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

MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
4 November 2019

REVISED
20 March 2020

ACCEPTED FOR PUBLICATION
6 April 2020

PUBLISHED
22 June 2020

PAPER

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

Filip Morawski[1], **Michał Bejger** and **Paweł Ciecieląg**

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 of 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 maximiszation 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 add 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

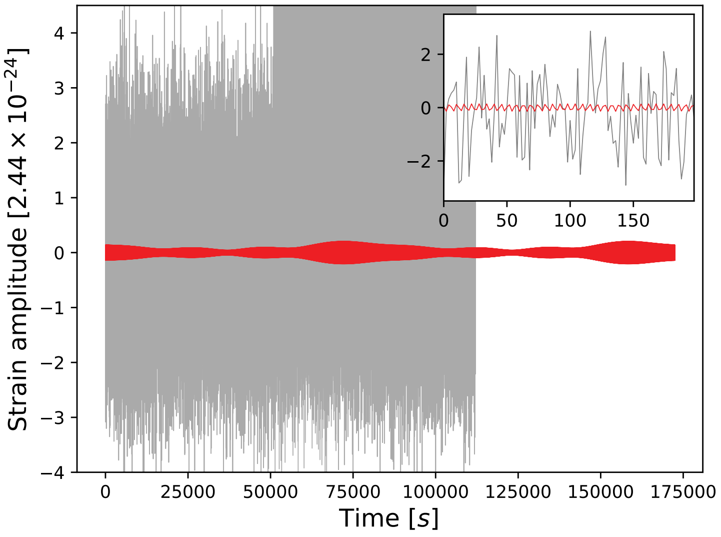


Figure 1. Example of continuous GW time-domain data input of `TD-Fstat search`. The grey time series of $T_0 = 2$ sidereal days length mimics the downsampled, narrow-banded data produced from the raw interferometer data [17, 20]. The data contains an almost-monochromatic astrophysical GW signal (red curve) of $\rho_{\mathrm{inj}} = 10$, 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 $\dot{f}_{\mathrm{astro}} = -3.03 \times 10^{-9}$ 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 $(f, \dot{f}, \delta, \alpha)_{\mathrm{inj}}$, 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, $\rho_{\mathrm{inj}} = h_0\sqrt{T_0/S}$. The output SNR ρ for a candidate signal corresponding to $(f, \dot{f}, \delta, \alpha)_{\mathrm{inj}}$ 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 $(f, \dot{f}, \delta, \alpha)_{\mathrm{inj}}$ 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 $(f, \dot{f}, \delta, \alpha)_{\mathrm{inj}}$, 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 **cgw**s or **line**s 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 $\rho_{\rm inj}$	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 $\rho_{\rm inj}$ 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 $\rho_{\rm inj}$, 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 $\rho_{\rm inj}$. 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 $\rho_{\rm inj} = 10$ case). For $\rho_{\rm inj} < 8$, 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 $\rho_{\rm inj}$ 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 $\rho_{\rm inj}$, 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

