points contributing to the peak (see figure 2(a) for comparison) with respect to the background noise made these candidates hardly distinguishable from the **noise** class.

On the other hand, the 2D CNN varied significantly in relation to the frequency. It reached the highest accuracy for the 100 Hz (99% for $\rho_{inj} > 10$). For the other frequencies, the maximum accuracy was gradually shifted toward increasing ρ_{inj} . Interestingly, the accuracy for 50 Hz reached the maximum for $\rho_{inj} = 10$; then it gradually decreased. The 2D CNN seemed to outperform the 1D model only for the narrow band of the frequency. Nevertheless, the general performance of this implementation was much worse.

Since the 1D CNN proved to be more accurate over a broad range of frequencies, we chose it as a more useful model in the classification of the \mathcal{F} -statistic signal candidates. Below we present the results of additional tests we performed to better understand its usability.

To test the model response toward a particular signal candidate, we computed sensitivity (in ML literature also referred to as the recall), defined as the fraction of relevant instances among the retrieved instances. Figure 10 presents the results. Classification of the **cgw** was directly proportional to ρ_{inj} up to a

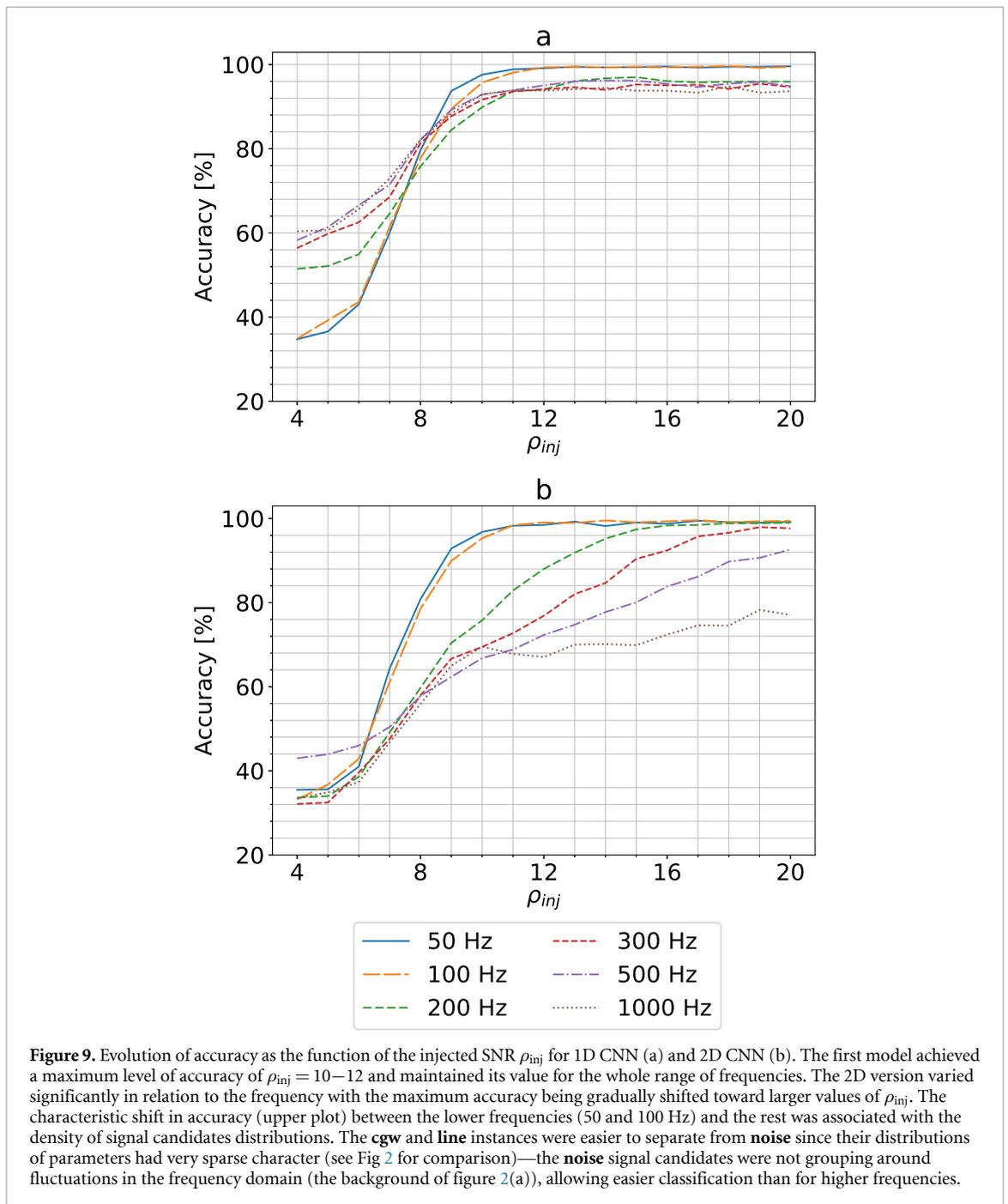


Figure 9. Evolution of accuracy as the function of the injected SNR ρ_{inj} for 1D CNN (a) and 2D CNN (b). The first model achieved a maximum level of accuracy of $\rho_{inj} = 10-12$ and maintained its value for the whole range of frequencies. The 2D version varied significantly in relation to the frequency with the maximum accuracy being gradually shifted toward larger values of ρ_{inj} . The characteristic shift in accuracy (upper plot) between the lower frequencies (50 and 100 Hz) and the rest was associated with the density of signal candidates distributions. The **cgw** and **line** instances were easier to separate from **noise** since their distributions of parameters had very sparse character (see Fig 2 for comparison)—the **noise** signal candidates were not grouping around fluctuations in the frequency domain (the background of figure 2(a)), allowing easier classification than for higher frequencies.

value of 11–12, and then the sensitivity was saturated around 95%–99% depending on the frequency. For ρ_{inj} approaching 4, sensitivity decreased to 0%. This result was expected since the injected signal at this level is buried so deeply in the noise that it is indistinguishable. Furthermore, by comparing figure 10 (a) with figure 9 (a), we deduced that the classification of **cgw** had the biggest influence on the total performance of the CNN.

The sensitivity of the **line** for higher frequencies (more than 300 Hz) was maintained at a relatively constant level of more than 95% even for the smallest ρ_{inj} . The decrease in sensitivity for lower frequencies was associated with the density of the signal candidate distribution. The outputs of TD-Fstat had sparser character, the lower the frequency was. The chosen 50 points for the input data were taken not only from the peak but also from the background noise (see top plots from figure 2). With decreasing ρ_{inj} , background points started to dominate and the candidates seemed to resemble **noise** class. This leads to misclassification of nearly all **line** samples for 50 Hz data.

In case of the **noise**, sensitivity was inversely proportional to frequency. Again this was associated with the density of the signal candidate distributions. For higher frequencies more points contributed to local

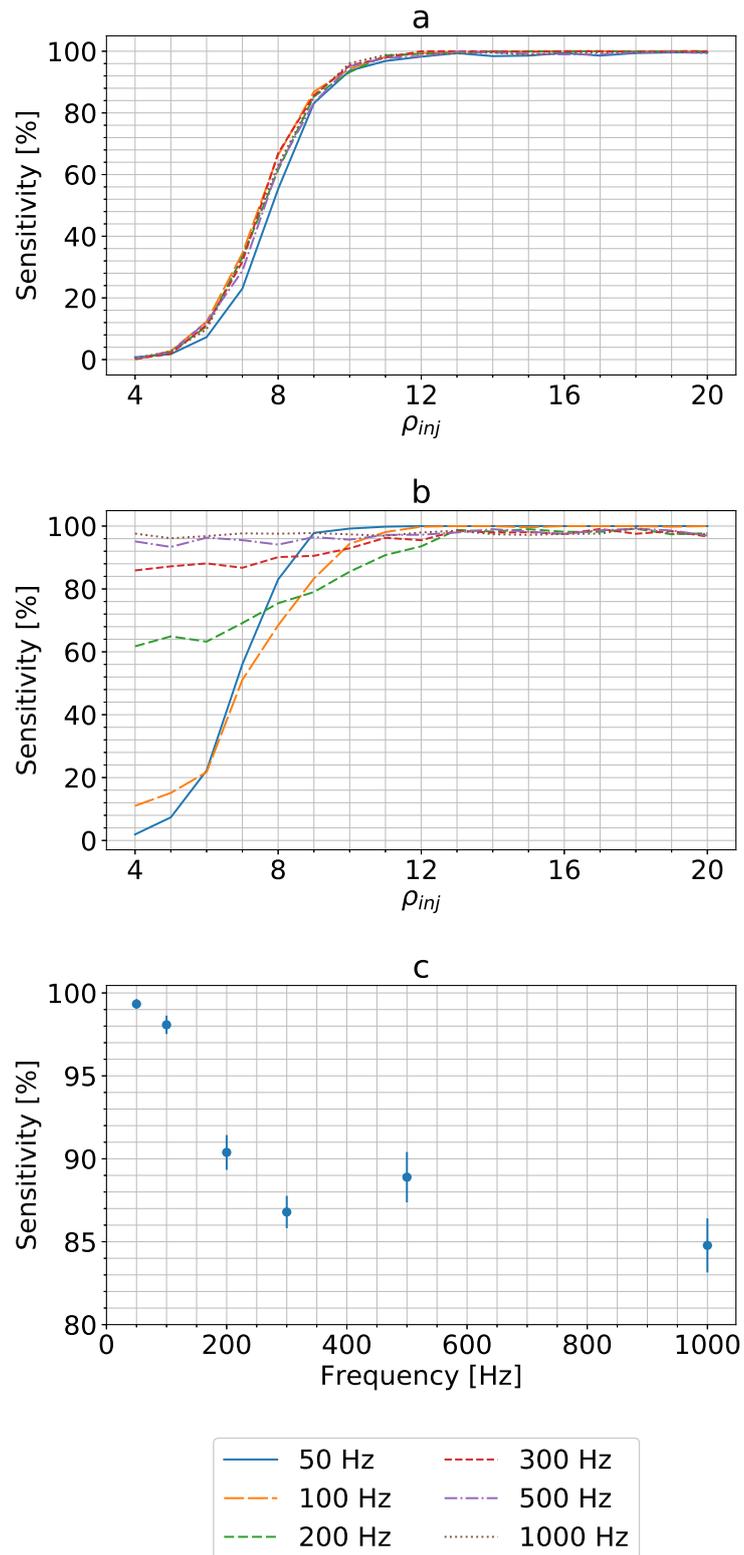


Figure 10. Evolution of sensitivity as a function of SNR ρ_{inj} of the 1D CNN for the three types of signal candidates: **cgw** – (a), **line** – (b), and **noise** – (c). The last panel shows average values for frequencies, because the **noise** classification sensitivity is not a function of the injected SNR ρ_{inj} , and stays approximately constant for each narrow-band frequency value.

fluctuations. As a result the 50 points chosen for the input data, instead of having random character, resembled different types of candidates.

We additionally performed tests on the signal candidates generated for different frequencies than specified in the table 1. We chose five new frequencies to test the model on: 20, 250, 400, 700, 900 Hz. The results were presented in figure 11. The 20 Hz case is missing since the number of available points (from initial distributions) to create a set of five 1D vectors was much smaller than the chosen length (some distributions for the **noise** class contained fewer than 10 points). Nevertheless, the CNN for the other

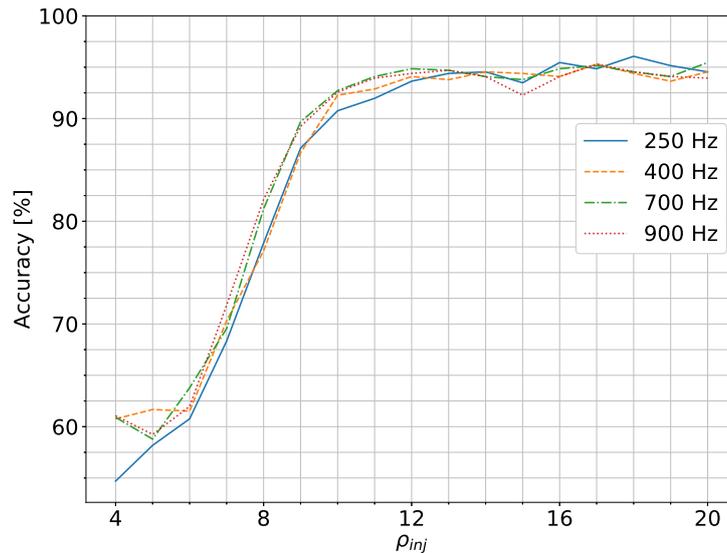


Figure 11. Evolution of accuracy as a function of the injected SNR ρ_{inj} for 1D CNN for signal candidates generated with frequencies different from those used for the training.

frequencies reached similar accuracies to those presented in figure 9(a). This result proved the generalization ability of the 1D CNN toward unknown frequencies. However, the limitation of the model was the minimum number of candidate signals available to create input data. Since this number was proportional to the number of grid points (frequency) of the searched signal, our CNN was not suited to searching for candidates below 50 Hz.

Although it is not immediately apparent from the 1D and 2D instances of the distributions of candidate signals, the \mathcal{F} -statistic values in the sky points contain non-negligible information about the signal content, and play a role in increasing the classification accuracy. A dedicated study of the influence of the distribution of the \mathcal{F} -statistic in the sky for astrophysical signals and detector artifacts will be addressed in a separate study.

5. Conclusions

We proved that the CNN can be successfully applied in the classification of TD-Fstat search results, multidimensional vector distributions corresponding to three signal types: GW signal, stationary line and noise. We compared 2D and 1D implementations of CNNs. The latter achieved much higher accuracy (94% with respect to 85%) over candidate signals generated for a broad range of frequencies and ρ_{inj} . For the majority of signals ($\rho_{inj} \geq 10$) the 1D CNN maintained more than 90% accuracy. This level of accuracy was preserved at the classification of the signal candidates injected in bands of unknown frequency (i.e. we show that the constructed CNNs are able to generalize the context).

The 2D CNN represented a different character. Although the overall accuracy was worse than that of the 1D model, the 2D version seemed to achieve better results as a binary classifier (between the **cgw** and the **noise**). Representation of the input data in the form of an image seemed to cause significant problems for the proper classification of the **line**. Even though the 2D CNN had worse generalization ability, it was able to outperform the 1D implementation for the narrow-band frequencies 100 Hz and below. Nevertheless, the 1D CNN, with its ability to generalize unknown samples (in particular with respect to the frequency), seemed to be the better choice for realistic applications.

This project is one of the few that research the application of DL as a supplementary component to MF. Adopting signal candidates as the DL input instead of raw data allows us to avoid problems that other researchers encountered. This approach limits the number of signals to those that exceeded the \mathcal{F} -statistic threshold, i.e. analyzed distribution instances are firmly characterized by known significance. As Gebhard *et al* [31] described, application of DL on raw data provides signal candidates of unknown or hard-to-define significance. Before DL could be used as a safe alternative to MF for the detection of GW, it has to be studied further. However, our results can already be considered in terms of a supporting role to MF. For example, it could be applied to the pre-processing of signal candidates for further follow-up steps via fast classification, and to limit the parameter space to be processed further. As our results show, a relatively simple CNN can also be used in the classification of spectral artifacts, e.g. as an additional tool for flagging and possibly also

removing spurious features from the data. Among the many possibilities for further development within the area of CW searches we are considering is the application of DL in the follow-up of signal candidates in multiple data segments (post-processing searches for patterns), as well as the analysis of data from a network of detectors.

Acknowledgments

The work was partially supported by the Polish National Science Centre Grant Nos. 2016/22/E/ST9/00037 and 2017/26/M/ST9/00978, the European Cooperation in Science and Technology COST action G2Net No. CA17137, and the European Commission Framework Programme Horizon 2020 Research and Innovation action ASTERICS under Grant No. 653477. The Quadro P6000 used in this research was donated by the NVidia Corporation via the GPU seeding grant. This research was supported in part by PL-Grid Infrastructure (the production runs were carried out on the ACK Cyfronet AGH Prometheus cluster). The authors thank Marek Cieřlar and Magdalena Sieniawska for useful insights and help with editing this publication.

