Neural network time-series classifiers for gravitational-wave searches in single-detector periods

A. Trovato

Dipartimento di Fisica, Università di Trieste, I-34127 Trieste, Italy
INFN, Sezione di Trieste, I-34127 Trieste, Italy

E-mail: agata.trovato@units.it

E. Chassande-Mottin

Université Paris Cité, CNRS, Astroparticule et Cosmologie, F-75013 Paris, France

M. Bejger

INFN, Sezione di Ferrara, I-44122 Ferrara, Italy
Nicolaus Copernicus Astronomical Center, Polish Academy of Sciences, ul. Bartycka 18, 00-716 Warsaw, Poland

R. Flamary

Ecole polytechnique, IP Paris, CMAP, F-91120 Palaiseau, France

N. Courty

Université Bretagne Sud, CNRS IRISA, F-35042 Rennes, France

Abstract. The search for gravitational-wave signals is limited by non-Gaussian transient noises that mimic astrophysical signals. Temporal coincidence between two or more detectors is used to mitigate contamination by these instrumental glitches. However, when a single detector is in operation, coincidence is impossible, and other strategies have to be used. We explore the possibility of using neural network classifiers and present the results obtained with three types of architectures: convolutional neural network, temporal convolutional network, and inception time. The last two architectures are specifically designed to process time-series data. The classifiers are trained on a month of data from the LIGO Livingston detector during the first LIGO-Virgo observing run (O1) to identify data segments that include the signature of a binary black hole merger. Their performances are assessed and compared. We then apply trained classifiers to the remaining three months of O1 data, focusing specifically on single-detector times that were not thoroughly examined. We find an outstanding signal on 2016-01-04 12:24:17 UTC. Although we are not able to constrain the significance of this event to the level conventionally followed in gravitational-wave searches, we show that the signal is compatible with the merger of two black holes with masses $m_1 = 50.7^{+10.4}_{-8.9}\,M_\odot$ and $m_2 = 24.4^{+20.2}_{-9.3}\,M_\odot$ at the luminosity distance of $d_L = 564^{+812}_{-338}$ Mpc.

1. Introduction

The breakthrough discovery of gravitational waves (GW) on September 14, 2015 [1], announced by the LIGO Scientific Collaboration [2] and the Virgo Collaboration [3], opened the era of the GW astronomy. The detection happened during the first observing run (O1) of the LIGO detector. With the subsequent observing runs, O2 and O3, performed jointly with Virgo, the list of detected GW signals has grown to 90 events. While the detected sources are mainly associated with the merger of binary black holes (BBH), they also include binary systems with neutron stars [4–7]. These detections are collected and characterized in the GW transient catalogs GWTC [8–11]. On May 2023 the forth observing run (O4) started with an increasing detectors sensitivity and consequently an enhanced expected rate of detections.

GW transient signals are detected in the data by a variety of data analysis pipelines, see e.g. [12] for a recent review. In particular, matched filtering [13] is a prominent technique to search for signals when an accurate waveform model is available, as in the case of compact star binary mergers. Algorithmically, this consists in correlating the data with a large set of template waveform models (the "template bank") that are representative of all the morphologies the expected signal can possibly take.

To make robust detection statements, those pipelines have to address a major difficulty: the presence in the data of short duration noise artefacts, often called "instrumental glitches" [14, 15], that can mimic the GW signal. A very powerful lever to discriminate the signal from noise glitches is time coincidence across two or more separate detectors (see [16] for a discussion on multi-detector noise rejection techniques).

Obviously, coincidence cannot be used during periods when only one detector operates. During the O1 and O2 observing runs, single-detector periods amount to about 30% of the observation time [17, 18]. During O3, thanks to a more stable and reliable operation, this fraction was reduced to about 15% in O3a [19] and 11% in O3b [20] (the first and second six months parts of O3). In total, more than five months of observing time fall in this category, so far. The O4 science run initiated recently may also have long periods of single detector times.

The lack of coincidence results in difficulties to disentangle the signal from glitches and to measure the statistical significance of a trigger. A number of studies investigate ways to measure the contrast between GW signal and noise background (including glitches) in single-detector data. Ref. [21] studies the possibility to extend the multivariate likelihood-ratio statistics used by the `GstLAL` pipeline to generate single-detector events. The likelihood estimation has been recently updated in view of the O4 run [22] and one of the improvements is the addition of a tuneable penalty in case of single detectors candidates in order to down weight their significance [23]. To extrapolate the significance measure of single-detector triggers produced by the PyCBC pipeline [24], a method proposed in [25] allows to recover loud signals in single-detector data. In both cases, it is shown that the search sensitivity is significantly reduced compared to multi-detector searches.