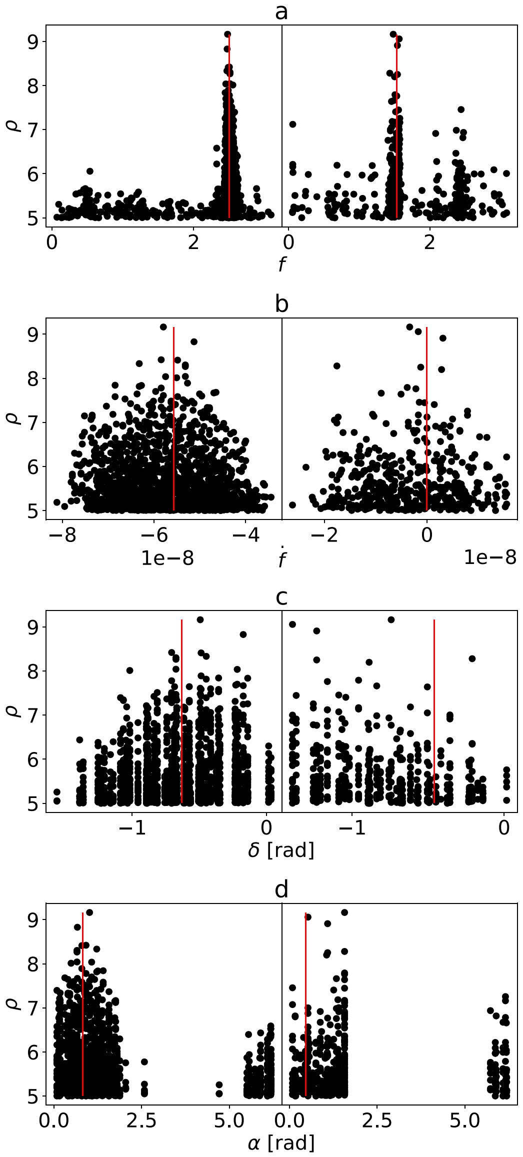


Figure 2. Exemplary `TD-Fstat search` outputs for continuous GW signal and stationary line generated for the $\rho_{\rm inj} = 10$ and the reference band frequency $f = 100$ Hz and parameters of the injected signal from figure 1 ($f = 2.16, \dot{f} = -3.81 \times 10^{-8}$, $\delta = 0.474$ and $\alpha = 5.84$) marked by red vertical lines in the above plots. The left column corresponds to the **cgw** and the right column to the **line**. The distributions represent the relation of recovered SNR ρ ($\mathcal{F}$-statistic SNR) with respect to: the frequency – *a*, the derivative of frequency – *b*, the declination – *c*, and the right ascension – *d*.

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 P6000[2] and performed

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

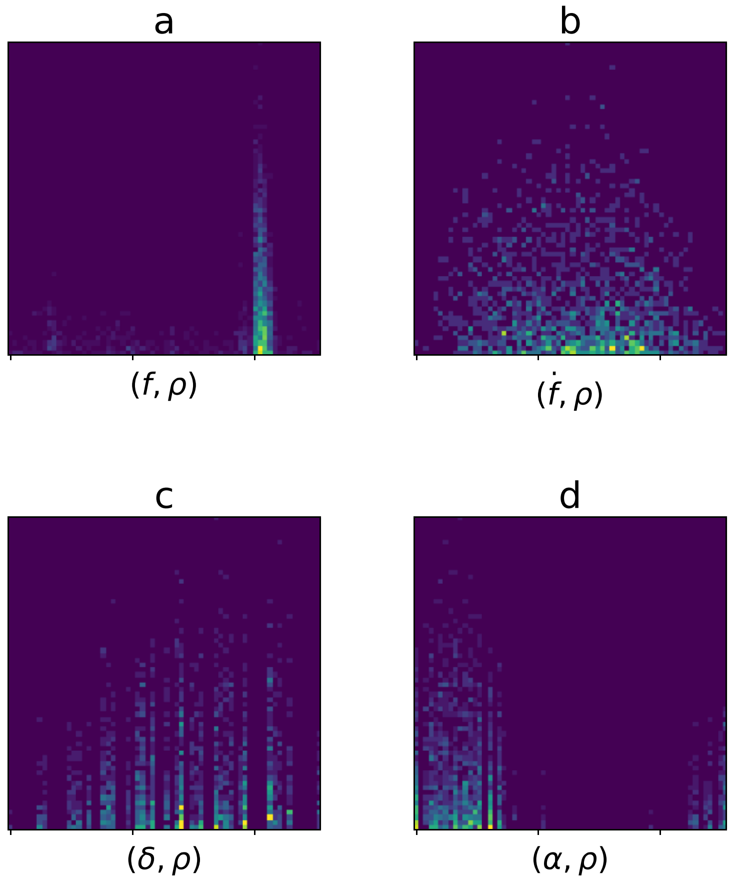


Figure 3. 2D representation of `TD-Fstat search` outputs used as an input to the 2D CNN. Images presented here have equal size of 64 × 64 pixels. They correspond to the distributions shown on the left column of figure 2: (a) – (frequency, ρ), (b) – (spindown, ρ), (c) – (declination, ρ), (d) – (right ascension, ρ). The colours correspond to the density of the distribution—the brighter it is, the more points contributed to the pixel.

the production runs on the Cyfronet Prometheus cluster[3] 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.