Data availability statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID iDs

Filip Morawski  <https://orcid.org/0000-0002-6194-8239>

Michał Bejger  <https://orcid.org/0000-0002-4991-8213>

Paweł Ciecieląg  <https://orcid.org/0000-0002-5871-4730>

References

- [1] Einstein A 1916 *Sitzungsberichte der Königlich Preußischen Akademie der Wissenschaften* (Berlin: Deutsche Akademie der Wissenschaften) pp 688–96
- [2] Aasi J, Abbott B P, Abbott R and Abbott T et al 2015 *Class. Quantum Grav.* **32** 074001
- [3] Acernese F, Agathos M, Agatsuma K and Aisa D et al 2015 *Class. Quantum Grav.* **32** 024001
- [4] Abbott B P et al 2016 *Phys. Rev. Lett.* **116** 061102
- [5] Abbott B P et al 2017 *Phys. Rev. Lett.* **118** 221101
- [6] Abbott B P et al 2017 *Phys. Rev. Lett.* **119** 141101
- [7] Abbott B P et al 2017 *Phys. Rev. Lett.* **119** 161101
- [8] The LIGO Scientific Collaboration, the Virgo Collaboration, Abbott B P, Abbott R, Abbott T D, Abraham S, Acernese F, Ackley K, Adams C, Adhikari R X and et al 2018 *Phys. Rev.* **X9**, 031040 (2019)
- [9] Lasky P D 2015 *Publ. Astron. Soc. Aust.* **32** e034
- [10] Sieniawska M and Bejger M 2019 *Universe* **5** 217
- [11] Abbott B P et al 2017 *Astrophys. J.* **839** 12
- [12] Abbott B P et al 2017 *Phys. Rev. D* **96** 122006
- [13] Abbott B P et al 2017 *Phys. Rev. D* **96** 062002
- [14] Abbott B P et al 2018 *Phys. Rev. D* **97** 102003
- [15] Bejger M 2017 arXiv preprint 1710.06607v1 [gr-qc]
- [16] Jaranowski P, Królak A and Schutz B F 1998 *Phys. Rev. D* **58** 063001
- [17] Astone P, Borkowski K M, Jaranowski P, Pietka M and Królak A 2010 *Phys. Rev. D* **82** 022005
- [18] Time-domain \mathcal{F} -statistic pipeline repository 2018 (<https://github.com/mbejger/polgraw-allsky>)
- [19] Time-domain \mathcal{F} -statistic pipeline documentation 2018 (<http://mbejger.github.io/polgraw-allsky>)
- [20] Aasi J et al 2014 *Class. Quantum Grav.* **31** 165014
- [21] Abbott B P et al 2017 *Phys. Rev. D* **96** 122004
- [22] Abbott B P, Abbott R, Abbott T D, Abraham S, Acernese F, Ackley K, Adams C, Adhikari R X and et al 2019 *Phys. Rev. D* **100** 024004
- [23] Walsh S et al 2016 *Phys. Rev. D* **94** 124010
- [24] Pisarski A and Jaranowski P 2015 *Class. Quantum Grav.* **32** 145014
- [25] Poghossyan G, Matta S, Streit A, Bejger M and Królak A 2015 *Comput. Phys. Commun.* **188** 167–76
- [26] Goodfellow I, Bengio Y and Courville A 2016 *Deep Learning* (London: The MIT Press)
- [27] Razzano M and Cuoco E 2018 *Class. Quant. Grav.* **35** 095016
- [28] Beheshtipour B and Papa M A 2020 *Phys. Rev. D* **101** 064009
- [29] George D and Huerta E A 2018 *Phys. Lett.* **778** 64–70
- [30] Dreissigacker C, Sharma R, Messenger C and Prix R 2019 *Phys. Rev. D* **100** 044009
- [31] Gebhard T D, Kilbertus N, Harry I and Schölkopf B 2019 *Phys. Rev. D* **100** 063015
- [32] Lecun Y, Bengio Y and Hinton G 2015 *Nature* **521** 436–44
- [33] Samuel A L 1959 *IBM J. Res. Dev.* **3** 210–29
- [34] Maronidis A, Chatzilari E, Nikolopoulos S and Kompatsiaris I 2018 *Digit. Signal Process.* **74** 14–29
- [35] Druzhkov P N and Kustikova V D 2016 *Pattern Recognit. Image Anal.* **26** 9–15

- [36] Zhang Z, Geiger J, Pohjalainen J, Mousa A E D, Jin W and Schuller B 2018 *ACM Trans. Intell. Syst. Technol.* **9** 1–49
- [37] Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K and Kuksa P 2011 *J. Mach. Learn. Res.* **12** 2493–537
- [38] Lukic V and Brüggem M 2016 *Proc. Int. Astron. Union* **12** 217–20
- [39] Hon M, Stello D and Yu J 2017 *Mon. Not. R. Astron. Soc.* **469** 4578–83
- [40] Research B 2018 (accessed February 9 2018) *DeepBench* (<https://github.com/baidu-research/DeepBench>)
- [41] Covas P B, Effler A, Goetz E, Meyers P M, Neunzert A, Oliver M, Pearlstone B L, Roma V J, Schofield R M S, Adya V B and *et al* 2018 *Phys. Rev. D* **97** 082002
- [42] Jaranowski P and Krolak A 2009 *Analysis of Gravitational-Wave Data* (Cambridge, UK: Cambridge University Press)
- [43] Sieniawska M, Bejger M and Królak A 2019 *Class. Quantum Grav.* **36** 225008
- [44] Srivastava N, Hinton G, Krizhevsky A, Sutskever I and Salakhutdinov R 2014 *J. Mach. Learn. Res.* **15** 1929–58
- [45] Kingma D P and Ba J 2014 arXiv:1412.6980
- [46] Chollet F *et al* 2015 Keras (<https://keras.io>)
- [47] Abadi M *et al* 2015 TensorFlow: Large-scale machine learning on heterogeneous systems software available from tensorflow.org (<https://www.tensorflow.org/>)
- [48] Nickolls J, Buck I, Garland M and Skadron K 2008 *Queue* **6** 40–53
- [49] Chetlur S, Woolley C, Vandermersch P, Cohen J, Tran J, Catanzaro B and Shelhamer E 2014 arXiv: 1410.0759

Part IV

Relationship status: it's complicated
(Reconstruction of the astrophysical
parameters)

Chapter 5

Paper IV: Neural network reconstruction of the dense matter equation of state derived from the parameters of neutron stars

Neural network reconstruction of the dense matter equation of state derived from the parameters of neutron stars

F. Morawski and M. Bejger

Nicolaus Copernicus Astronomical Center, Polish Academy of Sciences, Bartycka 18, 00-716 Warsaw, Poland
e-mail: fmorawski@camk.edu.pl

Received 9 April 2020 / Accepted 10 June 2020

ABSTRACT

Context. Neutron stars are currently studied with an rising number of electromagnetic and gravitational-wave observations, which will ultimately allow us to constrain the dense matter equation of state and understand the physical processes at work within these compact objects. Neutron star global parameters, such as the mass and radius, can be used to obtain the equation of state by directly inverting the Tolman–Oppenheimer–Volkoff equations. Here, we investigate an alternative approach to this procedure.

Aims. The aim of this work is to study the application of the artificial neural networks guided by the autoencoder architecture as a method for precisely reconstructing the neutron star equation of state, using their observable parameters: masses, radii, and tidal deformabilities. In addition, we study how well the neutron star radius can be reconstructed using only the gravitational-wave observations of tidal deformability, that is, using quantities that are not related in any straightforward way.

Methods. The application of an artificial neural network in the equation-of-state reconstruction exploits the non-linear potential of this machine learning model. Since each neuron in the network is basically a non-linear function, it is possible to create a complex mapping between the input sets of observations and the output equation-of-state table. Within the supervised training paradigm, we construct a few hidden-layer deep neural networks on a generated data set, consisting of a realistic equation of state for the neutron star crust connected with a piecewise relativistic polytropes dense core, with its parameters representative of state-of-the-art realistic equations of state.

Results. We demonstrate the performance of our machine-learning implementation with respect to the simulated cases with a varying number of observations and measurement uncertainties. Furthermore, we study the impact of the neutron star mass distributions on the results. Finally, we test the reconstruction of the equation of state trained on parametric polytropic training set using the simulated mass–radius and mass–tidal-deformability sequences based on realistic equations of state. Neural networks trained with a limited data set are capable of generalising the mapping between global parameters and equation-of-state input tables for realistic models.

Key words. equation of state – dense matter – stars: neutron

1. Introduction

Neutron stars (NS) are currently the best astrophysical sites for studying the details of dense matter physics in conditions that are inaccessible for terrestrial experiments (see e.g. [Haensel et al. 2007](#) for a textbook introduction) Specifically, this refers to the equation of state (EOS) of dense, cold, neutron-rich matter at densities many times higher than the nuclear saturation density $\rho_s \approx 2.7 \times 10^{14} \text{ g cm}^{-3}$, corresponding to the nuclear saturation baryon density $n_s \approx 0.16 \text{ fm}^{-3}$.

Because the complete theory of many-body nuclear interactions is not known in full, recent efforts have been focussed on inferring the EOS from astrophysical observations of NSs. Recent observations include the NICER simultaneous measurements of the mass and radius of PSR J0030+0451 ([Riley et al. 2019](#); [Miller et al. 2019](#)), the 170817 binary NS inspiral detection and parameter estimation done by the LIGO and Virgo Collaborations ([Abbott et al. 2017a, 2019, 2018](#)), including the measurement of the masses and tidal deformability of the system components, accompanied by the observations of high-energy photons by the *Fermi* and INTEGRAL satellites ([Abbott et al. 2017b](#)), as well as several observations of massive $\approx 2 M_\odot$ NSs ([Demorest et al. 2010](#); [Fonseca et al. 2016](#); [Antoniadis et al. 2013](#); [Cromartie et al. 2020](#)). These observations provide indirect but nevertheless informative answers to the question of how compact objects are structured and, hence, the nature of their internal com-

position. The procedure is based on solving the equations of stellar structure, typically the Tolman–Oppenheimer–Volkoff (TOV) equations ([Tolman 1939](#); [Oppenheimer & Volkoff 1939](#)) for an assumed EOS (or class of EOSs) to subsequently compare the global observable NS parameters, such as gravitational mass M , radius R , and, recently, tidal deformability Λ as well ([Flanagan & Hinderer 2008](#); [Van Oeveren & Friedman 2017](#)) to observed values; in the simplest case of the TOV equation, there is a strong relation between the sequence of global NS parameters (EOS functionals), such as $M(R)$ or $M(\Lambda)$, and the pressure-density $p(\rho)$ relation defining the EOS. Therefore, given a set of astrophysical measurements, it is possible, in principle, to recover the EOS by inverting the TOV equations. In reality, however, astrophysical observations are affected by measurement errors and they are not distributed optimally in the parameter space, meaning that an observer doesn't have any freedom in selecting the intrinsic parameters, such as the mass, M , of the observed object, to optimally cover the range of pressure and density so that the part of the EOS relation that is of interest may be revealed.

The most common strategy in the estimation of EOS utilises Bayesian inference, which is based on the inversion of the TOV equations and a limited number of observations. Examples of this approach were recently presented in the following works: [Steiner et al. \(2010, 2013\)](#), [Raithel et al. \(2016\)](#), [Holt & Lim \(2019\)](#), [Fasano et al. \(2019\)](#), [Hernandez Vivanco et al. \(2019\)](#) and [Traversi et al. \(2020\)](#). Here, instead of directly inverting

the TOV equations, we study an alternative approach based on a machine-learning (ML) artificial neural network (ANN), inspired by the autoencoder (AE) architecture (Hinton & Zemel 1993; Goodfellow et al. 2016). Similar machine learning techniques applied to results of numerical simulations and measurements currently make up a field of active research; for example, Haegel & Husa (2019) show that the final mass and spin of a Kerr black hole can be predicted from the initial values of parameters of black hole components. Specifically, this has been explored in the field of the NS EOS, Fujimoto et al. (2018, 2020) presented an application of a feed-forward neural network to infer the EOS from NS mass-radius measurements, whereas Ferreira & Providência (2019) compare machine-learning neural networks and support vector machine regression methods in unveiling the nuclear EOS parameters from NS observations.

Here, we study the application of ML to infer the dense-matter EOS pressure-density $p(\rho)$ relations from a simulated set of NS observations, using a neural network trained on a purposefully simple data set, based on piecewise relativistic polytropic EOS. We performed the analysis using simulated data containing electromagnetic as well as gravitational-waves observables: radii, masses, and tidal deformabilities, applying the current knowledge of the NS mass distribution function, and varying the number of simulated observations and measurement uncertainties. While trained and tested on the piecewise relativistic polytropic EOS data set, our ML model was also validated on realistic EOS examples: it successfully recovers the SLy4 EOS (Douchin & Haensel 2001) as well as the APR EOS (Akmal et al. 1998) and the BSK20 EOS (Goriely et al. 2010). Additionally, we show the ANN network is flexible enough to generalise the mapping of the mass-radius $M(R)$ relation from the mass–tidal-deformability $M(\Lambda)$ relation, effectively allowing for the possibility of inference of the NS radius from several GW-only measurements.

The outline of the article is as follows. In Sect. 2, we discuss the choice of the machine learning algorithms used. Section 3 is devoted to the description of the input and output data generation procedures, with a particular emphasis on the EOS and the NS structure. Section 4 contains results of the neural-networks estimation of the dense-matter EOS from NS observables: $M(R)$ and $M(\Lambda)$. We discuss the results in Sect. 5. We conclude in Sect. 6 with a summary of our study.

2. Machine learning

The machine learning field of computer science is based on the premise that algorithms can learn from examples in order to solve problems and make predictions without needing to be explicitly programmed (Samuel 1959). Among the many ML algorithms, the ANN currently belong to the most popular. Complex ANN consisting of many neurons combined with various training algorithms (such as the backpropagation and stochastic gradient descent – for textbook review, see e.g. Goodfellow et al. 2016 and references therein) are able to capture complicated non-linear relationships in the data by composing hierarchical internal representations. The complex (in other words, deeper) the algorithm is, the more abstract features it can, in principle, learn from the data.

The main motivation for employing ANN in our project is associated with non-linear potential of the this ML model. Since each neuron in the network is basically a non-linear function, it is possible to create a complex mapping between the input and the output of the model. This characteristic is necessary for the estimation of EOS based on observables, even when excluding

uncertainties since the analytical relations between the input and output parameters are non-linear.

The input to our model are astrophysical measurements of NS parameters (presented to the ANN as two arrays of M, R or M, Λ concatenated into one), whereas at the output we obtain an array of similar shape (concatenated p, ρ values). Further in text, we present additional project in which we reconstruct radius based on gravitational observables (M and Λ concatenated into one vector), the ANN output consists only of radius values. By definition, the size of the output is half the size of the input.

3. Data preparation

In this section, we describe the design of parametric EOSs and methods used to obtain the stellar parameters.

3.1. Equations of state and stellar structure

In order to cover a sufficiently-broad and representative space of solutions corresponding to $M(R)$ and $M(\Lambda)$ sequences, we employ the following simplified, parametric approach to the EOS. We assume the measurement of the low-density part of the EOS and adopt the SLy4 EOS description of Haensel & Pichon (1994) and Douchin & Haensel (2001) up to some baryon density n_0 , comparable to and typically larger than the nuclear saturation density ($n_s \equiv 0.16 \text{ fm}^{-3}$). At the n_0 a relativistic polytrope (Tooper 1965),

$$p(n) = \kappa n^\gamma, \quad \rho c^2 = \frac{P(n)}{\gamma - 1} + nm_b c^2, \quad (1)$$

replaces the SLy4 EOS. For each polytrope, the pressure p and the mass-energy density ρc^2 are defined using the pressure coefficient κ , the polytropic index γ responsible for the stiffness of the matter, and the mass of the baryon m_b . We select the γ index as a parameter of choice; consequently, κ and m_b are determined by demanding the chemical and mechanical equilibrium at n_0 . The first polytrope with γ_1 ends at some density, $n_1 > n_0$, where a second relativistic polytrope with γ_2 is attached, and continues until n_2 , where a polytrope with γ_3 starts. The bottom-left panel of Fig. 1 shows a schematic plot of the EOS. The parameter ranges are collected in Table 1.