Glitches vary widely in duration, frequency range and morphology. It is difficult to construct a statistical model able to capture the overall complexity of the glitch population. Their complex and time-evolving nature makes of glitch identification and rejection a good problem and use-case for machine learning (ML). In principle, this approach allows to train a classifier able to distinguish between different types of input (glitches versus real GW signal in our case), and thus learn a possibly very complex and high-dimensional statistical model from a set of examples.

As in many scientific fields, the use of ML has recently gained in popularity in the context of GW astronomy. There is a fairly large body of works pertaining to various aspects ranging from denoising, glitch classification and cancellation, waveform modelling, searches for GW signals, astrophysical parameter estimation, population studies (see e.g. [26, 27] for recent reviews).

In the context of GW signal searches, convolutional neural networks (CNN) [28] have been investigated to detect BBH signals for both single- and multi-detector case [29–34]. The primary motivation put forward in those contributions is the computational gain expected from the use of CNNs compared to matched filtering techniques.

So far the large fraction of those investigations use simulated Gaussian noise [29, 31, 32, 34]. In this case, it is not possible to learn the non-Gaussian component of the instrumental noise. Few studies use real GW data including glitches [30, 33]. The classifier obtained in those contributions are limited to false positive probability (i.e., noise or glitches classified as signal) of about 1%. This corresponds to a false alarm rate of once every 40 minutes, which is not sufficient in practice. A recent review [35] compares different approaches on a mock data challenge.

The purpose of this study is to enhance the ability of neural network based searches to reject noise artifact and improve their sensitivity, with a particular focus on analyzing data from a single detector. The goal is to achieve a false alarm rate similar to that of current on-line searches performed by the LIGO-Virgo-KAGRA collaboration (LVK), i.e. two false alarms per day [36]. We explore various network architectures, particularly those designed for time series classification [37, 38].

We trained and tested neural network classifiers using a dataset produced from one month of O1 data collected by the LIGO Livingston detector, during which no GW were detected using the matched filtering based searches.

Section 2 provides details on how the training and testing sets are generated, while Sec. 3 describes the structure of the various neural network classifiers being considered. The performance and efficiency of the classifiers are assessed using testing data, and the results are presented in Sec. 4. We applied these classifiers to the remaining three months of O1 data, including segments associated with the three GW events detected during O1. Sec. 5 summarizes the results of this analysis. We checked the classifiers' response obtained with the known detected events during O1. A particular focus is then given to the single-detector times. As far as we know, no other comparable analyses have been published about this part of the O1 data set. Interestingly, we found that only one data segment was classified as "signal" by the three classifiers we considered. We conducted additional checks of this event to better understand its exact nature. Finally, Sec. 6 concludes on the applicability of the proposed methodology.

2. Generation of datasets for training and testing

The typical approach for applying ML methods to GW detection is to treat it as a classification problem, see e.g., [29–32, 34]. In this approach, we aim to determine whether a given segment of GW strain data of fixed duration contains an astrophysical signal or not. This problem can be solved by developing an ML-based classifier that is trained using example data. We produce training data labeled as follows:

- noise: the data are compatible with stationary background noise, i.e., is free of transient instrumental artifacts (glitches) or known GW events,
- glitch: the data include one or several transient instrumental artifacts (glitches),
- signal: the data include a (simulated) astrophysical signal, added to the stationary background noise.

This three-class approach differs from other contributions in the literature, which consider only two classes. The presence of glitches is known to significantly alter the statistical distribution of the data. By assigning a specific label to data segments containing glitches, the idea is that this may aid the classifier in achieving improved performance. Furthermore, the relative significance assigned to each class could offer valuable information when evaluating the contents of a given segment.

Training and testing data are extracted from the dataset of the observing run O1, which was publicly released via the Gravitational Wave Open Science Center (GWOSC) [39]. Specifically, we utilize the data from the LIGO Livingston detector spanning a one-month period between November 25, 2015 (GPS time 1132444817) and December 25, 2015 (GPS time 1135036817). Throughout this duration, no GW signals were detected by the standard search pipelines.