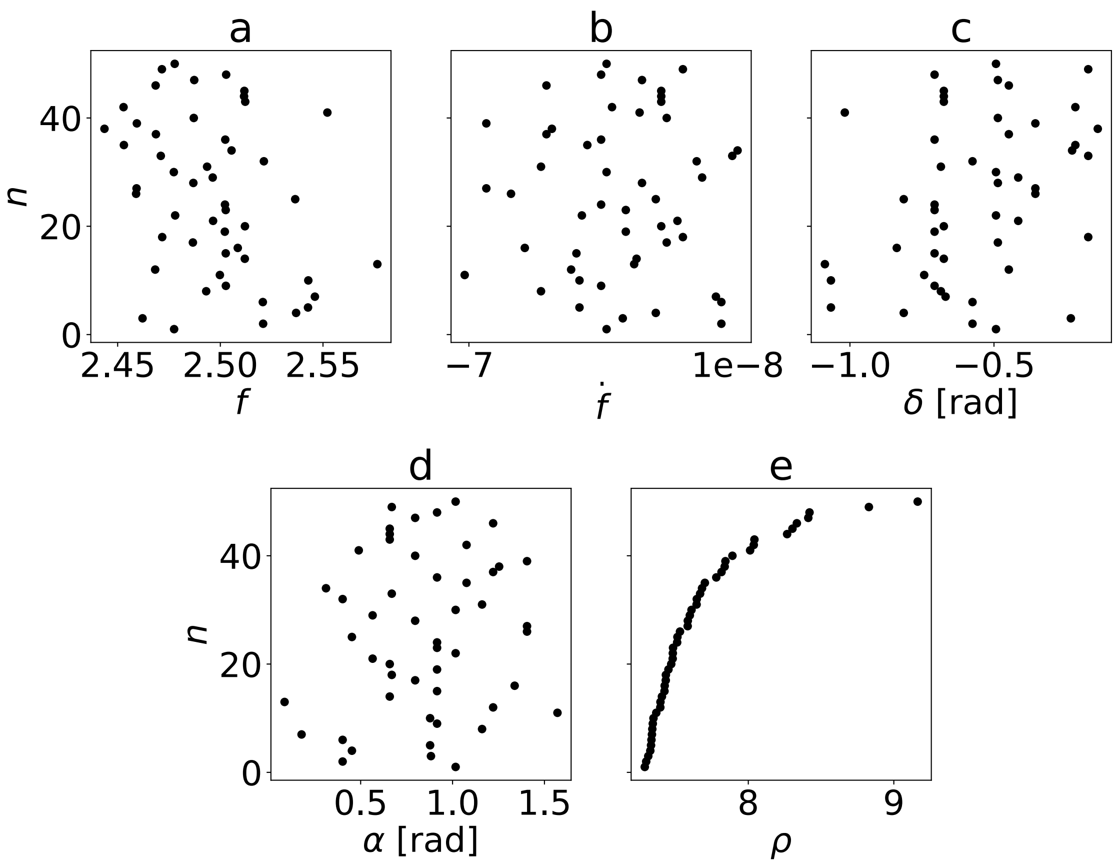
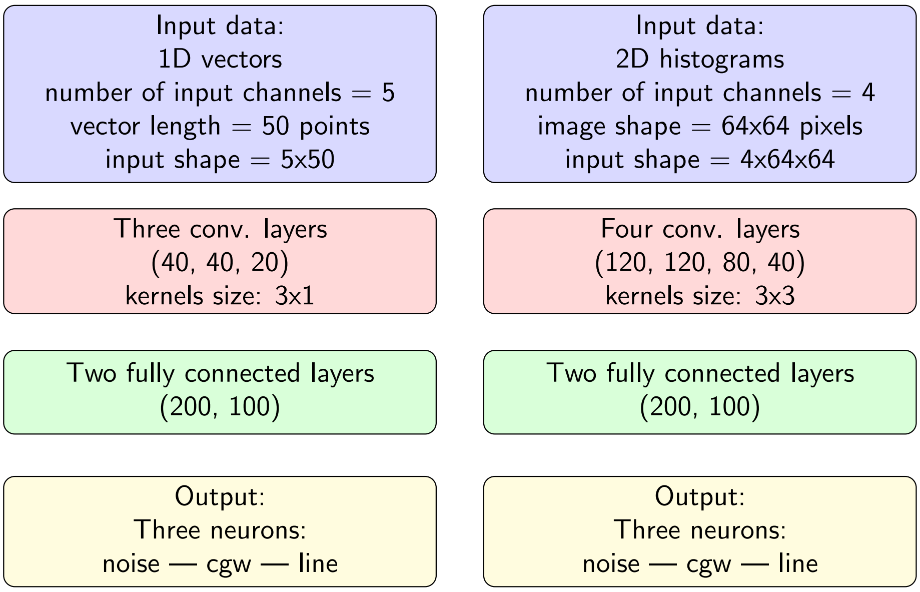