For a given EOS, we solve the equations of hydrostatic equilibrium for a spherically symmetric distribution of mass. The space-time metric is:

$$ds^2 = e^{\nu(r)} c^2 dt^2 - \frac{dr^2}{1 - 2GM(r)/rc^2} - r^2(d\theta^2 + \sin^2 \theta d\phi^2), \quad (2)$$

with the gravitational mass $M(r)$ inside the radius r

$$\frac{dM(r)}{dr} = 4\pi\rho(r)r^2. \quad (3)$$

Then the resulting Tolman-Oppenheimer-Volkoff equations (Tolman 1939; Oppenheimer & Volkoff 1939),

$$\begin{aligned} \frac{dP(r)}{dr} &= -\frac{G}{r^2} \left(\rho(r) + \frac{P(r)}{c^2} \right) \left(M(r) + \frac{4\pi r^3 P(r)}{c^2} \right) \\ &\times \left(1 - \frac{2GM(r)}{c^2 r} \right)^{-1}, \end{aligned} \quad (4)$$

supplied with the equation for the metric function $\nu(r)$,

$$\frac{d\nu(r)}{dr} = -\left(\frac{2}{P(r) + \rho(r)c^2} \right) \frac{dP(r)}{dr}, \quad (5)$$

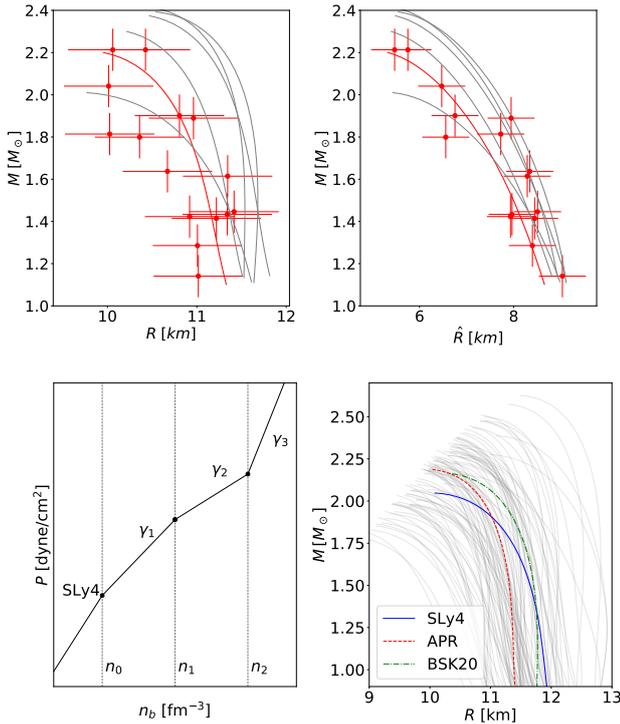


Fig. 1. *Top panels:* left – sample $M(R)$ relations; right – corresponding $M(\hat{R})$ relations. Red curve and corresponding observations: example of a training datum, consisting on 20 points from the $M(R)$ and $M(\hat{R})$, selected assuming random normal distribution with the mean value equal to true values and standard deviations $\sigma_M = 0.1 M_\odot$, $\sigma_{\hat{R}} = 1.0$ km and $\sigma_{\hat{R}} = 1.0$ km. The configurations with the same M and R have, in general, different \hat{R} . *Bottom panels:* left – schematic of a model EOS, composed in the low-density part from the realistic crust of the SLy4 EOS (Haensel & Pichon 1994; Douchin & Haensel 2001) and piecewise relativistic polytropes (Tooper 1965), right – mass-radius $M(R)$ relations generated using the piecewise relativistic polytrope model (thin solid grey curves) and astrophysical models of NS sequences, based on the following EOSs: the SLy4 EOS (solid blue curve), the APR EOS (dashed red curve) and the BSK20 EOS (dash-dotted green curve).

are integrated from the center towards the surface (where the pressure, P , vanishes, which defines the radius of the star, R) using a Runge-Kutta 4th order numerical scheme with a variable integration step (Press et al. 1992) for a range of central parameters of the EOS (e.g. the central pressures P_c) resulting in the $M(R)$ sequence.

In addition to gravitational mass, M , and radius, R , we also calculate the static lowest-order tidal deformability of the star, defined as

$$\lambda = \frac{2}{3} R^5 k_2. \quad (6)$$

The parameter λ represents the star’s reaction on the external tidal field (e.g. exerted by a companion in a tight binary system, as observed in Abbott et al. 2017a). It is obtained in the lowest-order approximation, by calculating the second (quadrupole) tidal Love number k_2 (Love 1911), a function of stellar parameters and hence the EOS:

$$\begin{aligned} k_2 = & \frac{8}{5} x^5 (1 - 2x)^2 (2 - y + 2x(y - 1)) (2x(6 - 3y + 3x(5y - 8)) \\ & + 4x^3 (13 - 11y + x(3y - 2) + 2x^2(1 + y)) \\ & + 3(1 - 2x)^2 (2 - y + 2x(y - 1)) \ln(1 - 2x))^{-1}, \end{aligned} \quad (7)$$

Table 1. Ranges of piecewise polytrope EOS parameters used in the study.

	n_0 [fm^{-3}]	γ_1	n_1 [fm^{-3}]	γ_2	n_2 [fm^{-3}]	γ_3
min	0.1	2.5	n_0	2.0	n_1	3.0
max	0.2	3.5	0.3	2.5	0.4	4.0

Notes. n_0 , n_1 , and n_2 are the baryon densities at which the relativistic polytropes (Eq. (1), see also Fig. 1) with indices γ_1 , γ_2 , and γ_3 are attached, respectively ($n_0 < n_1 < n_2$). The step sizes used in the data generation were: $\delta_{\gamma_1} = \delta_{\gamma_3} = 0.25$, $\delta_{\gamma_2} = 0.1$, $\delta_{n_0} = 0.025$. In case of δ_{n_1} and δ_{n_2} the step varied during computation since the minimum values dependent on n_0 and n_1 , respectively, but were not larger than δ_{n_0} .

where $x = GM/Rc^2$ denotes the star’s compactness, and y the solution of

$$\begin{aligned} \frac{dy}{dr} = & -\frac{y^2}{r} - \frac{1 + 4\pi Gr^2/c^2(P/c^2 - \rho)}{(r - 2GM(r)/c^2)} y \\ & + \left(\frac{2G/c^2(M(r) + 4\pi r^3 P/c^2)}{\sqrt{r}(r - 2GM(r)/c^2)} \right)^2 + \frac{6}{r - 2GM(r)/c^2} \\ & - \frac{4\pi Gr^2/c^2}{r - 2GM(r)/c^2} \left(5\rho + 9P/c^2 + \frac{(\rho + P/c^2)^2 c^2}{\rho dP/d\rho} \right), \end{aligned} \quad (8)$$

evaluated at the stellar surface (Flanagan & Hinderer 2008; Van Oeveren & Friedman 2017). In the following we use the mass-normalised value of the λ parameter,

$$\Lambda = \lambda (GM/c^2)^{-5}. \quad (9)$$

In order to relate the Λ parameter with the stellar radius R , we produce a following radius-like parameter $\hat{R}(M, \Lambda)$, which we call the tidal radius (proposed in Wade et al. 2014):

$$\hat{R} = 2M\Lambda^{1/5}. \quad (10)$$

This function of Λ and M is henceforth used in the study. Sample $M(R)$ and $M(\hat{R})$ relations are presented in the top panels of Fig. 1, along with simulated measurement points (the procedure of obtaining them is presented in Sect. 3.2). Moreover, in the bottom right panel of Fig. 1, we present a bundle of $M(R)$ relations used in the training, generated for piecewise relativistic polytropes to compare the training set with the astrophysical models based on the SLy4, APR and BSK20 EOSs. The training data cover the space of parameters similar to astrophysical models; therefore, we expect that the algorithm will generalise the EOS reconstruction toward previously unseen types of curves (types of curves on which it wasn’t trained on).

3.2. Neutron-star mass function and simulated measurement errors

In order to investigate the influence of the amount and precision of data – the number of observations N and their measurement errors – we restrict the sample of $M(R)$ and $M(\hat{R})$ for masses from the astrophysically-realistic range above $1 M_\odot$, which corresponds to observed NS masses in Galactic binary NS systems (Alsing et al. 2018), and in the GW170817 and GW190425 events (Abbott et al. 2017a, 2020). We exclude from further analysis all piecewise polytropic solutions that are not compatible with current state of observations, giving the NS maximum masses above $1.9 M_\odot$ (conservative choice motivated by the observations

of massive NSs, see Demorest et al. 2010; Fonseca et al. 2016; Antoniadis et al. 2013; Cromartie et al. 2020).

To realistically recreate astrophysical observations, we select the measurement points from a realistic NS mass function (mass probability distribution) out of which the mass values are to be randomly selected. Consistently with current observations of NSs in the Galaxy, the mass function is represented by a double-Gaussian distribution (Alsing et al. 2018) with the main peak around the Chandrasekhar mass and the second, smaller peak corresponding to the NS masses close to $2 M_{\odot}$, namely $\mathcal{N}(\mu_1, \sigma_1) + \mathcal{N}(\mu_2, \sigma_2)$, where $\mu_1 = 1.34$, $\sigma_1 = 0.07$, $\mu_2 = 1.80$, $\sigma_2 = 0.21$ (see Fig. 1 in Alsing et al. 2018 for details). This NS mass function is consistent with a recent GW observation of a heavy NS binary system (Abbott et al. 2020).

The training data set is prepared by assuming that the measurements are witness to a measurement error. After randomly choosing N values of the gravitational mass M from the above-mentioned mass distribution, we construct the training samples corresponding to a given $M(R)$ or $M(\hat{R})$ point by drawing values from normal distributions $\mathcal{N}(M(R), \sigma_i)$ or $\mathcal{N}(M(\hat{R}), \sigma_i)$, with $i = M, R, \hat{R}$ respectively. For the σ_i parameters, we chose the values in the range of 0.01 – $0.1 M_{\odot}$ for σ_M and 0.01 – 1 km for σ_R . Uncertainties for tidal deformabilities are defined in terms of \hat{R} and were in range $\sigma_{\hat{R}} = 0.01$ – 1 km, which corresponds to $\sigma_{\Lambda} = 10^2$ – 10^3 . An example of the training sequence, obtained assuming the double-Gaussian mass distribution and $\sigma_M = 0.1 M_{\odot}$, $\sigma_R = \sigma_{\hat{R}} = 1.0$ km is shown in Fig. 1 (marked red in top panels). Gray curves (and the red curve) correspond to $M(R)$ and $M(\hat{R})$ relations computed with the TOV equations for some examples of the piecewise polytropic EOS described in Sect. 3.1. The scattered points correspond to the actual input data fed to our model; they are based on the red curve values according to the procedure describe above.

In total the training dataset contains 13982 piecewise polytropic EOSs (see Table 1 for the details), out of which the $M(R)$ and $M(\hat{R})$ sequences were produced by solving the TOV equations. For each of these sequences, we then randomly selected N values of M (N equal to 10, 15, 20, 30, 40, or 50 observations) using the above-mentioned NS mass distribution, and recover the corresponding values of R and \hat{R} . For each input EOS, this procedure is repeated a fixed number of $N_s = 30$ times. As a result, each input EOS is represented in the training stage by N_s different realisations of N observations of $M(R)$ or $M(\hat{R})$, subject to “observational errors” by drawing the values from normal distributions parametrised by σ_i . This step allows us to effectively estimate the errors that ANN makes in the prediction of output sequences, that is, the error of reconstructing pressures and densities. To compute these errors, we then calculate the differences between the estimated output and the ideal expected result (the “ground truth” values). The errors are averaged for each measurement in a given collection of realisations. This step is repeated for all the EOSs in the training dataset, returning the set of error distributions: in the case of 20 measurements, we recover 20 distributions. The error bars presented for the output values in Sect. 4 are the mean values of these distributions.

We contrast the reconstruction errors with the ANN loss function as they represent different features. Loss function is a metric defining overall performance of the ANN in terms of how well the predicted values are to original ground truth values in general. Reconstruction errors give detailed information about differences between predicted pressures and densities and their corresponding ground truth values. Furthermore, the reconstruction error changes with respect to the values of pressure and density.

At the last step of data preparation, pressures and densities were converted to the decimal logarithm values and scaled together with masses, radii, and tidal radii to the range $(0, 1)$. Rescaling is required by the ANN non-linear functions since their domain is in the range $(0, 1)$.

The data sets were then split into two separate subsets: a training set (70% of all instances from the total dataset) and the testing set (30% of the total dataset). In cases when the ANN was tested against the measurements corresponding to realistic tabulated EOS, the simulated measurement data was generated in the same way as for the piecewise polytropic EOS.

3.3. ANN

In the design of the ANN, we used parts of the AE architecture. The AE (Kramer 1991) is a specific type of network capable of learning how to efficiently compress and encode the data into the so-called latent space representation and, later, to decompress and reconstruct the initial data as closely as possible. The core functionality of the AE is data dimensionality reduction. During training, AE learns how to ignore the noise and extract only crucial features of the data. Dimensionality reduction is in particular useful in the application of AEs aiming for data clustering. Specifically, the features of latent representation of an AE may be used to characterise the data, for example, by employing the conditional training of the variational AE using the training data with parameter labels to subsequently study the distribution of parameters in the latent space of variables. In the present exploratory work, we employ the simplest encoder-decoder structure of AE and we do not use the properties of the latent space, leaving that aspect to a future work.

The final architecture of our ANN was chosen based on empirical tests based on the data. As an output criterion for the loss function we use the mean squared error (MSE). We tested architectures ranging from one to eight hidden layers. The optimal network, reaching the minimum value for MSE, was the one containing four hidden layers with the following number of neurons: 512, 256, 256, 512.

The final set of hyper-parameters used for the training was the following (parameters defined as in e.g. Goodfellow et al. 2016):

- ReLU as the activation function for hidden layers,
- sigmoid activation function for the output layer,
- ADAM optimiser (Kingma & Ba 2014),
- batch size of 128,
- 0.001 learning rate.

The ANN architecture was implemented using the Python Keras library (Chollet 2015) on top of the TensorFlow library (Abadi et al. 2015), with support for the GPU. We developed the model on the NVidia Quadro P6000¹ and performed the production runs on the Cyfronet Prometheus cluster² equipped with Tesla K40 GPUs, running CUDA 10.0 (Nickolls et al. 2008) and the cuDNN 7.3.0 (Chetlur et al. 2014).

4. Results

The results presented below are split into subsections. The first present the results of EOS reconstruction from $M(R)$ and $M(\hat{R})$ simulated measurements with errors using ANN trained on piecewise polytropic EOS results. The second subsection

¹ Benefiting from the donation via the NVidia GPU seeding grant.

² Prometheus, Academic Computer Centre CYFRONET AGH, Kraków, Poland.

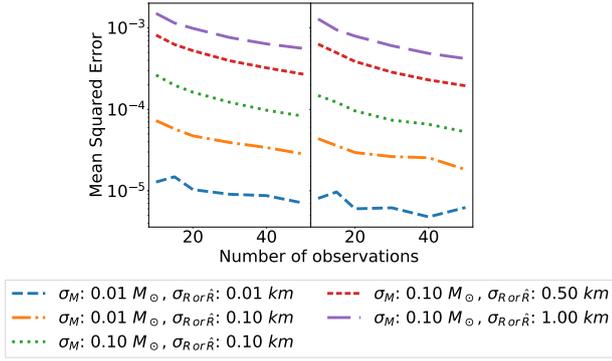


Fig. 2. Dependence of the MSE (ANN loss function described in Sect. 3.2) as a function of number of observations N and measurement uncertainties in the case of EOS estimation based on $M(R)$ (left figure) and $M(\hat{R})$ (right figure) observations.

shows the application of ANN on the realistic EOS resulting from microscopic calculations (SLy4 EOS, Douchin & Haensel 2001), that is, a reconstruction of the EOS which is not a piecewise polytropic model. We also study an application of the ANN to a direct reconstruction of the NS radius with the GW-only observations of the tidal deformability.

4.1. Translating the NS observations, $M(R)$ or $M(\Lambda)$, to EOS

Here, we present the results of the ANN application to the reconstruction of the EOS based on the gravitational mass, M , and radius, R , observations, which may be a result of electromagnetic observations of, for example, the NICER mission, as well as the EOS reconstruction based on the gravitational-wave observations of mass, M , and tidal deformability, Λ (which we reparametrise as \hat{R} ; see Eq. (10)). The ANN described in Sect. 3.3 is trained on data sets with varying number of observations and measurement uncertainties. The resulting figures of merit – the ANN loss function MSE – are shown in Fig. 2 with the left plot corresponding to the EOS reconstruction using $M(R)$ data and the right using $M(\hat{R})$ data.