In this period the available L1 data amounts in total to about 13.3 days (1,147,457 s), of which 3.6 days (312,284 s) where in single-detector time, i.e. 27% of the time.

The raw data are sampled at 16 kHz. We have downsampled the data to 2048 Hz, bandpass-filtered between 20 Hz and 1 kHz and whitened by applying the inverse amplitude spectral density (ASD) in the frequency domain.[2] The ASD is estimated over stretches of variable length, depending on the duration of uninterrupted data taking periods (minimum duration is 37 s and maximum is 100573 s). The data are divided into one-second non-overlapping segments.

[2] For the preparation of the training and testing data, we acknowledge the use of the following software packages: `GWpy` [40], `PyCBC` [24] and `LALSuite` [41].

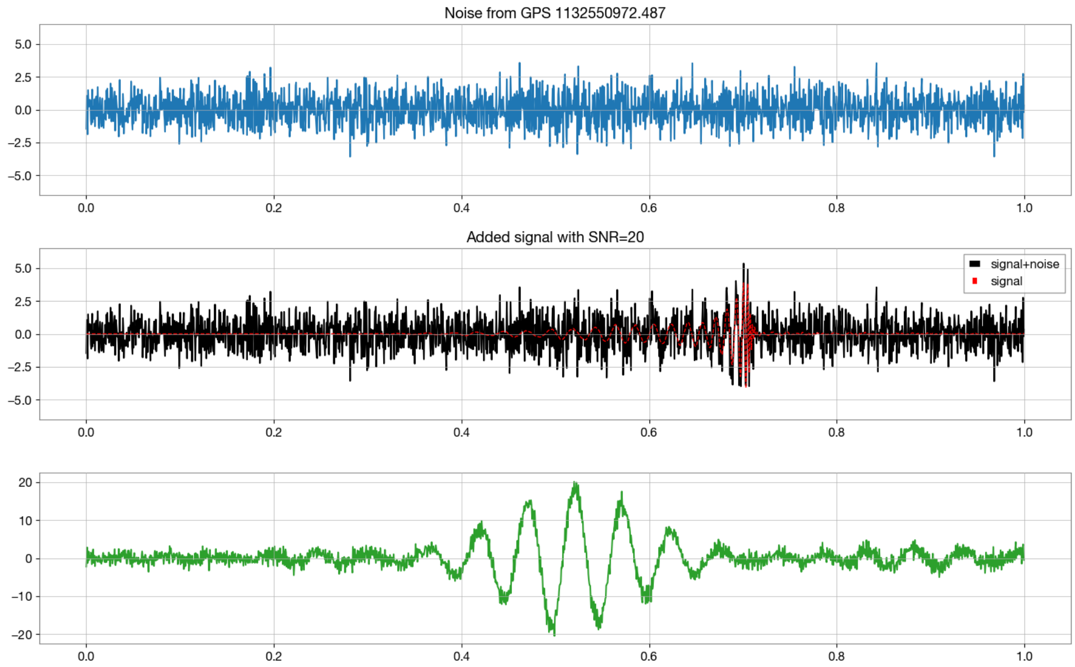


Figure 1. Instances of the classes noise (blue), signal (black) and glitch (green). Top (noise): one-second data segment recorded by LIGO Livingston at the GPS time 1132550972.487. Middle (signal): a simulated BBH waveform with SNR of 20 (dashed red line) is injected in the previous timeseries. Bottom (glitch): Data recorded at the GPS time 1132580628.41 which contains a low-frequency transient instrumental artifact.

The data are divided into the three classes introduced above as explained in the next sections. Representative instances of the three classes are shown in Fig. 1.

2.1. The noise *class*

The noise class corresponds to segments that are free of known GW signals, glitches (see next section) or hardware injections.[3] Overall, 750,000 noise samples are generated from the one-month O1 dataset.

2.2. The glitch *class*

A database of glitches is created using two different sources: the unmodelled transient search *coherent WaveBurst* (`cWB`) [42, 43] and the citizen science project `Gravity Spy` [44].