Figure 4. 1D representation of `TD-Fstat search` outputs used as an input to the 1D CNN. The outputs are limited to the 50 maximum values of ρ (plots presented here correspond to the distributions shown on the left column of figure 2(a)): (a) – frequency, (b) – spindown, (c) – declination, (d) – right ascension and (e) – SNR ρ. The vector of ρ was sorted since it allowed higher accuracy to be reached during training.



(a) 1D CNN

(b) 2D CNN

Figure 5. Diagrams showing the networks' layer structure and architecture.

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.

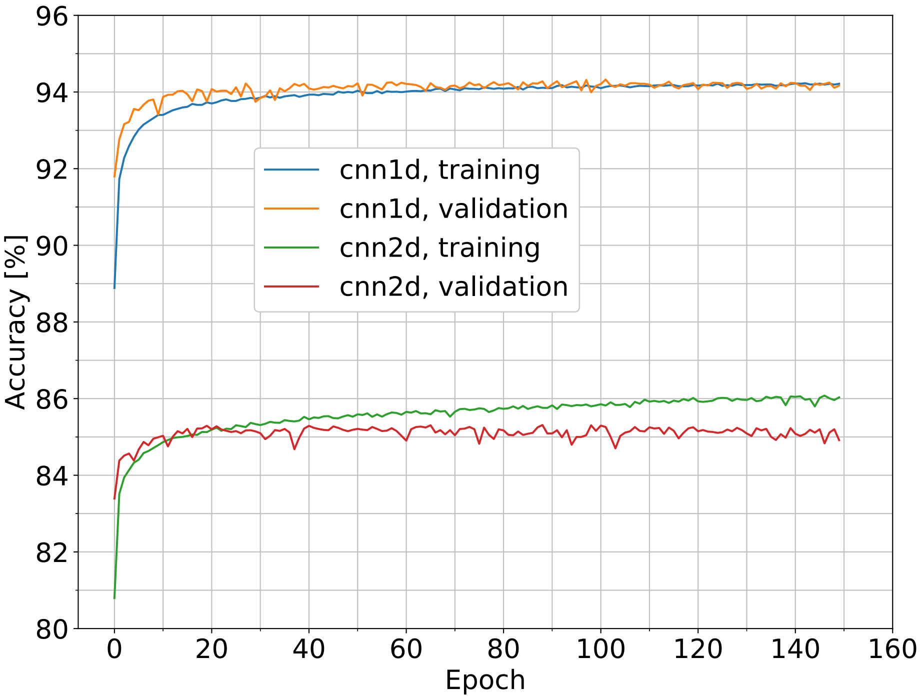


Figure 6. Evolution of the accuracy as a function of training epoch for the three-label classification for the 1D CNN (upper curves) and 2D CNN (lower curves). Both models reached maximum accuracy after 40 epochs (based on the