The accuracy of EOS estimation is mostly influenced by the assumed measurement uncertainties in both presented cases. The value of MSE is proportional to the measurement errors; it reaches the highest value for the largest of considered uncertainties: $\sigma_M = 0.1 M_\odot$ for mass, M , $\sigma_R = 1$ km for the radius, R , and $\sigma_{\hat{R}} = 1$ km for the tidal radius, \hat{R} . Furthermore, the number of observations N had little effect on the MSE; the increase in N slightly decreased the MSE in all studied cases.

The top panels of Fig. 3 present two examples of the EOS reconstruction for the small and large measurement uncertainties in the case of $N = 20$ $M(R)$ observations. Both EOSs are recovered correctly within reconstruction errors computed as specified in the Sect. 3.2 with respect to the ground-truth values of related input EOSs (marked with dashed lines on the right panel). The error ranges in case of EOS estimation using $M(R)$ data for different measurement uncertainties are presented in the upper part of Table 2. The resulting σ_p and σ_ρ spans increase proportionally, with increasing σ_M and σ_R . Furthermore, in all presented cases, the ranges for pressure errors were wider than density errors, indicating that ANN was more uncertain in the reconstruction of pressure values. The increase in the reconstruction errors is expected because the overall performance of the ANN was worse during the training (see the blue and violet curves in Fig. 2 for comparison). Another effect related to the worse per-

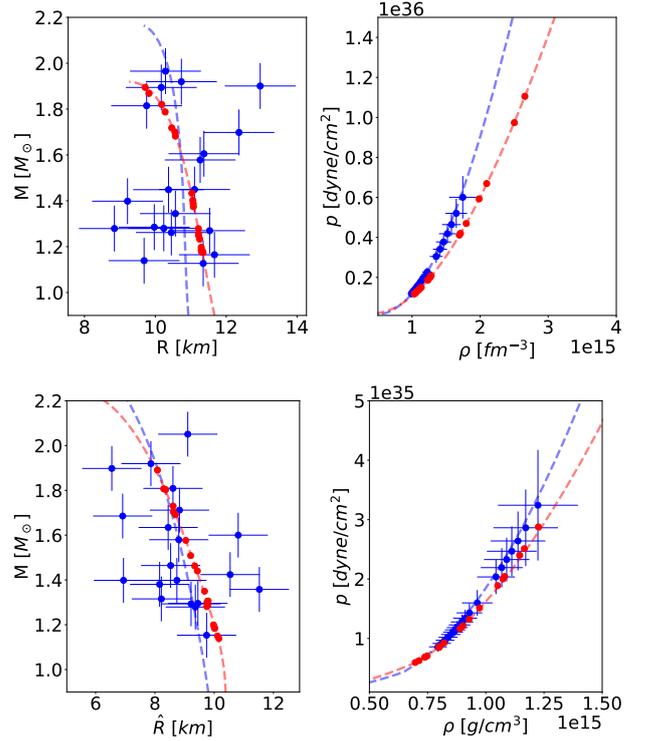


Fig. 3. Top panels: example of the input data ($M(R)$ measurements with errors, left plot), and corresponding output data from ANN ($p(\rho)$ relation, right plot) for the estimation of EOS from $M(R)$. Both input samples consist of 20 observations with masses randomly selected from a mass distribution (Sect. 3.2) and measurement uncertainties equal to $\sigma_M = 0.1 M_\odot$, $\sigma_R = 1$ km (blue sample), $\sigma_M = 0.01 M_\odot$, $\sigma_R = 0.01$ km (red sample). For the description of the uncertainties on the output, see the text. Bottom panels: example of the input data ($M(\hat{R})$ measurements with errors, left plot), and corresponding output data from ANN ($p(\rho)$ relation, right plot) for the estimation of EOS from $M(\hat{R})$. Both input samples consist of 20 observations with masses randomly selected from a mass distribution (Sect. 3.2) and measurement uncertainties equal to $\sigma_M = 0.1 M_\odot$, $\sigma_R = 1$ km (blue sample), $\sigma_M = 0.01 M_\odot$, $\sigma_{\hat{R}} = 0.01$ km (red sample). Dashed curves correspond to original (ground-truth, error-free) sequences of input and output of the TOV equations. Presented examples correspond to different EOSs.

formance of the EOS reconstruction is the significant increase of errors and decrease in the accuracy of reconstruction for higher $p(\rho)$ values. Several effects may be responsible for this result, for example, the impact of adopted NS mass distribution. Naturally, if the dataset contains a smaller number of high (close to $2 M_\odot$) M samples, the high $p(\rho)$ values of the EOS are less efficiently probed. As a result, the EOS reconstruction is less certain overall in this range. We discuss alternative explanations in Sect. 5.

The examples shown in the bottom panels of Fig. 3 corresponded to the EOS reconstruction using $M(\hat{R})$ data for two cases of small and large measurement uncertainties and $N = 20$ observations. Both EOSs are estimated correctly within reconstruction errors with respect to the ground-truth values of corresponding EOSs (marked with dashed lines) and the errors are proportional to the values of the density and pressure, similarly to the $M(R)$ case.

4.2. Application on realistic EOS

We test the ANN trained on piecewise polytropic EOS (and the TOV solutions obtained with them) on a realistic microscopic

Table 2. Reconstruction error ranges for σ_p and σ_ρ of the ANN for studied measurement uncertainties in case of EOS reconstruction for the $M(R)$ data (upper table) and $M(\hat{R})$ data (lower table).

$M(R)$ input data	σ_p [dyne cm ⁻²]	σ_ρ [g cm ⁻³]
$\sigma_M = 0.01 M_\odot, \sigma_R = 0.01$ km	$5 \times 10^{12} - 10^{13}$	$10^{33} - 7 \times 10^{33}$
$\sigma_M = 0.01 M_\odot, \sigma_R = 0.1$ km	$9 \times 10^{12} - 4 \times 10^{13}$	$2 \times 10^{33} - 2 \times 10^{34}$
$\sigma_M = 0.1 M_\odot, \sigma_R = 0.1$ km	$10^{13} - 5 \times 10^{13}$	$5 \times 10^{33} - 4 \times 10^{34}$
$\sigma_M = 0.1 M_\odot, \sigma_R = 0.5$ km	$2 \times 10^{13} - 10^{14}$	$5 \times 10^{33} - 8 \times 10^{34}$
$\sigma_M = 0.1 M_\odot, \sigma_R = 1.0$ km	$4 \times 10^{13} - 2 \times 10^{14}$	$8 \times 10^{33} - 10^{35}$
$M(\hat{R})$ input data	σ_p [dyne cm ⁻²]	σ_ρ [g cm ⁻³]
$\sigma_M = 0.01 M_\odot, \sigma_{\hat{R}} = 0.01$ km	$5 \times 10^{12} - 8 \times 10^{12}$	$10^{33} - 4 \times 10^{33}$
$\sigma_M = 0.01 M_\odot, \sigma_{\hat{R}} = 0.1$ km	$9 \times 10^{12} - 3 \times 10^{13}$	$2 \times 10^{33} - 2 \times 10^{34}$
$\sigma_M = 0.1 M_\odot, \sigma_{\hat{R}} = 0.1$ km	$10^{13} - 4 \times 10^{13}$	$4 \times 10^{33} - 2 \times 10^{34}$
$\sigma_M = 0.1 M_\odot, \sigma_{\hat{R}} = 0.5$ km	$2 \times 10^{13} - 10^{14}$	$5 \times 10^{33} - 7 \times 10^{34}$
$\sigma_M = 0.1 M_\odot, \sigma_{\hat{R}} = 1.0$ km	$4 \times 10^{13} - 2 \times 10^{14}$	$8 \times 10^{33} - 9 \times 10^{34}$

Notes. The reconstruction errors are computed as specified in Sect. 3.2.

EOSs: the SLy4 EOS (Douchin & Haensel 2001), the APR EOS (Akmal et al. 1998) and the BSK20 EOS (Goriely et al. 2010). To generate data for this test, we followed the approach detailed in Sect. 3 as in the case of the polytropic EOSs. Figure 4 contain the results of EOS reconstruction using the $M(R)$ data (top panels) and $M(\hat{R})$ (bottom panels) for $N = 20$, $\sigma_M = 0.1 M_\odot$ and $\sigma_R = 1$ km. Among the realistic microphysical EOS we have considered, the EOS relation reconstructed for the APR EOS and BSK20 EOS agree with original (ground truth) input values almost perfectly, whereas the SLy4 EOS model is reconstructed less precisely; however, the reconstructed EOS relation agrees with the ground truth values (dashed line) within reconstruction errors from Table 2.

These results show that the ANN trained on a relatively simple dataset of relativistic piecewise polytropes is able to generalise the task of EOS reconstruction towards an unknown during its training of realistic EOS.

4.3. Radius reconstruction using Λ measurements

We also present the results of an additional analysis which aims to directly reconstruct the NS radius R from GW-only observations of masses and tidal deformabilities. As Eq. (9) shows, the tidal deformability is related to M and R and to the second Love number k_2 , all of which are functionals on the EOS. In the general case, the $\Lambda - R$ relation cannot be simply obtained (see e.g. Zhang et al. 2020; De et al. 2018 and references therein). From the point of view of the $M(R)$ diagram, the relation between Λ and R depends on the slope of $M(R)$, which is indirectly a function of the NS susceptibility to deformations (see Sieniawska et al. 2019 for examples of configurations with the same M and R , but different Λ values; their Sect. 3.2, Figs. 9 and 10).

In order to study the ability of reconstructing the R based on M and Λ observations, we modified the ANN described in the Sect. 3.3 since, for this case, the size of the output was twice smaller (M and \hat{R} concatenated at the input and R and at the output). We considered the same measurement uncertainties σ_M and $\sigma_{\hat{R}}$ as in the EOS reconstruction. The results of the ANN training are shown in Fig. 5 in terms of MSE.

Similarly as in the estimation of EOS, the strongest influence on the radius computation had the measurement uncertainties. The MSE changed in range between 5×10^{-5} and 10^{-2} for the data varying in uncertainties from $\sigma_M = 0.01 M_\odot$ and $\sigma_{\hat{R}} = 0.01$ km to $\sigma_M = 0.1 M_\odot$ and $\sigma_{\hat{R}} = 1$ km. Moreover, the impact of observations number was insignificant.

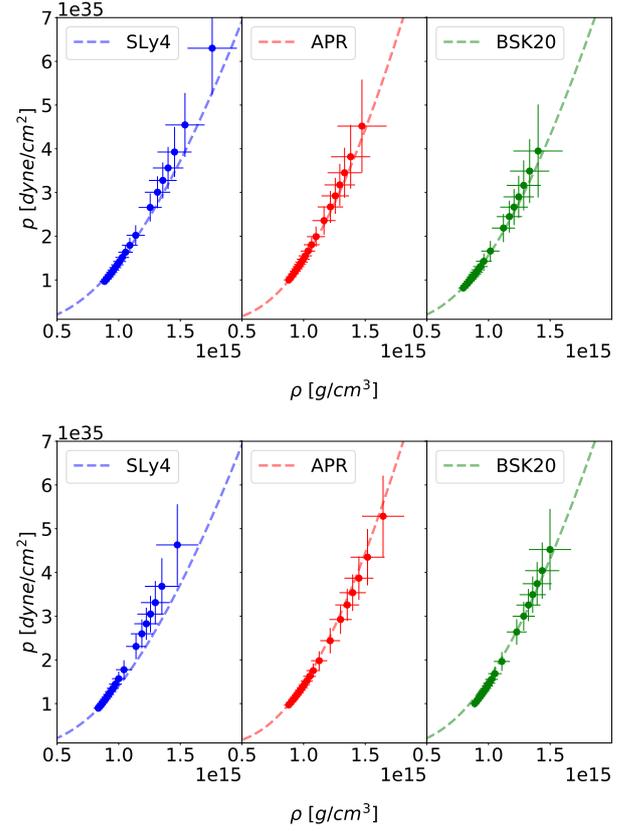


Fig. 4. Top panels: ANN-reconstructed EOS from the $M(R)$ data for the SLy4 EOS model (left plot), the APR EOS model (middle plot) and the BSK20 EOS model (right plot). Results are computed for the input $M(R)$ data consisting of 20 observations with measurement uncertainties equal to $m_{\text{err}} = 0.1 M_\odot$, $\hat{r}_{\text{err}} = 1$ km. Bottom panels: ANN-reconstructed EOS from the $M(\hat{R})$ data for the same EOS as in the top panels. Results for the input $M(\hat{R})$ data consisting of 20 observations with measurement uncertainties equal to $m_{\text{err}} = 0.1 M_\odot$, $\hat{r}_{\text{err}} = 1$ km. Dashed lines correspond to the exact EOS relations.

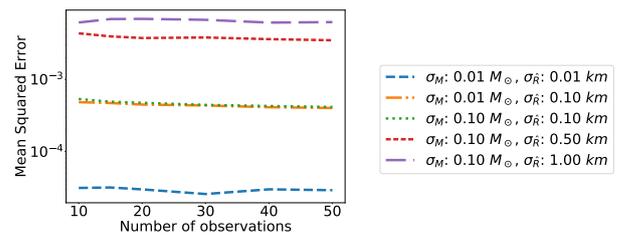


Fig. 5. Evolution of MSE (ANN loss) in the function of number of observations and measurement uncertainties in case of R computation based on $M(\hat{R})$.

The examples of the radius estimation are presented in Fig. 6. Top panels show the radius computed by ANN using piecewise polytropic data for two cases of measurement uncertainties: $\sigma_M = 0.01 M_\odot$ and $\sigma_{\hat{R}} = 0.01$ km (red sample) and $\sigma_M = 0.1 M_\odot$ and $\sigma_{\hat{R}} = 1.0$ km (blue sample). Bottom panels present the estimated radius for the data corresponding to the realistic cases: the SLy4 EOS, APR EOS, and BSK20 EOS for $\sigma_M = 0.1 M_\odot$ and $\sigma_{\hat{R}} = 1.0$ km. Within the reconstruction errors, σ_R , all cases were correctly reconstructed, in comparison to the dash line representing the exact values of radii computed from the TOV equations. However, the σ_R increase proportionally to σ_M and $\sigma_{\hat{R}}$. Furthermore, the errors varied randomly with respect to the value

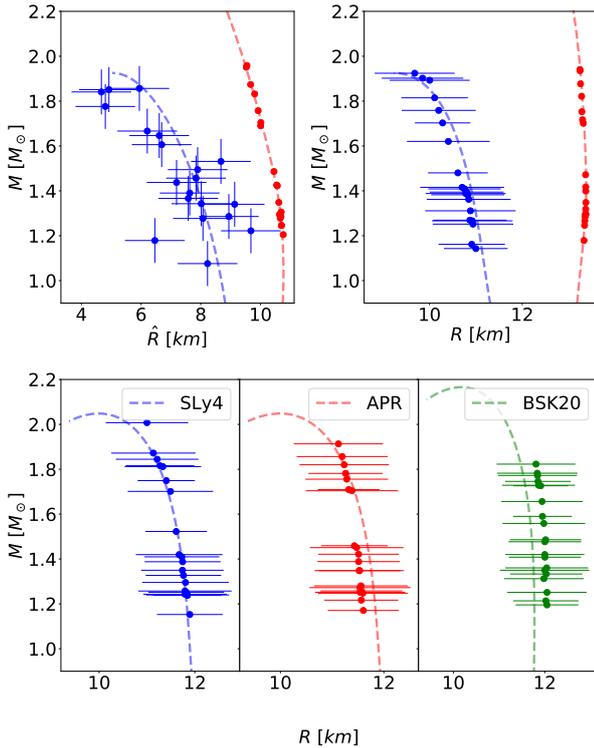


Fig. 6. Top panels: example of the input data (left plot) and corresponding output data from ANN (right plot) in the case of R computation based on $M(\hat{R})$ for the piecewise relativistic polytropes. Bottom panels: radius reconstructed from the $M(\hat{R})$ data for the SLy4 EOS model (left plot), the APR EOS model (middle plot), and the BSK20 EOS model (right plot). All results were computed for the input $M(R)$ data consisting of 20 observations with measurement uncertainties equal to $\sigma_M = 0.1 M_\odot$, $\sigma_{\hat{R}} = 1$ km. Dashed lines correspond to exact values obtained by solving the TOV equations.

of radius. In contrast to pressure and density errors, no trend in radius errors was present.