The cWB pipeline is an open-source software package designed to search for a wide range of GW transients without prior knowledge of the signal waveform. To evaluate the analysis background, cWB uses a resampling technique [42] that involves applying

[3] During O1, hardware injections, which are simulated signals created by manipulating mirrors in the arms of the interferometers, were added to the LIGO detectors for testing and calibration. See `https://www.gw-openscience.org/o1_inj`

non-physical time-shifts to the data prior to analysis. Loud, i.e., high signal-to-noise ratio (SNR), background triggers resulting from this procedure are good candidates for glitches. The loudest triggers in LIGO Livingston with the SNR higher than 5.8 were selected (258,480 glitches). This list was complemented with the `Gravity Spy` database (13,144 glitches). The timestamps and duration of the identified glitches from these two sources are collected in a single list, which is then used to label the one-second data segments from the O1 observing run. If the glitch duration is shorter than 1 second, the associated segment is labeled as a glitch. Note that the glitch has a random position within the one-second window. If the glitch duration is longer than 1 second, all segments that overlap with that glitch duration are labeled as glitch. Only the glitches whose time belongs to the data segments available on GWOSC are considered. In many cases, the glitches are closer in time than one second, so multiple glitches can fall in the same one-second segment.

From the one month O1 data, a total of $150,000$ segments receive the glitch label.

2.3. The signal *class*

The samples from the signal class are produced by adding simulated GW signals from BBH systems to the one-month O1 data in periods without known GW signals or hardware injections. For the training set, the data segments used to generate samples of the signal class are not utilized for the noise class nor the glitch class, while for the testing set, the same data segments are used for both the noise and signal classes. To generate the astrophysical signals, the waveform model `SEOBNRv4` [45] is employed, with a lower frequency cutoff of 30 Hz. The simulated signals are sampled, whitened, and band-pass filtered in the same manner as the data segments.

The masses of the binary BH used for generating the simulated signals in the class signal are chosen to ensure that they fall within the mass range observed by the LVK, and that the signals are short enough to be contained within the one-second data segments. Specifically, the component masses m_1 and m_2 are chosen randomly, with the constraint that $m_1 > m_2 \geq 10M_\odot$ and the total mass $M = m_1 + m_2$ is uniformly distributed in $33M_\odot \leq M \leq 60M_\odot$. We consider non-spinning BH, so the dimensionless spin magnitudes χ_1 and χ_2 are set to 0. The phase at coalescence, the polarization angle are drawn uniformly in $(0, 2\pi)$, and the inclination angle in $(0, \pi)$. Since the focus is on a single detector, the right ascension and declination are not particularly important and are thus fixed to zero.

The amplitude of the added signals is computed such that the corresponding *optimal* SNR ρ_{opt} is uniformly distributed between 8 and 20. Following [46], it is defined as

$$\rho_{opt}^2 = 4 \int_0^\infty \frac{|\tilde{h}(f)|^2}{S_n(f)} \mathrm{d}f, \tag{1}$$

where $\tilde{h}(f)$ denotes the Fourier transform of the template $h(t)$ and $S_n(f)$ is the power spectral density of the detector noise. To generate the signals, a fiducial luminosity distance d_L of 100 Mpc is initially chosen, and then scaled to obtain the desired ρ_{opt}. The final values of d_L range from 1 to 1300 Mpc approximately.

The simulated signals are added at a random position within the segment, while ensuring the chirping part of the signal is completely contained in the segment. The final part of the signal is randomly shifted between -0.25 s and 0.3 s with respect to the center of the one-second segment. A total of 750,000 signal samples are generated.

Overall, the training set consists of $250,000$ segments for the noise class, the same number for the signal class, and $70,000$ for the glitch class. A 20% fraction of the training set is allocated for validation. The testing set, used to evaluate the classifier, comprises $500,000$ samples for both the noise and signal classes, and $80,000$ for the glitch class. This ensures sufficient statistical data for characterizing the classifier's performance. In total, the training and testing datasets comprise 1,650,000 one-second segments, with 45% for the noise class, 45% for the signal class, and 10% for the glitch class. This amounts to a storage space of 26 gigabytes. Out of the total number of segments, 28% is utilized for training, 7% for validation, and 65% for testing.

3. Classifier architectures

This section discusses the type of neural network architectures considered in this study. Similarly to other works [29–34] the classifier is directly fed by the one-second segment of strain time series, so a vector size of 2048. We experiment[4] with three different network architectures, namely the CNN, as well as two other architectures specialised for time-series classification: Temporal Convolutional Network (TCN) [37] and Inception Time (IT) [38]. The last two are specialised for time series classification and, to our knowledge, have never been tested with this type of problem. The architectures are described in more detail in the following subsections. The model hyperparameters are tuned after a coarse exploration of the parameter space.