results of the validation set). We prolonged training to 150 epochs to investigate the onset of overfitting. The 1D CNN was still properly learning (although without an increase in validation accuracy), whereas the 2D CNN showed overfitting—validation accuracy (red curve) was maintained at a constant level when training accuracy (green curve) was increasing.

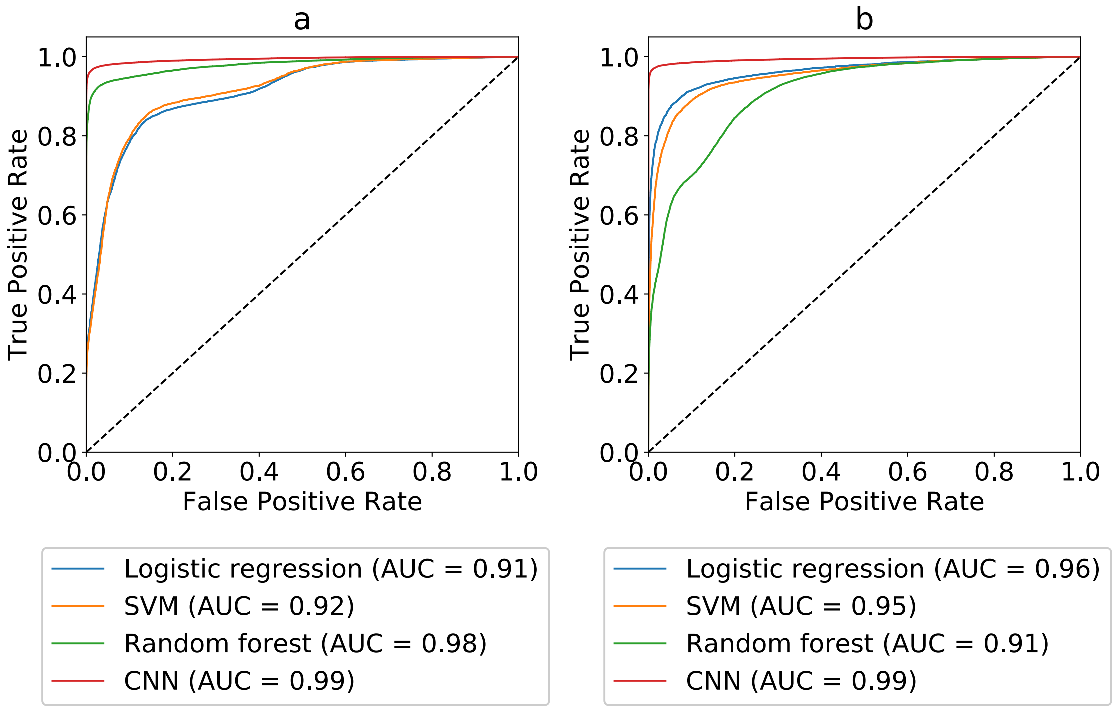


Figure 7. Receiver-operating-characteristic (ROC) curves for compared ML algorithms trained on 1D data representation (*a*) and 2D data representation (*b*). The presented results are for the binary classification problem in which *Positive* stands for **cgw** whereas *Negative* corresponds to the combined **line** and **noise** class. The black dashed line in the middle corresponds to random guessing (AUC stands for the Area Under Curve).