In general, the radius reconstruction from tidal deformability using ANN is possible, which demonstrates an additional ability on the part of ANN to build a non-linear mapping between astrophysical parameters of interest.

5. Discussion

The above results point us to a conclusion that the application of ANN in EOS reconstruction from astrophysical observations works for the majority of our data, with decreasing reliability for data with the largest measurement errors.

In comparison with similar approaches to the same problem (Fujimoto et al. 2018, 2020), our work extends the study of applications of the NN to NS multi-messenger astrophysics in several ways: we directly output the $p(\rho)$ EOS table, not limiting the output to selected EOS parameters, meaning that our implementation is, in principle, not bound to specific prescription of the EOS. In addition, we study the application of the AE architecture to the problem of EOS reconstruction, investigate as input the tidal deformability parameters as a function of mass, not only $M(R)$, that is, we try to simulate a situation in which the data comes exclusively from GW measurements, and we also investigate varying number of measurements, measurement errors, and realistic mass functions; for an additional investigation related to the last point, see the text below.

Motivated by the issue behind the significant increase of the reconstruction EOS errors for higher densities and pressures in cases of large measurement uncertainties ($m_{\text{err}} = 0.1 M_\odot$ and $r_{\text{err}} = 1$ km) we performed an additional analysis. We understand this as a feature of the non-linearity of the mapping between the observed values of M , R and \hat{R} and the EOS. As shown in Fig. 3, for example, the measurements at high masses probe a significantly larger range of pressure and densities than those at lower masses. In addition, the values of radii R and tidal deformabilities Λ (and hence \hat{R}) are typically smaller for larger masses: stars are more compact and also less prone to deformation. Sampling the measurements from the high-mass range, where the differences between measurements are small but the errors are comparable to the low-mass measurements, should result in worse reconstruction in the high pressure and density range of the EOS.

In order to study this further also from the point of view of the choice of mass function, we performed additional simulations. Since the double-Gaussian function we initially adopted has its main distribution peak in the low-mass range (around the Chandrasekhar mass), the majority of generated observation points correspond to lower pressures and densities, which are precisely reconstructed by the algorithm. However, the high mass, and therefore the high pressure and density range, is covered sparsely; hence the corresponding high pressures and densities may be reconstructed less precisely. To test this explanation, a new training data using alternative NS mass distribution were prepared. We considered a uniform mass distribution in the range between 1 and $2.2 M_\odot$. During the training on the uniform mass distribution data set, the ANN reached lower values of MSE with respect to the results presented in Sect. 4 with differences of around one order of magnitude in all considered cases. As a result the EOS reconstruction was characterised by smaller reconstruction errors for pressure and density; see examples of reconstruction in Fig. 7 for 20 observations with $\sigma_M = 0.1 M_\odot$ and $\sigma_R = 1$ km. Moreover, predicted values probed range of higher values with respect to results from Sect. 4.2. The uniform mass distribution allow to generate observations close to maximum value of $2.2 M_\odot$ (including measurement uncertainties), whereas the previously used double-Gaussian function returned masses rarely higher than $2 M_\odot$.

Our results suggest that to efficiently probe the high-mass end of the NS distribution, either measurement uncertainties should be significantly decreased with respect to the low-mass range or coverage of masses should be more uniform. The first possibility may be feasible with the 3rd generation GW detectors, such as the *Einstein Telescope* (Maggiore et al. 2020). On the other hand, the EOS is accurately reconstructed for the low-mass range (low pressure and density regime), which offers the possibility of comparing nuclear parameters with the data from terrestrial experiments.

It is also worth mentioning that a precise reconstruction of EOS using ANN requires training data that is representative of the problem. In order to reconstruct astrophysical EOS models (SLy4, APR, and BSK20), we have selected an appropriate training set. However, ANN tested on different EOS covering different ranges for M , R , Λ , p , and ρ would result in a worse reconstruction. To avoid this problem, it's necessary to optimise the parameter space of the training set and choose astrophysical models accordingly. It would be straightforward to expand the training dataset with a specific parametric description of dense matter, such as the MIT bag, to describe the deconfined quark matter (Chodos et al. 1974, see Sieniawska et al. 2019 for an example of piecewise relativistic polytrope EOS

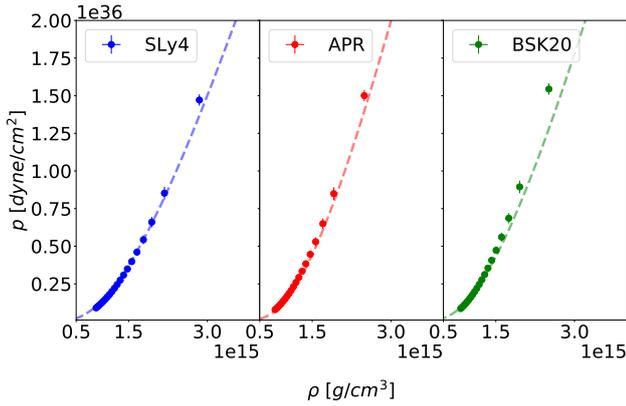


Fig. 7. ANN reconstructed EOS using the $M(R)$ data for the SLy4 EOS model (*left plot*), the APR EOS model (*middle plot*) and the BSK20 EOS model (*right plot*) for the uniform NS mass distribution. The presented results were computed for input data consisting of 20 observations with measurement uncertainties equal to $\sigma_M = 0.1 M_\odot$, $\sigma_R = 1$ km. Dashed lines correspond to exact values obtained by solving the TOV equations.

supplemented by quark EOS approximation of Zdunik 2000). In such a case, the ANN would potentially serve as a tool to discover the presence of exotic phases or signatures of dense-matter phase transitions.

6. Conclusions

We show that the ANN can be successfully applied in the reconstruction of the dense matter EOS from NS observations, either electromagnetic (masses and radii) or based on gravitational-wave measurements (masses and tidal deformabilities). We study the influence of the number of observations and the measurement uncertainties on the EOS reconstruction. The latter factor turned out to have a more significant effect on ANN performance, quantified in terms of the loss function (MSE). Furthermore, we show that the ANN trained on piecewise relativistic polytropes is capable of generalising the EOS reconstruction toward samples it wasn't previously exposed to: realistic EOSs resulting from microscopic calculations: the SLy4, APR, and BSK20 EOS models.

We also introduce reconstruction errors for ANN: σ_ρ and σ_p . The presented values vary proportionally to either the uncertainties of measurement with regard to the observables or to the values of pressures and densities. To decrease reconstruction errors, we suggest that either measurement uncertainties should be reduced, which is possible with the new generation of telescopes and detectors (i.e. *Einstein* Telescope for gravitational observations), or masses should be generated more uniformly. Moreover, we show that ANN can be successfully used in the reconstruction of radius based on the gravitational observables, which can be particularly useful for gravitational astronomy.

Among the many possibilities for further development in studies of NS parameters using ML methods, we plan to focus on the promising direction of variational auto-encoders. The latent space of these algorithms contain features that allow for an in-depth understanding of the distribution of parameters of the input data. Studies of the latent space could be used, for example, to infer information on the nuclear parameters of the EOS or assess the plausibility of the existence of a dense-matter phase transition.

Acknowledgements. The work was partially supported by the Polish National Science Centre grant no. 2016/22/E/ST9/00037 and the European Cooperation in Science and Technology COST action G2net no. CA17137. The Quadro P6000 used in this research was donated by the NVIDIA Corporation. This research was supported in part by PL-Grid Infrastructure (the Prometheus cluster).

References

- Abadi, M., Agarwal, A., Barham, P., et al. 2015, [TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems](https://arxiv.org/abs/1605.08877), Software available from: [tensorflow.org](https://www.tensorflow.org)
- Abbott, B. P., Abbott, R., Abbott, T. D., et al. 2017a, *Phys. Rev. Lett.*, **119**, 161101
- Abbott, B. P., Abbott, R., Abbott, T. D., et al. 2017b, *ApJ*, **848**, L13
- Abbott, B. P., Abbott, R., Abbott, T. D., et al. 2018, *Phys. Rev. Lett.*, **121**, 161101
- Abbott, B. P., Abbott, R., Abbott, T. D., et al. 2019, *Phys. Rev. X*, **9**, 011001
- Abbott, B. P., Abbott, R., Abbott, T. D., et al. 2020, *ApJ*, **892**, L3
- Akmal, A., Pandharipande, V. R., & Ravenhall, D. G. 1998, *Phys. Rev. C*, **58**, 1804
- Alsing, J., Silva, H. O., & Berti, E. 2018, *MNRAS*, **478**, 1377
- Antoniadis, J., Freire, P. C. C., Wex, N., et al. 2013, *Science*, **340**, 448
- Chetlur, S., Woolley, C., Vandermersch, P., et al. 2014, ArXiv e-prints [arXiv:1410.0759]
- Chodos, A., Jaffe, R. L., Johnson, K., Thorn, C. B., & Weisskopf, V. F. 1974, *Phys. Rev. D*, **9**, 3471
- Chollet, F., et al. 2015, Keras, <https://keras.io>
- Cromartie, H. T., Fonseca, E., Ransom, S. M., et al. 2020, *Nat. Astron.*, **4**, 72
- De, S., Finstad, D., Lattimer, J. M., et al. 2018, *Phys. Rev. Lett.*, **121**, 091102
- Demorest, P., Pennucci, T., Ransom, S., Roberts, M., & Hessels, J. 2010, *Nature*, **467**, 1081
- Douchin, F., & Haensel, P. 2001, *A&A*, **380**, 151
- Fasano, M., Abdelsalhin, T., Maselli, A., & Ferrari, V. 2019, *Phys. Rev. Lett.*, **123**, 141101
- Ferreira, M., & Providência, C. 2019, ArXiv e-prints [arXiv:1910.05554]
- Flanagan, E. E., & Hinderer, T. 2008, *Phys. Rev. D*, **77**, 021502
- Fonseca, E., Pennucci, T. T., Ellis, J. A., et al. 2016, *ApJ*, **832**, 167
- Fujimoto, Y., Fukushima, K., & Murase, K. 2018, *Phys. Rev. D*, **98**, 023019
- Fujimoto, Y., Fukushima, K., & Murase, K. 2020, *Phys. Rev. D*, **101**, 054016
- Goodfellow, I., Bengio, Y., & Courville, A. 2016, *Deep Learning* (The MIT Press)
- Goriely, S., Chamel, N., & Pearson, J. M. 2010, *Phys. Rev. C*, **82**, 035804
- Haegel, L., & Husa, S. 2019, *CQG*, **37**, 135005
- Haensel, P., & Pichon, B. 1994, *A&A*, **283**, 313
- Haensel, P., Potekhin, A. Y., & Yakovlev, D. G. 2007, *Neutron Stars 1: Equation of State and Structure* (New York, USA: Springer), 326, 1
- Hernandez Vivanco, F., Smith, R., Thrane, E., et al. 2019, *Phys. Rev. D*, **100**, 103009
- Hinton, G. E., & Zemel, R. S. 1993, *Proceedings of the 6th International Conference on Neural Information Processing Systems, NIPS'93* (San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.), 3
- Holt, J. W., & Lim, Y. 2019, *AIP Conf. Proc.*, **2127**, 020019
- Kingma, D. P., & Ba, J. 2014, ArXiv e-prints [arXiv:1412.6980]
- Kramer, M. A. 1991, *AIChe J.*, **37**, 233
- Love, A. E. H. 1911, *Some Problems of Geodynamics* (Cambridge Univ. Press)
- Maggiore, M., Van Den Broeck, C., Bartolo, N., et al. 2020, *J. Cosmol. Astropart. Phys.*, **2020**, 050
- Miller, M. C., Lamb, F. K., Dittmann, A. J., et al. 2019, *ApJ*, **887**, L24
- Nickolls, J., Buck, I., Garland, M., & Skadron, K. 2008, *Queue*, **6**, 40
- Oppenheimer, J. R., & Volkoff, G. M. 1939, *Phys. Rev.*, **55**, 374
- Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. 1992, *Numerical recipes in FORTRAN. The art of scientific computing*, 2nd edn. (Cambridge University Press)
- Raitel, C. A., Özel, F., & Psaltis, D. 2016, *ApJ*, **831**, 44
- Riley, T. E., Watts, A. L., Bogdanov, S., et al. 2019, *ApJ*, **887**, L21
- Samuel, A. L. 1959, *IBM J. Res. Dev.*, **3**, 210
- Sieniawska, M., Turczański, W., Bejger, M., & Zdunik, J. L. 2019, *A&A*, **622**, A174
- Steiner, A. W., Lattimer, J. M., & Brown, E. F. 2010, *ApJ*, **722**, 33
- Steiner, A. W., Lattimer, J. M., & Brown, E. F. 2013, *ApJ*, **765**, L5
- Tolman, R. C. 1939, *Phys. Rev.*, **55**, 364
- Tooper, R. F. 1965, *ApJ*, **142**, 1541
- Traversi, S., Char, P., & Pagliara, G. 2020, *ApJ*, **897**, 165
- Van Oeveren, E. D., & Friedman, J. L. 2017, *Phys. Rev. D*, **95**, 083014
- Wade, L., Creighton, J. D. E., Ochsner, E., et al. 2014, *Phys. Rev. D*, **89**, 103012
- Zdunik, J. L. 2000, *A&A*, **359**, 311
- Zhang, N.-B., Qi, B., & Wang, S.-Y. 2020, *Chinese Phys. C*, **44**, 064103

Part V

What's next?

Chapter 6

Summary and outlook

I attempted to demonstrate the versatility and enormous benefits of using ML methods in GW astronomy from two perspectives in my thesis. Specifically, *how* those methods can be used and *what* we can learn from them. These objectives, I believe, were met. I demonstrated that ANNs are an excellent tool for searching for various GW signal types in a noisy environment by utilizing different data representations and ML models architectures. By the most important conclusion of my thesis, I perceive the detection capabilities of the studied ANNs. They were able to not only detect but also distinguish relatively weak GW signals from detector artifacts. The methods' capabilities of correct detection improve the overall quality of the data and what we can infer later from detected GWs. For example, by using the convolutional AE described in paper I, I was able to partially reconstruct signals and precisely localize them in time in addition to searching for anomalies. As a result, these extra features may help with parameter estimation and sky localization of the emission source, both of which are important in our understanding of astrophysical objects.

The research presented in this thesis contributes to the growing field of extensive studies on the applications of ML in the analysis of data from GW detectors. The importance of this research becomes clear when viewed in the context of large-scale scientific projects and science in general. Current and planned experiments such as the Cherenkov Telescope Array, Square Kilometer Array, Large Synoptic Sky Survey, ET, and LISA all share one feature: Big Data. All of them will generate massive amounts of data that will be inefficient (or, in some cases, impossible) to store on hard drives. These large-scale projects require cutting-edge software capable of real-time analysis to extract scientifically interesting information from raw data that scientists can later work on. With this requirement comes a significant challenge: how can our programs determine which data is more relevant than others, especially in the case of unknown phenomena?

At least some of these issues appear to be addressed by ML and AI. They are extremely efficient, capable of detecting even complex patterns in data as well as able to infer some astrophysical information. Extensive research on anomaly detection may also aid in the extraction of unknown patterns from data that indicate unknown phenomena worth studying.

Several collaborations associated with the mentioned projects have already recognized the importance of ML. Dedicated ML working groups are formed to develop customized,

sophisticated data analysis methods. Furthermore, large international initiatives such as the European Open Science Cloud or The European Science Cluster for Astronomy and Particle Physics are examples of efforts addressing the critical aspects of challenges confronting modern scientific experiments, with AI serving as a primary tool in addressing them.