3.1. Convolutional Neural Network (CNN)

CNN were first introduced for image classification [28]. They are now used for a wide variety of tasks, including the detection of GW signals [29–34]. In this study, we tested a range of CNNs similar to those considered in previous works.

We limited ourselves to shallow networks with five layers, four convolutional layers, and one final fully connected embedding layer. For simplicity, we only report here on the best-performing CNN, whose structure is detailed in Table 1.

The convolutional layers are defined by the number of output filters, the length of the 1D convolution window (kernel size), the stride length of the convolution, and the activation function. The dense layer only requires the definition of the activation function. The input of inner convolutional layers is downsampled with a max pooling operation over a window size indicated in the table. The output of convolutional layers is

[4] Implementations are based on the `TensorFlow` library [47] with the `Keras` API [48].

processed by a dropout layer that randomly sets the input units to 0 with the frequency rate specified in the table. A global average pooling, followed by a dropout with a rate of 10%, is applied to the output of the last convolutional layer.

Table 1. Structure of the CNN considered in this study. The type of the layer is either convolutional (Conv) or fully connected (Dense). The activation function is either the rectified linear unit (`relu`) or the `softmax` function [28].

Layer number	1	2	3	4	5
Type	Conv	Conv	Conv	Conv	Dense
Number of filters	256	128	64	64	-
Kernel size	16	8	8	4	-
Stride length	4	2	2	1	-
Activation function	`relu`	`relu`	`relu`	`relu`	`softmax`
Dropout rate	0.5	0.5	0.25	0.25	-
Max pooling	4	4	2	2	-

3.2. Temporal Convolutional Network (TCN)

TCN [37, 49] is a neural network architecture specifically developed for sequence modeling problems. TCN has been shown to outperform generic state-of-the-art architectures over a diverse range of tasks and datasets. The TCN architecture is based on causal convolutions, where an output at time t is only convolved with past inputs from the previous layer. This allows the network to collect information from further in the past, using a combination of deeper networks (augmented with residual layers) and dilated convolutions.

In this study, we have tuned the hyperparameters of the TCN model to find a compromise between the best performance and a reasonable training time. We ended up using a network with a TCN layer consisting of $N = 6$ dilated convolutional layers with 32 filters, a kernel size of $k = 16$, default values of dilation factors $d_{k=1\ldots6} = (1, 2, 4, 8, 16, 32)$ for the 6 convolutional layers, and a dropout rate of 0.1. The output of the TCN layer goes into a final dropout layer with a rate of 0.5, and a dense embedding layer closes the model.

A key parameter that governs the training efficiency is the receptive field, that is the size of the region in the input data that produces a given feature in the output. The receptive field of the TCN can be expressed as $R = 1 + 2\,(k-1)\,d_{\rm tot}$ where $d_{\rm tot} = \sum d_k$ [37]. With the above configuration, we have $R \approx 1900$. The data used in this work have a sampling rate of 2048 Hz so each segment of data has 2048 points. The training with TCN is effective when R is much larger than the length of input sequence [37]. To satisfy this constrain, only for this model, it is necessary to downsample the input data to 1024 Hz, therefore producing an input vector of size 1024.

3.3. Inception Time (IT)

IT [38] is a deep network ensemble designed specifically for time series classification. It leverages the concept of residual networks and incorporates Inception module [50]. In a nutshell, the Inception module first produces a one-dimensional summary of the input multivariate time series (this is the "bottleneck" layer), and then convolves this summary through multiple filters of different lengths, leading to a multivariate output that provides inherently multi-resolution features. The module output is finally reduced by max pooling (pool size of 4) before passing to the next module.

The IT architecture is composed of five ResNet networks with a sequence of depth d Inception modules, with two residual blocks. The outputs of the five models are combined through a global average pooling and a final softmax layer, used to produce the classification probabilities for the different classes. In this study we have used the standard implementation of IT provided by the authors [38] with networks of depth $d = 10$, each with a bottleneck size of 32 processed through 32 filters with kernel sizes 20, 40 and 80.

3.4. Training process

The three classifiers are optimized using the training set described in Sec. 2 to minimize the categorical cross-entropy loss function. The default implementations of the Adam optimizer are utilized, with a batch size of 24 [47]. This training procedure is repeated 10 times, and the instance exhibiting the best ROC curve on the testing dataset (as explained in Sec. 4) is chosen.