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				
	Logistic regression	*SVM*	*Random forest*	*CNN*
1D data	33.8	31.9	89.2	96.3
2D data	72.8	57.6	38.4	96.8

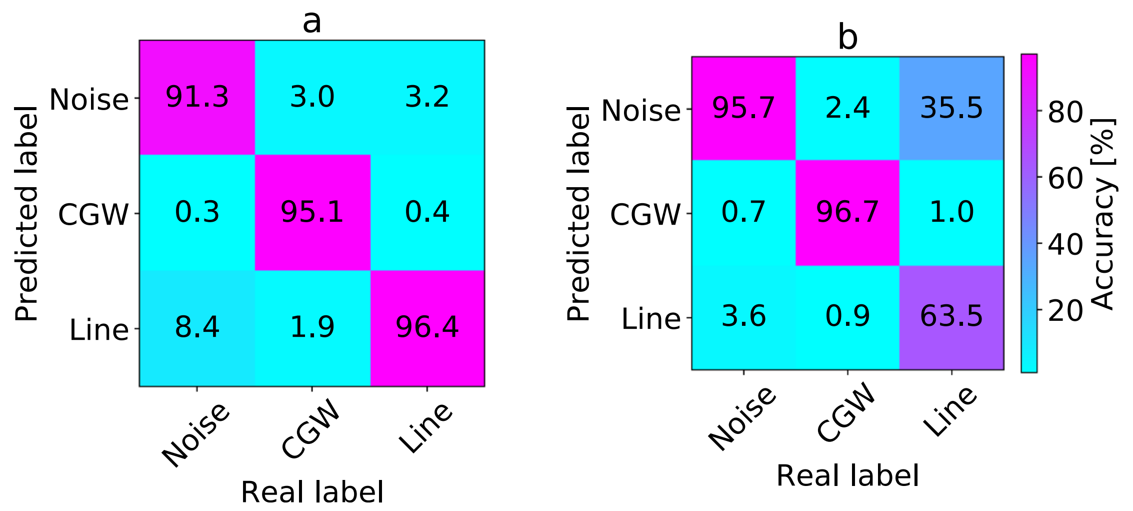


Figure 8. Confusion matrix for the three-label classification evaluated on the test set for the 1D CNN – *a* and 2D CNN – *b* after the training. Although the **cgw** and the **noise** were classified on a similar level, the **line** caused significant problem for the 2D model. The majority of **line** instances resembled **noise** class in the image representation.

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 $\rho_{\text{inj}} \geq 10$ (reaching a value of more than 90% for all of them). Interestingly, candidates with $\rho_{\text{inj}} < 8$ 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 $\rho_{\text{inj}} < 8$. 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 $\rho_{\text{inj}} > 10$). For the other frequencies, the maximum accuracy was gradually shifted toward increasing ρ_{inj}. Interestingly, the accuracy for 50 Hz reached the maximum for $\rho_{\text{inj}} = 10$; 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