Bibliography

- [1] Albert Einstein. “The Foundation of the General Theory of Relativity”. In: *Annalen Phys.* 49.7 (1916). Ed. by Jong-Ping Hsu and D. Fine, pp. 769–822. DOI: [10.1002/andp.200590044](https://doi.org/10.1002/andp.200590044) (pages 3, 4).
- [2] A. Einstein. “Näherungsweise Integration der Feldgleichungen der Gravitation”. In: *Sitzungsberichte der Königlich Preussischen Akademie der Wissenschaften (Berlin), Seite 688-696.* (1916) (pages 3, 4).
- [3] B. P. Abbott, R. Abbott, T. D. Abbott, M. R. Abernathy, et al. “Observation of Gravitational Waves from a Binary Black Hole Merger”. In: *Phys. Rev. Lett.* 116.6, 061102 (2016), p. 061102. DOI: [10.1103/PhysRevLett.116.061102](https://doi.org/10.1103/PhysRevLett.116.061102). arXiv: [1602.03837](https://arxiv.org/abs/1602.03837) (pages 3, 8, 13, 14).
- [4] Michele Maggiore. *Gravitational Waves. Vol. 1: Theory and Experiments.* Oxford Master Series in Physics. Oxford University Press, 2007 (page 4).
- [5] Éanna É Flanagan and Scott A Hughes. “The basics of gravitational wave theory”. In: *New Journal of Physics* 7 (Sept. 2005), pp. 204–204. DOI: [10.1088/1367-2630/7/1/204](https://doi.org/10.1088/1367-2630/7/1/204) (page 5).
- [6] Richard Price and Yan Wang. “The transverse traceless gauge and quadrupole sources”. In: *American Journal of Physics - AMER J PHYS* 76 (Oct. 2008), pp. 930–933. DOI: [10.1119/1.2937908](https://doi.org/10.1119/1.2937908) (page 5).
- [7] Albert Einstein. “Über Gravitationswellen”. In: *Sitzungsber. Preuss. Akad. Wiss. Berlin (Math. Phys.)* 1918 (1918), pp. 154–167 (page 6).
- [8] George David Birkhoff and Rudolph Ernest Langer. *Relativity and modern physics.* 1923 (page 6).
- [9] C. Brans and R. H. Dicke. “Mach’s Principle and a Relativistic Theory of Gravitation”. In: *Phys. Rev.* 124 (3 Nov. 1961), pp. 925–935. DOI: [10.1103/PhysRev.124.925](https://doi.org/10.1103/PhysRev.124.925) (page 6).
- [10] Kostas D. Kokkotas. “Gravitational waves”. In: *Acta Phys. Polon. B* 38 (2007). Ed. by Michal Praszalowicz, Marek Kutschera, and Edward Malec, pp. 3891–3923 (page 6).
- [11] B. P. Abbott, R. Abbott, T. D. Abbott, F. Acernese, K. Ackley, C. Adams, T. Adams, P. Addesso, R. X. Adhikari, V. B. Adya, and et al. “Gravitational Waves and Gamma-Rays from a Binary Neutron Star Merger: GW170817 and GRB 170817A”. In: *The Astrophysical Journal* 848.2 (Oct. 2017), p. L13. DOI: [10.3847/2041-8213/aa920c](https://doi.org/10.3847/2041-8213/aa920c) (page 6).
- [12] James William Rohlf. *Modern Physics from A to Z.* New York: John Wiley and Sons, 1994 (page 6).
- [13] F. A. E. Pirani. “On the Physical significance of the Riemann tensor”. In: *Acta Physica Polonica* 15 (Jan. 1956), pp. 389–405 (page 7).
- [14] T. N. Ukwatta, P. R. Woźniak, and N. Gehrels. “Machine-z: rapid machine-learned redshift indicator for Swift gamma-ray bursts”. In: *Monthly Notices of the Royal Astronomical Society* 458.4 (Mar. 2016), pp. 3821–3829. DOI: [10.1093/mnras/stw559](https://doi.org/10.1093/mnras/stw559). eprint: <https://academic.oup.com/mnras/article-pdf/458/4/3821/13455388/stw559.pdf> (page 7).
- [15] Mark Zimmermann and Eugene Szedenits. “Gravitational waves from rotating and precessing rigid bodies: Simple models and applications to pulsars”. In: *Phys. Rev. D* 20 (2 July 1979), pp. 351–355. DOI: [10.1103/PhysRevD.20.351](https://doi.org/10.1103/PhysRevD.20.351) (page 7).
- [16] S. Bonazzola and É. Gourgoulhon. “Gravitational waves from pulsars: Emission by the magnetic field induced distortion”. In: *Astronomy and Astrophysics* 312 (1995), pp. 675–690 (page 7).
- [17] Piotr Jaranowski and Andrzej Królak. “Analysis of Gravitational-Wave Data”. In: *Analysis of Gravitational-Wave Data, by Piotr Jaranowski, Andrzej Krolak, Cambridge, UK: Cambridge University Press, 2009* (Aug. 2009). DOI: [10.1017/CB09780511605482](https://doi.org/10.1017/CB09780511605482) (pages 7, 8).
- [18] B. Paczyński. “Gravitational Waves and the Evolution of Close Binaries”. In: *Acta Astronomica* 17 (Jan. 1967), p. 287 (pages 8, 13).

- [19] R. A. Hulse and J. H. Taylor. “Discovery of a pulsar in a binary system.” In: *Ap. J. L.* 195 (Jan. 1975), pp. L51–L53. DOI: [10.1086/181708](https://doi.org/10.1086/181708) (pages 8, 13).
- [20] J. M. Weisberg, J. H. Taylor, and L. A. Fowler. “Gravitational waves from an orbiting pulsar”. In: *Scientific American* 245 (Oct. 1981), pp. 74–82. DOI: [10.1038/scientificamerican1081-74](https://doi.org/10.1038/scientificamerican1081-74) (page 8).
- [21] J. Weber. “Gravitational Radiation”. In: *Phys. Rev. Lett.* 18 (13 Mar. 1967), pp. 498–501. DOI: [10.1103/PhysRevLett.18.498](https://doi.org/10.1103/PhysRevLett.18.498) (page 8).
- [22] M. Cerdonio et al. “The Ultracryogenic gravitational wave detector AURIGA”. In: *Class. Quant. Grav.* 14 (1997), pp. 1491–1494. DOI: [10.1088/0264-9381/14/6/016](https://doi.org/10.1088/0264-9381/14/6/016) (page 8).
- [23] E. Mauceli, Z. K. Geng, W. O. Hamilton, W. W. Johnson, S. Merkwitz, A. Morse, B. Price, and N. Solomonson. “The Allegro gravitational wave detector: Data acquisition and analysis”. In: *Physical Review D* 54.2 (July 1996), pp. 1264–1275. DOI: [10.1103/physrevd.54.1264](https://doi.org/10.1103/physrevd.54.1264) (page 8).
- [24] F. Ronga et al. “Detection of high energy cosmic rays with the resonant gravitational wave detectors NAUTILUS and EXPLORER and comparison with the direct measurements with an aluminum superconductive bar”. In: *Nucl. Phys. B Proc. Suppl.* 190 (2009). Ed. by R. Caruso, A. Insolia, A. Letessier-Selvon, and V. Pirronello, pp. 44–51. DOI: [10.1016/j.nuclphysbps.2009.03.067](https://doi.org/10.1016/j.nuclphysbps.2009.03.067) (page 8).
- [25] Odylio Denys Aguiar. “Past, present and future of the Resonant-Mass gravitational wave detectors”. In: *Research in Astronomy and Astrophysics* 11.1 (Dec. 2010), pp. 1–42. DOI: [10.1088/1674-4527/11/1/001](https://doi.org/10.1088/1674-4527/11/1/001) (page 8).
- [26] J. Aasi et al. “Advanced LIGO”. In: *Class. Quant. Grav.* 32 (2015), p. 074001. DOI: [10.1088/0264-9381/32/7/074001](https://doi.org/10.1088/0264-9381/32/7/074001). arXiv: [1411.4547](https://arxiv.org/abs/1411.4547) [gr-qc] (pages 8, 11, 12).
- [27] F. Acernese, M. Agathos, K. Agatsuma, D. Aisa, et al. “Advanced Virgo: a second-generation interferometric gravitational wave detector”. In: *Class. Quant. Grav.* 32.2, 024001 (2015), p. 024001. DOI: [10.1088/0264-9381/32/2/024001](https://doi.org/10.1088/0264-9381/32/2/024001). arXiv: [1408.3978](https://arxiv.org/abs/1408.3978) (pages 8, 11, 12).
- [28] A. Fölsing, E. Osers, and Viking Press. *Albert Einstein: A Biography*. Viking, 1997 (page 9).
- [29] Rajpal Sirohi. “Interferometry: From Hooke till Date”. In: Sept. 2017, pp. 65–71. DOI: [10.1007/978-981-10-3908-9_8](https://doi.org/10.1007/978-981-10-3908-9_8) (page 9).
- [30] Wikipedia contributors. *Concept of LIGO: A schematic diagram of a laser interferometer with light storage arms*. [Online; accessed 8-October-2021]. 2012 (page 9).
- [31] A. Perot and Charles Fabry. “On the Application of Interference Phenomena to the Solution of Various Problems of Spectroscopy and Metrology”. In: *The Astrophysical Journal* 9 (Feb. 1899), p. 87. DOI: [10.1086/140557](https://doi.org/10.1086/140557) (page 10).
- [32] B. P. Abbott et al. “A guide to LIGO–Virgo detector noise and extraction of transient gravitational-wave signals”. In: *Class. Quant. Grav.* 37.5 (2020), p. 055002. DOI: [10.1088/1361-6382/ab685e](https://doi.org/10.1088/1361-6382/ab685e). arXiv: [1908.11170](https://arxiv.org/abs/1908.11170) [gr-qc] (page 10).
- [33] M. W. Regehr, F. J. Raab, and S. E. Whitcomb. “Demonstration of a power-recycled Michelson interferometer with Fabry–Perot arms by frontal modulation”. In: *Opt. Lett.* 20.13 (July 1995), pp. 1507–1509. DOI: [10.1364/OL.20.001507](https://doi.org/10.1364/OL.20.001507) (page 11).
- [34] Shuichi Sato, Masatake Ohashi, Masa-Katsu Fujimoto, Mitsuhiro Fukushima, Koichi Waseda, Shinji Miyoki, Nergis Mavalvala, and Hiroaki Yamamoto. “High-gain power recycling of a Fabry–Perot Michelson interferometer for a gravitational-wave antenna”. In: *Appl. Opt.* 39.25 (Sept. 2000), pp. 4616–4620. DOI: [10.1364/AO.39.004616](https://doi.org/10.1364/AO.39.004616) (page 11).
- [35] C J Moore, R H Cole, and C P L Berry. “Gravitational-wave sensitivity curves”. In: *Classical and Quantum Gravity* 32.1 (Dec. 2014), p. 015014. DOI: [10.1088/0264-9381/32/1/015014](https://doi.org/10.1088/0264-9381/32/1/015014) (page 11).
- [36] T. Akutsu et al. “KAGRA: 2.5 Generation Interferometric Gravitational Wave Detector”. In: *Nature Astron.* 3.1 (2019), pp. 35–40. DOI: [10.1038/s41550-018-0658-y](https://doi.org/10.1038/s41550-018-0658-y). arXiv: [1811.08079](https://arxiv.org/abs/1811.08079) [gr-qc] (page 12).
- [37] Benno Willke et al. “The GEO 600 gravitational wave detector”. In: *Classical and Quantum Gravity* 19 (Mar. 2002). DOI: [10.1088/0264-9381/19/7/321](https://doi.org/10.1088/0264-9381/19/7/321) (page 12).
- [38] Iyer B et al. *LIGO-India, proposal of the consortium for Indian initiative in gravitational-wave observations (IndIGO)*. <http://web.archive.org/web/20080207010024>. Accessed: 2010-09-30. 2011 (page 12).
- [39] K. Ackley, V. B. Adya, P. Agrawal, P. Altin, G. Ashton, M. Bailes, E. Baltinas, A. Barbuio, D. Beniwal, C. Blair, and et al. “Neutron Star Extreme Matter Observatory: A kilohertz-band gravitational-wave detector in the global network”. In: *Publications of the Astronomical Society of Australia* 37 (2020). DOI: [10.1017/pasa.2020.39](https://doi.org/10.1017/pasa.2020.39) (page 12).

- [40] R Adhikari et al. “LIGO III Blue ConceptLIGO”. In: *Technical Document LIGO-G1200573*. See dcc.ligo.org/LIGO-G1200573-v1/public (2012) (page 12).
- [41] David Reitze, Rana X Adhikari, Stefan Ballmer, Barry Barish, Lisa Barsotti, GariLynn Billingsley, Duncan A. Brown, Yanbei Chen, Dennis Coyne, Robert Eisenstein, Matthew Evans, Peter Fritschel, Evan D. Hall, Albert Lazzarini, Geoffrey Lovelace, Jocelyn Read, B. S. Sathyaprakash, David Shoemaker, Joshua Smith, Calum Torrie, Salvatore Vitale, Rainer Weiss, Christopher Wipf, and Michael Zucker. *Cosmic Explorer: The U.S. Contribution to Gravitational-Wave Astronomy beyond LIGO*. 2019. arXiv: [1907.04833](https://arxiv.org/abs/1907.04833) [[astro-ph.IM](#)] (page 12).
- [42] M. Punturo et al. “The Einstein Telescope: A third-generation gravitational wave observatory”. In: *Class. Quant. Grav.* 27 (2010). Ed. by Fulvio Ricci, p. 194002. DOI: [10.1088/0264-9381/27/19/194002](https://doi.org/10.1088/0264-9381/27/19/194002) (page 12).
- [43] Pau Amaro-Seoane et al. “Laser Interferometer Space Antenna”. In: *arXiv e-prints*, arXiv:1702.00786 (Feb. 2017), arXiv:1702.00786. arXiv: [1702.00786](https://arxiv.org/abs/1702.00786) [[astro-ph.IM](#)] (page 12).
- [44] Wen-Cong Chen, Dong-Dong Liu, and Bo Wang. “Detectability of Ultra-compact X-Ray Binaries as LISA Sources”. In: *The Astrophysical Journal* 900.1 (Aug. 2020), p. L8. DOI: [10.3847/2041-8213/abae66](https://doi.org/10.3847/2041-8213/abae66) (page 13).
- [45] Kip S. Thorne. “Gravitational waves”. In: *1994 DPF Summer Study on High-energy Physics: Particle and Nuclear Astrophysics and Cosmology in the Next Millennium (Snowmass 94)*. June 1995, pp. 0160–184. arXiv: [gr-qc/9506086](https://arxiv.org/abs/gr-qc/9506086) (page 13).
- [46] Bernard Schutz. “Low-Frequency Sources of Gravitational Waves: A Tutorial”. In: *Fundamental physics in space : proceedings of the Alpbach Summer School 1997, ESA Publications Division, 229-240 (1997)* (Oct. 1997) (page 13).
- [47] B. P. Abbott, R. Abbott, T. D. Abbott, M. R. Abernathy, F. Acernese, K. Ackley, C. Adams, T. Adams, P. Addesso, and et al. “The basic physics of the binary black hole merger GW150914”. In: *Annalen der Physik* 529.1-2 (Oct. 2016), p. 1600209. DOI: [10.1002/andp.201600209](https://doi.org/10.1002/andp.201600209) (page 13).
- [48] Matthew Pitkin, Stuart Reid, Sheila Rowan, and Jim Hough. “Gravitational Wave Detection by Interferometry (Ground and Space)”. In: *Living Reviews in Relativity* 14.1 (July 2011). DOI: [10.12942/lrr-2011-5](https://doi.org/10.12942/lrr-2011-5) (page 13).
- [49] B. P. Abbott, R. Abbott, T. D. Abbott, S. Abraham, F. Acernese, K. Ackley, C. Adams, R. X. Adhikari, V. B. Adya, C. Affeldt, and et al. “GWTC-1: A Gravitational-Wave Transient Catalog of Compact Binary Mergers Observed by LIGO and Virgo during the First and Second Observing Runs”. In: *Physical Review X* 9.3 (Sept. 2019). DOI: [10.1103/physrevx.9.031040](https://doi.org/10.1103/physrevx.9.031040) (page 13).
- [50] R. Abbott, T. D. Abbott, S. Abraham, F. Acernese, K. Ackley, A. Adams, C. Adams, R. X. Adhikari, V. B. Adya, C. Affeldt, and et al. “GWTC-2: Compact Binary Coalescences Observed by LIGO and Virgo during the First Half of the Third Observing Run”. In: *Physical Review X* 11.2 (June 2021). DOI: [10.1103/physrevx.11.021053](https://doi.org/10.1103/physrevx.11.021053) (page 13).
- [51] R. Abbott, T. D. Abbott, S. Abraham, F. Acernese, K. Ackley, A. Adams, C. Adams, R. X. Adhikari, V. B. Adya, C. Affeldt, and et al. “Observation of Gravitational Waves from Two Neutron Star–Black Hole Coalescences”. In: *The Astrophysical Journal Letters* 915.1 (June 2021), p. L5. DOI: [10.3847/2041-8213/ac082e](https://doi.org/10.3847/2041-8213/ac082e) (page 13).
- [52] B. P. Abbott et al. “GW170817: Observation of Gravitational Waves from a Binary Neutron Star Inspiral”. In: *Phys. Rev. Lett.* 119.16 (2017), p. 161101. DOI: [10.1103/PhysRevLett.119.161101](https://doi.org/10.1103/PhysRevLett.119.161101). arXiv: [1710.05832](https://arxiv.org/abs/1710.05832) [[gr-qc](#)] (pages 13, 14).
- [53] B. P. Abbott et al. “First targeted search for gravitational-wave bursts from core-collapse supernovae in data of first-generation laser interferometer detectors”. In: *Phys. Rev. D* 94 (10 Nov. 2016), p. 102001. DOI: [10.1103/PhysRevD.94.102001](https://doi.org/10.1103/PhysRevD.94.102001) (page 14).
- [54] B. P. Abbott, R. Abbott, T. D. Abbott, S. Abraham, F. Acernese, K. Ackley, C. Adams, V. B. Adya, C. Affeldt, M. Agathos, and et al. “Optically targeted search for gravitational waves emitted by core-collapse supernovae during the first and second observing runs of advanced LIGO and advanced Virgo”. In: *Physical Review D* 101.8 (Apr. 2020). DOI: [10.1103/physrevd.101.084002](https://doi.org/10.1103/physrevd.101.084002) (page 14).
- [55] Marek Szczepanczyk, Javier Antelis, Michael Benjamin, Marco Cavaglia, Dorota Gondek-Rosinska, Travis Hansen, Sergey Klimenko, Manuel Morales, Claudia Moreno, Soma Mukherjee, Gaukhar Nurbek, Jade Powell, Neha Singh, Satzhan Sitmukhambetov, Pawel Szewczyk, Jonathan Westhouse, Oscar Valdez, Gabriele Vedovato, Yanyan Zheng, and Michele Zanolin. *Detecting and reconstructing gravitational waves from the next Galactic core-collapse supernova in the Advanced Detector Era*. 2021. arXiv: [2104.06462](https://arxiv.org/abs/2104.06462) [[astro-ph.HE](#)] (page 14).

- [56] Sidney van den Bergh and Gustav A. Tammann. “Galactic and Extragalactic Supernova Rates”. In: *Annual Review of Astronomy and Astrophysics* 29.1 (1991), pp. 363–407. DOI: [10.1146/annurev.aa.29.090191.002051](https://doi.org/10.1146/annurev.aa.29.090191.002051). eprint: <https://doi.org/10.1146/annurev.aa.29.090191.002051> (page 14).
- [57] E. N. Alexeyev and L. N. Alexeyeva. “Twenty years of Galactic observations in searching for bursts of collapse neutrinos with the Baksan underground scintillation telescope”. In: *Journal of Experimental and Theoretical Physics* 95.1 (July 2002), pp. 5–10. DOI: [10.1134/1.1499896](https://doi.org/10.1134/1.1499896) (page 14).
- [58] Kei Kotake, Katsuhiko Sato, and Keitaro Takahashi. “Explosion mechanism, neutrino burst and gravitational wave in core-collapse supernovae”. In: *Reports on Progress in Physics* 69.4 (Mar. 2006), pp. 971–1143. DOI: [10.1088/0034-4885/69/4/r03](https://doi.org/10.1088/0034-4885/69/4/r03) (page 14).
- [59] Brendan Reed and C. J. Horowitz. “Total energy in supernova neutrinos and the tidal deformability and binding energy of neutron stars”. In: *Phys. Rev. D* 102 (10 Nov. 2020), p. 103011. DOI: [10.1103/PhysRevD.102.103011](https://doi.org/10.1103/PhysRevD.102.103011) (page 14).
- [60] Jade Powell and Bernhard Müller. “Gravitational wave emission from 3D explosion models of core-collapse supernovae with low and normal explosion energies”. In: *Monthly Notices of the Royal Astronomical Society* 487.1 (May 2019), pp. 1178–1190. DOI: [10.1093/mnras/stz1304](https://doi.org/10.1093/mnras/stz1304) (page 14).
- [61] David Radice, Viktoriya Morozova, Adam Burrows, David Vartanyan, and Hiroki Nagakura. “Characterizing the Gravitational Wave Signal from Core-collapse Supernovae”. In: *The Astrophysical Journal* 876.1 (Apr. 2019), p. L9. DOI: [10.3847/2041-8213/ab191a](https://doi.org/10.3847/2041-8213/ab191a) (page 14).
- [62] H Andresen, R Glas, and H-Th Janka. “Gravitational-wave signals from 3D supernova simulations with different neutrino-transport methods”. In: *Monthly Notices of the Royal Astronomical Society* 503.3 (Mar. 2021), pp. 3552–3567. DOI: [10.1093/mnras/stab675](https://doi.org/10.1093/mnras/stab675). eprint: <https://academic.oup.com/mnras/article-pdf/503/3/3552/36846778/stab675.pdf> (page 14).
- [63] Bernhard Müller. *Gravitational Waves from Core-Collapse Supernovae*. 2017. arXiv: [1703.04633](https://arxiv.org/abs/1703.04633) [astro-ph.HE] (page 14).
- [64] John M. Blondin, Anthony Mezzacappa, and Christine DeMarino. “Stability of Standing Accretion Shocks, with an Eye toward Core-Collapse Supernovae”. In: *The Astrophysical Journal* 584.2 (Feb. 2003), pp. 971–980. DOI: [10.1086/345812](https://doi.org/10.1086/345812) (page 15).
- [65] Magdalena Sieniawska and Michał Bejger. “Continuous Gravitational Waves from Neutron Stars: Current Status and Prospects”. In: *Universe* 5.11 (Oct. 2019), p. 217. DOI: [10.3390/universe5110217](https://doi.org/10.3390/universe5110217). arXiv: [1909.12600](https://arxiv.org/abs/1909.12600) [astro-ph.HE] (page 15).
- [66] Piotr Jaranowski, Andrzej Królak, and Bernard F. Schutz. “Data analysis of gravitational-wave signals from spinning neutron stars: The signal and its detection”. In: *Phys. Rev. D* 58 (6 Aug. 1998), p. 063001. DOI: [10.1103/PhysRevD.58.063001](https://doi.org/10.1103/PhysRevD.58.063001) (page 15).
- [67] B. Haskell. “R-modes in neutron stars: Theory and observations”. In: *International Journal of Modern Physics E* 24.09 (Sept. 2015), p. 1541007. DOI: [10.1142/s0218301315410074](https://doi.org/10.1142/s0218301315410074) (page 15).
- [68] Marek Cieřlar, Tomasz Bulik, Małgorzata Curyło, Magdalena Sieniawska, Neha Singh, and Michał Bejger. “Detectability of continuous gravitational waves from isolated neutron stars in the Milky Way”. In: *Astronomy & Astrophysics* 649 (May 2021), A92. DOI: [10.1051/0004-6361/202039503](https://doi.org/10.1051/0004-6361/202039503) (page 15).
- [69] Hannah Middleton, Patrick Clearwater, Andrew Melatos, and Liam Dunn. “Search for gravitational waves from five low mass x-ray binaries in the second Advanced LIGO observing run with an improved hidden Markov model”. In: *Phys. Rev. D* 102 (2 July 2020), p. 023006. DOI: [10.1103/PhysRevD.102.023006](https://doi.org/10.1103/PhysRevD.102.023006) (page 15).
- [70] Lars Bildsten. “Gravitational Radiation and Rotation of Accreting Neutron Stars”. In: *The Astrophysical Journal* 501.1 (July 1998), pp. L89–L93. DOI: [10.1086/311440](https://doi.org/10.1086/311440) (page 15).
- [71] F Gittins, N Andersson, and D I Jones. “Modelling neutron star mountains”. In: *Monthly Notices of the Royal Astronomical Society* 500.4 (Nov. 2020), pp. 5570–5582. DOI: [10.1093/mnras/staa3635](https://doi.org/10.1093/mnras/staa3635) (page 15).
- [72] P. V. S. Pavan Chandra, Mrunal Korwar, and Arun M. Thalappilil. “Continuous gravitational waves and magnetic monopole signatures from single neutron stars”. In: *Phys. Rev. D* 101 (7 Apr. 2020), p. 075028. DOI: [10.1103/PhysRevD.101.075028](https://doi.org/10.1103/PhysRevD.101.075028) (page 15).
- [73] Nelson Christensen. “Stochastic gravitational wave backgrounds”. In: *Reports on Progress in Physics* 82.1 (Nov. 2018), p. 016903. DOI: [10.1088/1361-6633/aae6b5](https://doi.org/10.1088/1361-6633/aae6b5) (page 16).
- [74] Ruth Durrer. *The Cosmic Microwave Background*. 2nd ed. Cambridge University Press, 2020. DOI: [10.1017/9781316471524](https://doi.org/10.1017/9781316471524) (page 16).

- [75] B. P. Abbott, R. Abbott, T. D. Abbott, S. Abraham, F. Acernese, K. Ackley, C. Adams, R. X. Adhikari, V. B. Adya, C. Affeldt, and et al. “Search for Transient Gravitational-wave Signals Associated with Magnetar Bursts during Advanced LIGO’s Second Observing Run”. In: *The Astrophysical Journal* 874.2 (Apr. 2019), p. 163. DOI: [10.3847/1538-4357/ab0e15](https://doi.org/10.3847/1538-4357/ab0e15) (page 16).
- [76] Noah Pearson, Cynthia Tendafileva, and Joel Meyers. “Searching for gravitational waves with strongly lensed repeating fast radio bursts”. In: *Phys. Rev. D* 103 (6 Mar. 2021), p. 063017. DOI: [10.1103/PhysRevD.103.063017](https://doi.org/10.1103/PhysRevD.103.063017) (page 16).
- [77] Saeed Vaseghi. “Wiener Filters”. In: (Sept. 2001), pp. 178–204. DOI: [10.1002/0470841621.ch6](https://doi.org/10.1002/0470841621.ch6) (page 17).
- [78] Tito Dal Canton and Ian W. Harry. *Designing a template bank to observe compact binary coalescences in Advanced LIGO’s second observing run*. 2017. arXiv: [1705.01845](https://arxiv.org/abs/1705.01845) [gr-qc] (page 17).
- [79] Tito Dal Canton, Alexander H. Nitz, Andrew P. Lundgren, Alex B. Nielsen, Duncan A. Brown, Thomas Dent, Ian W. Harry, Badri Krishnan, Andrew J. Miller, Karl Wette, Karsten Wiesner, and Joshua L. Willis. “Implementing a search for aligned-spin neutron star-black hole systems with advanced ground based gravitational wave detectors”. In: *Phys. Rev. D* 90 (8 Oct. 2014), p. 082004. DOI: [10.1103/PhysRevD.90.082004](https://doi.org/10.1103/PhysRevD.90.082004) (page 17).
- [80] S. A. Usman et al. “The PyCBC search for gravitational waves from compact binary coalescence”. In: *Classical and Quantum Gravity* 33.21, 215004 (Nov. 2016), p. 215004. DOI: [10.1088/0264-9381/33/21/215004](https://doi.org/10.1088/0264-9381/33/21/215004). arXiv: [1508.02357](https://arxiv.org/abs/1508.02357) [gr-qc] (page 17).
- [81] Surabhi Sachdev et al. “The GstLAL Search Analysis Methods for Compact Binary Mergers in Advanced LIGO’s Second and Advanced Virgo’s First Observing Runs”. In: (2019). arXiv: [1901.08580](https://arxiv.org/abs/1901.08580) [gr-qc] (page 17).
- [82] *Time-domain \mathcal{F} -statistic pipeline repository*. <https://github.com/mbejger/polgraw-allsky>. Accessed: 2018-09-04 (page 17).
- [83] The LIGO Scientific Collaboration et al. *All-sky Search for Continuous Gravitational Waves from Isolated Neutron Stars in the Early O3 LIGO Data*. 2021. arXiv: [2107.00600](https://arxiv.org/abs/2107.00600) [gr-qc] (page 17).
- [84] S. Klimenko et al. “Method for detection and reconstruction of gravitational wave transients with networks of advanced detectors”. In: *Phys. Rev. D* 93.4, 042004 (Feb. 2016), p. 042004. DOI: [10.1103/PhysRevD.93.042004](https://doi.org/10.1103/PhysRevD.93.042004). arXiv: [1511.05999](https://arxiv.org/abs/1511.05999) [gr-qc] (page 17).
- [85] S Klimenko et al. “A coherent method for detection of gravitational wave bursts”. In: *Classical and Quantum Gravity* 25.11 (2008), p. 114029 (page 17).
- [86] Yves Meyer and D. H. Salinger. *Wavelets and operators*. Cambridge University Press, 1993 (page 17).
- [87] M Cavaglià, S Gaudio, T Hansen, K Staats, M Szczepańczyk, and M Zanolin. “Improving the background of gravitational-wave searches for core collapse supernovae: a machine learning approach”. In: *Machine Learning: Science and Technology* 1.1 (Feb. 2020), p. 015005. DOI: [10.1088/2632-2153/ab527d](https://doi.org/10.1088/2632-2153/ab527d) (page 17).
- [88] Bruce Allen and Joseph D. Romano. “Detecting a stochastic background of gravitational radiation: Signal processing strategies and sensitivities”. In: *Physical Review D* 59.10 (Mar. 1999). DOI: [10.1103/physrevd.59.102001](https://doi.org/10.1103/physrevd.59.102001) (page 17).
- [89] Sanjit Mitra et al. “Gravitational wave radiometry: Mapping a stochastic gravitational wave background”. In: *Physical Review D* 77.4 (Feb. 2008). DOI: [10.1103/physrevd.77.042002](https://doi.org/10.1103/physrevd.77.042002) (page 17).
- [90] John T. Whelan, Santosh Sundareshan, Yuanhao Zhang, and Prabath Peiris. “Model-based cross-correlation search for gravitational waves from Scorpius X-1”. In: *Physical Review D* 91.10 (May 2015). DOI: [10.1103/physrevd.91.102005](https://doi.org/10.1103/physrevd.91.102005) (page 17).
- [91] L. K. Nuttall. “Characterizing transient noise in the LIGO detectors”. In: *Phil. Trans. Roy. Soc. Lond. A* 376.2120 (2018), p. 20170286. DOI: [10.1098/rsta.2017.0286](https://doi.org/10.1098/rsta.2017.0286). arXiv: [1804.07592](https://arxiv.org/abs/1804.07592) [astro-ph.IM] (page 18).
- [92] M Cabero, A Lundgren, A H Nitz, T Dent, D Barker, E Goetz, J S Kissel, L K Nuttall, P Schale, R Schofield, and D Davis. “Blip glitches in Advanced LIGO data”. In: *Classical and Quantum Gravity* 36.15 (July 2019), p. 155010. DOI: [10.1088/1361-6382/ab2e14](https://doi.org/10.1088/1361-6382/ab2e14) (page 18).
- [93] S Soni et al. *Discovering features in gravitational-wave data through detector characterization, citizen science and machine learning*. 2021. arXiv: [2103.12104](https://arxiv.org/abs/2103.12104) [gr-qc] (page 18).
- [94] D Davis et al. “LIGO detector characterization in the second and third observing runs”. In: *Classical and Quantum Gravity* 38.13 (June 2021), p. 135014. DOI: [10.1088/1361-6382/abfd85](https://doi.org/10.1088/1361-6382/abfd85) (page 18).