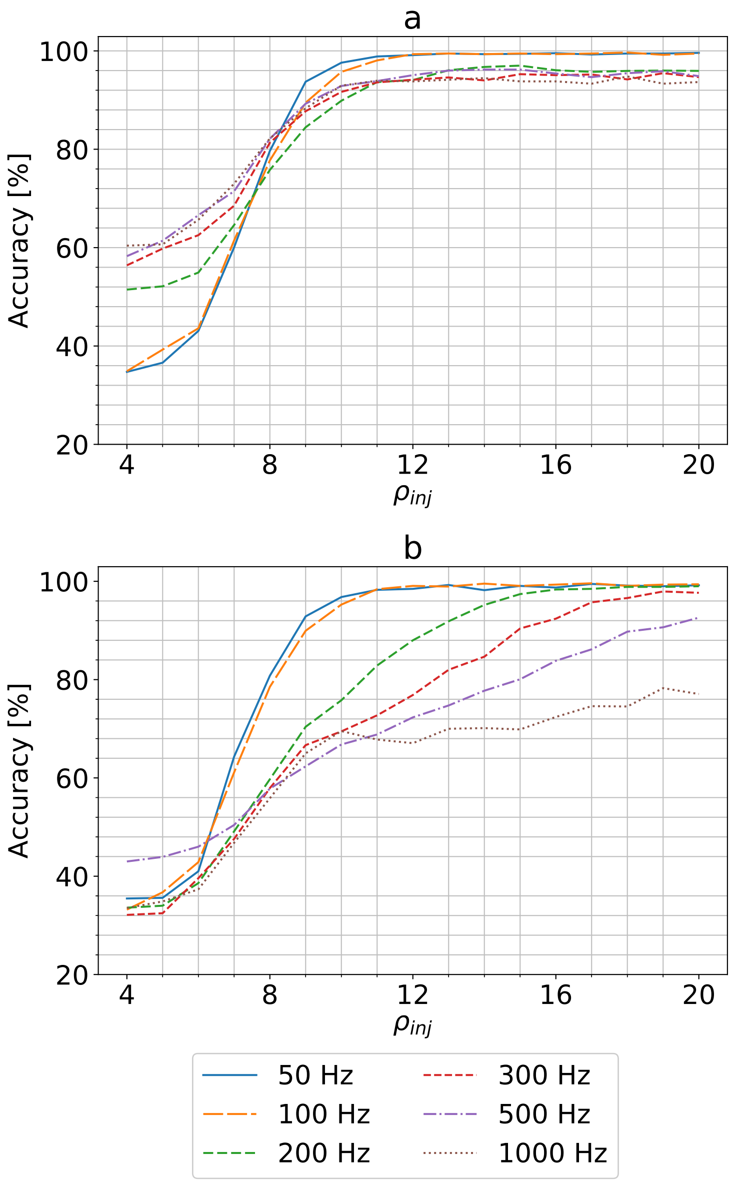


Figure 9. Evolution of accuracy as the function of the injected SNR ρ_{inj} for 1D CNN (a) and 2D CNN (b). The first model achieved a maximum level of accuracy of $\rho_{\text{inj}} = 10-12$ and maintained its value for the whole range of frequencies. The 2D version varied significantly in relation to 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