- [95] Chris Pankow, Katerina Chatziioannou, Eve A. Chase, Tyson B. Littenberg, Matthew Evans, Jessica McIver, Neil J. Cornish, Carl-Johan Haster, Jonah Kanner, Vivien Raymond, and et al. “Mitigation of the instrumental noise transient in gravitational-wave data surrounding GW170817”. In: *Physical Review D* 98.8 (Oct. 2018). DOI: [10.1103/physrevd.98.084016](https://doi.org/10.1103/physrevd.98.084016) (page 18).
- [96] B P Abbott et al. “Effects of data quality vetoes on a search for compact binary coalescences in Advanced LIGO’s first observing run”. In: *Classical and Quantum Gravity* 35.6 (Feb. 2018), p. 065010. DOI: [10.1088/1361-6382/aaaafa](https://doi.org/10.1088/1361-6382/aaaafa) (page 18).
- [97] A. L. Samuel. “Some Studies in Machine Learning Using the Game of Checkers”. In: *IBM Journal of Research and Development* 3.3 (1959), pp. 210–229. DOI: [10.1147/rd.33.0210](https://doi.org/10.1147/rd.33.0210) (page 19).
- [98] Kenji Kawaguchi, Leslie Pack Kaelbling, and Yoshua Bengio. *Generalization in Deep Learning*. 2020. arXiv: [1710.05468](https://arxiv.org/abs/1710.05468) [stat.ML] (page 19).
- [99] Marko Misić, Đorđe Đurđević, and Milo Tomasević. “Evolution and trends in GPU computing”. In: Jan. 2012, pp. 289–294 (page 20).
- [100] Qiong Liu and Ying Wu. “Supervised Learning”. In: *Encyclopedia of the Sciences of Learning*. Ed. by Norbert M. Seel. Boston, MA: Springer US, 2012, pp. 3243–3245. DOI: [10.1007/978-1-4419-1428-6_451](https://doi.org/10.1007/978-1-4419-1428-6_451) (page 20).
- [101] H B Barlow. “Unsupervised learning: introduction”. In: *Unsupervised Learning: Foundations of Neural Computation*. Ed. by Geoffrey E. Hinton and Terrence Joseph Sejnowski. Bradford Company Scituate, MA, USA, 1999, pp. 1–17 (page 20).
- [102] Bahar Azari, Christiana Westlin, Ajay Satpute, J. Hutchinson, Philip Kragel, Katie Hoemann, Zulqarnain Khan, Jolie Wormwood, Karen Quigley, Deniz Erdogmus, Jennifer Dy, Dana Brooks, and Lisa Barrett. “Comparing supervised and unsupervised approaches to emotion categorization in the human brain, body, and subjective experience.” In: (Mar. 2020). DOI: [10.31234/osf.io/egh2t](https://doi.org/10.31234/osf.io/egh2t) (page 20).
- [103] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: A Bradford Book, 2018 (page 20).
- [104] Xin Yan and Xiao Gang Su. *Linear Regression Analysis: Theory and Computing*. USA: World Scientific Publishing Co., Inc., 2009 (page 21).
- [105] David R Cox. “The regression analysis of binary sequences”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 20.2 (1958), pp. 215–232 (page 21).
- [106] Corinna Cortes and Vladimir Vapnik. “Support-vector networks”. In: *Machine learning* 20.3 (1995), pp. 273–297 (page 21).
- [107] Harris Drucker, Chris J. C. Burges, Linda Kaufman, Alex Smola, and Vladimir Vapnik. “Support Vector Regression Machines”. In: *Proceedings of the 9th International Conference on Neural Information Processing Systems*. NIPS’96. Denver, Colorado: MIT Press, 1996, pp. 155–161 (page 21).
- [108] Johannes Fürnkranz. “Decision Tree”. In: *Encyclopedia of Machine Learning*. Ed. by Claude Sammut and Geoffrey I. Webb. Boston, MA: Springer US, 2010, pp. 263–267. DOI: [10.1007/978-0-387-30164-8_204](https://doi.org/10.1007/978-0-387-30164-8_204) (page 21).
- [109] Leo Breiman. “Machine Learning, Volume 45, Number 1 - SpringerLink”. In: *Machine Learning* 45 (Oct. 2001), pp. 5–32. DOI: [10.1023/A:1010933404324](https://doi.org/10.1023/A:1010933404324) (page 21).
- [110] Leo Breiman. “Bias, Variance , And Arcing Classifiers”. In: *Technical Report 460, Statistics Department, University of California* (Nov. 2000) (page 21).
- [111] Robert E Schapire. “Explaining adaboost”. In: *Empirical inference*. Springer, 2013, pp. 37–52 (page 22).
- [112] Tianqi Chen and Carlos Guestrin. “XGBoost: A Scalable Tree Boosting System”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’16. San Francisco, California, USA: ACM, 2016, pp. 785–794. DOI: [10.1145/2939672.2939785](https://doi.org/10.1145/2939672.2939785) (page 22).
- [113] Xin Jin and Jiawei Han. “K-Means Clustering”. In: *Encyclopedia of Machine Learning*. Ed. by Claude Sammut and Geoffrey I. Webb. Boston, MA: Springer US, 2010, pp. 563–564. DOI: [10.1007/978-0-387-30164-8_425](https://doi.org/10.1007/978-0-387-30164-8_425) (page 22).
- [114] Douglas Reynolds. “Gaussian Mixture Models”. In: *Encyclopedia of Biometrics*. Ed. by Stan Z. Li and Anil Jain. Boston, MA: Springer US, 2009, pp. 659–663. DOI: [10.1007/978-0-387-73003-5_196](https://doi.org/10.1007/978-0-387-73003-5_196) (page 22).
- [115] L.J.P. van der Maaten and G.E. Hinton. “Visualizing High-Dimensional Data Using t-SNE”. In: *Journal of Machine Learning Research* 9 (2008), pp. 2579–2605 (page 22).

- [116] Leland McInnes, John Healy, and James Melville. *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*. 2020. arXiv: [1802.03426 \[stat.ML\]](https://arxiv.org/abs/1802.03426) (page 22).
- [117] Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. “Testing the Manifold Hypothesis”. In: *Journal of the American Mathematical Society* 29 (Oct. 2013). DOI: [10.1090/jams/852](https://doi.org/10.1090/jams/852) (page 22).
- [118] Karl Pearson F.R.S. “LIII. On lines and planes of closest fit to systems of points in space”. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11 (1901), pp. 559–572. DOI: [10.1080/14786440109462720](https://doi.org/10.1080/14786440109462720) (page 22).
- [119] Giuseppe Bonaccorso. *Machine Learning Algorithms: A Reference Guide to Popular Algorithms for Data Science and Machine Learning*. Packt Publishing, 2017 (page 22).
- [120] F. Rosenblatt. “The perceptron: A probabilistic model for information storage and organization in the brain.” In: *Psychological Review* 65.6 (1958), pp. 386–408. DOI: [10.1037/h0042519](https://doi.org/10.1037/h0042519) (pages 22, 23).
- [121] “Bio-inspired Neurocomputing”. In: *Studies in Computational Intelligence* (2021). DOI: [10.1007/978-981-15-5495-7](https://doi.org/10.1007/978-981-15-5495-7) (page 23).
- [122] Qi Wang, Yue Ma, Kun Zhao, and Yingjie Tian. “A Comprehensive Survey of Loss Functions in Machine Learning”. In: *Annals of Data Science* (Apr. 2020). DOI: [10.1007/s40745-020-00253-5](https://doi.org/10.1007/s40745-020-00253-5) (page 25).
- [123] Sebastian Ruder. “An overview of gradient descent optimization algorithms”. In: *arXiv preprint arXiv:1609.04747* (2016) (pages 25, 27).
- [124] Robin M. Schmidt, Frank Schneider, and Philipp Hennig. *Descending through a Crowded Valley – Benchmarking Deep Learning Optimizers*. 2021. arXiv: [2007.01547 \[cs.LG\]](https://arxiv.org/abs/2007.01547) (page 27).
- [125] Agnes Lydia and Sagayaraj Francis. “Adagrad - An Optimizer for Stochastic Gradient Descent”. In: Volume 6 (May 2019), pp. 566–568 (page 27).
- [126] Geoffrey Hinton. *Neural Networks for Machine Learning*. 2012 (page 27).
- [127] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: [1412.6980 \[cs.LG\]](https://arxiv.org/abs/1412.6980) (page 27).
- [128] D. Rumelhart, Geoffrey E. Hinton, and R. J. Williams. “Learning representations by back-propagating errors”. In: *Nature* 323 (1986), pp. 533–536 (page 27).
- [129] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. The MIT Press, 2016 (page 27).
- [130] Jürgen Schmidhuber. “Deep learning in neural networks: An overview”. In: *Neural Networks* 61 (Jan. 2015), pp. 85–117. DOI: [10.1016/j.neunet.2014.09.003](https://doi.org/10.1016/j.neunet.2014.09.003) (pages 28, 29).
- [131] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015. arXiv: [1409.1556 \[cs.CV\]](https://arxiv.org/abs/1409.1556) (page 28).
- [132] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. “Going deeper with convolutions”. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 1–9. DOI: [10.1109/CVPR.2015.7298594](https://doi.org/10.1109/CVPR.2015.7298594) (page 28).
- [133] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. *Deep Residual Learning for Image Recognition*. 2015. arXiv: [1512.03385 \[cs.CV\]](https://arxiv.org/abs/1512.03385) (page 28).
- [134] Hossein Gholamalinezhad and Hossein Khosravi. *Pooling Methods in Deep Neural Networks, a Review*. 2020. arXiv: [2009.07485 \[cs.CV\]](https://arxiv.org/abs/2009.07485) (page 28).
- [135] Sepp Hochreiter and Jürgen Schmidhuber. “Long short-term memory”. In: *Neural computation* 9.8 (1997), pp. 1735–1780 (page 30).
- [136] Mark A. Kramer. “Nonlinear principal component analysis using autoassociative neural networks”. In: *AIChE Journal* 37.2 (1991), pp. 233–243. DOI: <https://doi.org/10.1002/aic.690370209>. eprint: [https://aiche.onlinelibrary.wiley.com/doi/pdf/10.1002/aic.690370209](https://onlinelibrary.wiley.com/doi/pdf/10.1002/aic.690370209) (page 30).
- [137] Diederik P. Kingma and Max Welling. “An Introduction to Variational Autoencoders”. In: *Foundations and Trends® in Machine Learning* 12.4 (2019), pp. 307–392. DOI: [10.1561/22000000056](https://doi.org/10.1561/22000000056) (page 30).
- [138] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. “Generative Adversarial Nets”. In: *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*. NIPS’14. Montreal, Canada: MIT Press, 2014, pp. 2672–2680 (page 30).
- [139] Vanessa Buhrmester, David Münch, and Michael Arens. *Analysis of Explainers of Black Box Deep Neural Networks for Computer Vision: A Survey*. 2019. arXiv: [1911.12116 \[cs.AI\]](https://arxiv.org/abs/1911.12116) (page 31).

- [140] Neil A. Thacker, Carole J. Twining, Paul D. Tar, Scott Notley, and Visvanathan Ramesh. *Fundamental Issues Regarding Uncertainties in Artificial Neural Networks*. 2020. arXiv: [2002.11152 \[stat.ML\]](#) (page 31).
- [141] Jakob Gawlikowski et al. *A Survey of Uncertainty in Deep Neural Networks*. 2021. arXiv: [2107.03342 \[cs.LG\]](#) (page 31).
- [142] Murat Seçkin Ayhan and Philipp Berens. “Test-time Data Augmentation for Estimation of Heteroscedastic Aleatoric Uncertainty in Deep Neural Networks”. In: 2018 (page 31).
- [143] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. *On Calibration of Modern Neural Networks*. 2017. arXiv: [1706.04599 \[cs.LG\]](#) (page 31).
- [144] Andrew Gordon Wilson and Pavel Izmailov. *Bayesian Deep Learning and a Probabilistic Perspective of Generalization*. 2020. arXiv: [2002.08791 \[cs.LG\]](#) (page 31).
- [145] Daniel George and E.A. Huerta. In: *Phys. Lett. B* 778 (2018), pp. 64–70. DOI: <https://doi.org/10.1016/j.physletb.2017.12.053> (page 32).
- [146] Hunter Gabbard, Michael Williams, Fergus Hayes, and Chris Messenger. “Matching Matched Filtering with Deep Networks for Gravitational-Wave Astronomy”. In: *Phys. Rev. Lett.* 120 (14 Apr. 2018), p. 141103. DOI: [10.1103/PhysRevLett.120.141103](#) (page 32).
- [147] Timothy D. Gebhard, Niki Kilbertus, Ian Harry, and Bernhard Schölkopf. “Convolutional neural networks: A magic bullet for gravitational-wave detection?” In: *Phys. Rev. D* 100 (6 Sept. 2019), p. 063015. DOI: [10.1103/PhysRevD.100.063015](#) (page 32).
- [148] C. Dreissigacker, R. Sharma, C. Messenger, Ruining Zhao, and R. Prix. In: *Phys. Rev. D* 100 (4 Aug. 2019), p. 044009. DOI: [10.1103/PhysRevD.100.044009](#) (page 32).
- [149] Christoph Dreissigacker and Reinhard Prix. “Deep-learning continuous gravitational waves: Multiple detectors and realistic noise”. In: *Physical Review D* 102.2 (July 2020). DOI: [10.1103/physrevd.102.022005](#) (page 32).
- [150] Andrew L. Miller et al. “How effective is machine learning to detect long transient gravitational waves from neutron stars in a real search?” In: *Phys. Rev. D* 100 (6 Sept. 2019), p. 062005. DOI: [10.1103/PhysRevD.100.062005](#) (page 32).
- [151] Joe Bayley, Chris Messenger, and Graham Woan. “Robust machine learning algorithm to search for continuous gravitational waves”. In: *Physical Review D* 102.8 (Oct. 2020). DOI: [10.1103/physrevd.102.083024](#) (page 32).
- [152] Man Leong Chan, Ik Siong Heng, and Chris Messenger. “Detection and classification of supernova gravitational wave signals: A deep learning approach”. In: *Physical Review D* 102.4 (Aug. 2020). DOI: [10.1103/physrevd.102.043022](#) (page 32).
- [153] P. Astone et al. “New method to observe gravitational waves emitted by core collapse supernovae”. In: *Phys. Rev. D* 98 (12 Dec. 2018), p. 122002. DOI: [10.1103/PhysRevD.98.122002](#) (page 32).
- [154] Hunter Gabbard, Chris Messenger, Ik Siong Heng, Francesco Tonolini, and Roderick Murray-Smith. *Bayesian parameter estimation using conditional variational autoencoders for gravitational-wave astronomy*. 2020. arXiv: [1909.06296 \[astro-ph.IM\]](#) (page 32).
- [155] Stephen R. Green, Christine Simpson, and Jonathan Gair. “Gravitational-wave parameter estimation with autoregressive neural network flows”. In: *Physical Review D* 102.10 (Nov. 2020). DOI: [10.1103/physrevd.102.104057](#) (page 32).
- [156] M. Razzano and E. Cuoco. In: *Clas. Quant. Grav.* 35.9 (Apr. 2018), p. 095016. DOI: [10.1088/1361-6382/aab793](#) (page 32).
- [157] Jade Powell et al. “Classification methods for noise transients in advanced gravitational-wave detectors II: performance tests on Advanced LIGO data”. In: *Classical and Quantum Gravity* 34.3 (Jan. 2017), p. 034002. DOI: [10.1088/1361-6382/34/3/034002](#) (page 32).
- [158] Daniel George, Hongyu Shen, and E. A. Huerta. “Classification and unsupervised clustering of LIGO data with Deep Transfer Learning”. In: *Physical Review D* 97.10 (May 2018). DOI: [10.1103/physrevd.97.101501](#) (page 32).
- [159] S. Bahaadini, V. Noroozi, N. Rohani, S. Coughlin, M. Zevin, J.R. Smith, V. Kalogera, and A. Katsaggelos. “Machine learning for Gravity Spy: Glitch classification and dataset”. In: *Information Sciences* 444 (2018), pp. 172–186. DOI: <https://doi.org/10.1016/j.ins.2018.02.068> (page 32).
- [160] H. Shen, D. George, E. A. Huerta, and Z. Zhao. “Denoising Gravitational Waves with Enhanced Deep Recurrent Denoising Auto-encoders”. In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2019, pp. 3237–3241. DOI: [10.1109/ICASSP.2019.8683061](#) (page 32).

- [161] Alejandro Torres-Forné, Elena Cuoco, José A. Font, and Antonio Marquina. “Application of dictionary learning to denoise LIGO’s blip noise transients”. In: *Physical Review D* 102.2 (July 2020). DOI: [10.1103/physrevd.102.023011](https://doi.org/10.1103/physrevd.102.023011) (page 32).
- [162] J McGinn, C Messenger, M J Williams, and I S Heng. “Generalised gravitational wave burst generation with generative adversarial networks”. In: *Classical and Quantum Gravity* 38.15 (June 2021), p. 155005. DOI: [10.1088/1361-6382/ac09cc](https://doi.org/10.1088/1361-6382/ac09cc) (page 32).
- [163] Elena Cuoco et al. “Enhancing gravitational-wave science with machine learning”. In: *Machine Learning: Science and Technology* 2.1 (Dec. 2020), p. 011002. DOI: [10.1088/2632-2153/abb93a](https://doi.org/10.1088/2632-2153/abb93a) (page 32).