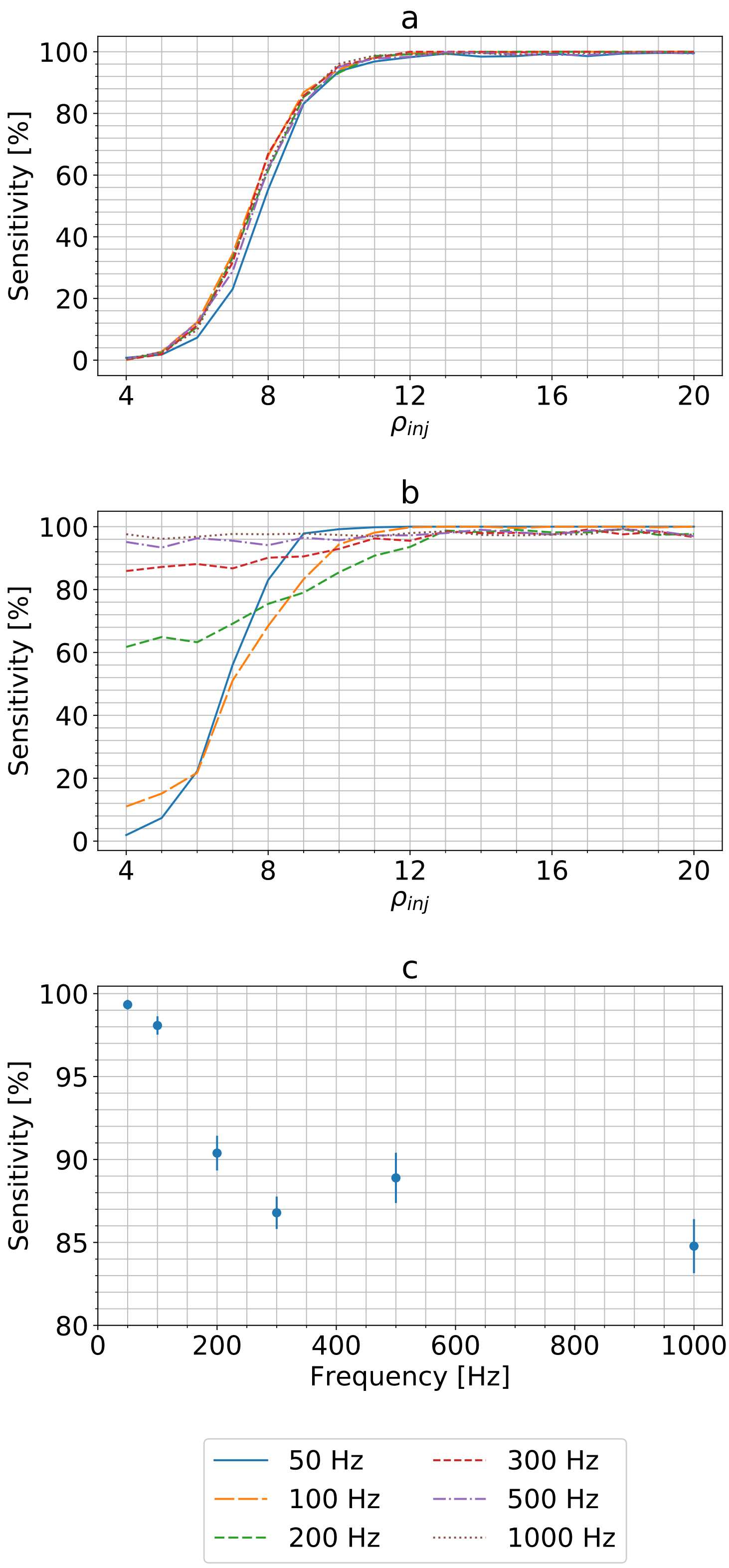


Figure 10. Evolution of sensitivity as a function of SNR ρ_{inj} of the 1D CNN for the three types of signal candidates: **cgw** – (a), **line** – (b), and **noise** – (c). The last panel shows average values for frequencies, because the **noise** classification sensitivity is not a function of the injected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

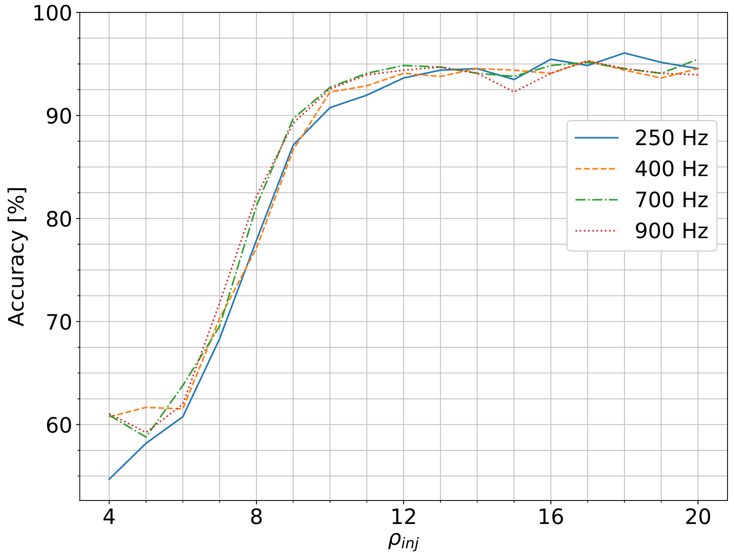


Figure 11. Evolution of accuracy as a function of the injected SNR ρ_{inj} for 1D CNN for signal candidates generated with frequencies different from those used for the training.

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 ($\rho_{\text{inj}} \geq 10$) 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 Cieślar 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 Preuß Ischen 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.* **X 9**, 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] Poghosyan 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 Brüggen 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