Throughout the training process, the model's area under the ROC curve [51] is evaluated on the validation data, and the model with the highest value is ultimately selected. The CNN, TCN, and IT models are trained for 50, 150, and 20 epochs, respectively. The best models are obtained at the 24th epoch for CNN, the 34th epoch for TCN, and the 5th epoch for IT[5]. On the Tesla K40d GPU we used, the training times per epoch were 220 seconds for CNN, 1000 seconds for TCN, and 3320 seconds for IT.

3.5. Decision statistic

The final objective is to detect with high confidence the segments with a true astrophysical signal, i.e., to classify them as signal and to reject the other segments as noise or glitch.[6] We aim to constrain false alarms to a rate of two per day (similar to the current on-line search pipelines). This implies that we should reject all but one noise or glitch segment from the testing set in 1.7×10^5 trials.

[5] After the 5th epoch, the IT model displayed signs of overfitting.

[6] The two classes noise and glitch will be later merged *a posteriori* into a single class associated with the absence of an astrophysical signal.

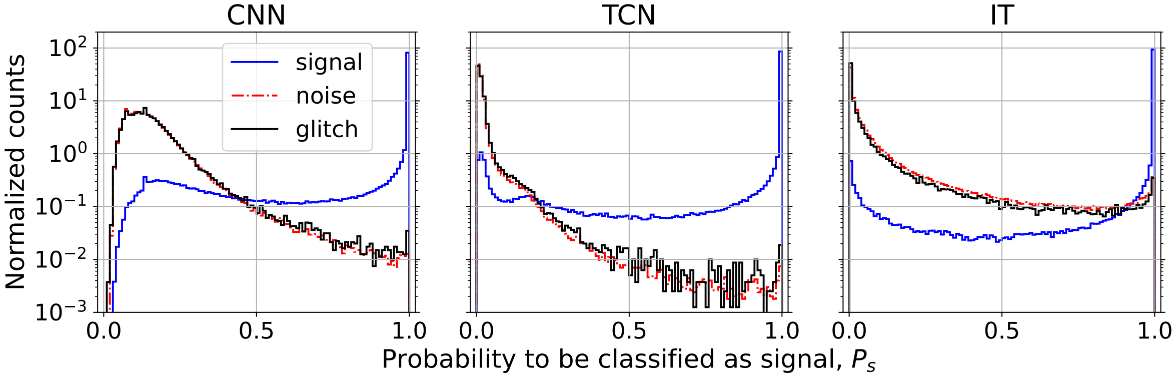


Figure 2. Distributions of P_s (the class membership probability assigned to the signal class), conditioned on the class of the input segment from the testing set: signal (blue), noise (dot-dashed red), or glitch (black). These distributions were computed for the CNN (left), TCN (middle), and IT (right) architectures. The histograms are normalized to have a unit sum. The classifiers do not distinguish between samples from the noise and glitch classes, thus resulting in practically identical probability distributions (see Sec. 4 for a discussion on this point).

The classifiers output the probability of class membership for each of the classes, that is three numbers between 0 and 1, summing to 1. The final detection is performed by applying a threshold to the membership probability P_s assigned to the signal class, which thus defines our *decision statistic*. The class membership probability is computed by the softmax activation function applied to the raw output (the "logits tensor") of the fully connected embedding layer which concludes the classifiers. Because of the high-confidence level required, this threshold is very close to 1, thus requiring attention on the numerical precision for the evaluation of the membership probability (This was already noted in [34]). This has consequences on the way the classification loss is computed from the membership probability at the training stage. We found that the categorical cross-entropy loss should be directly computed from the logits tensor rather than from the class membership probability after the softmax transformation.

This numerical precision issue has an impact on the performance both the TCN and IT classifiers. The right panel of Fig. 4 provides an illustration for IT. This plot compares the ROC curves based on the detection statistic P_s (see Sec. 4 for the details on how the curves are computed) obtained with IT with softmax (red) and without (green). The shaded area represents the range between the best and worst models among the 10 instances computed at training. When softmax is used the uncertainty in the performance is larger (the shaded area is wider) and the classification efficiency reached at small false alarm rates is lower.

4. Classifier evaluation with the testing data

This section describes the results obtained with the three classifiers presented above applied to the testing set.

The classifiers all exhibit poor separation power between the noise and glitch classes. This can be attributed to several factors, including the absence of a distinct boundary between the two classes (potentially due to contamination and mislabeling), the considerable variation in glitch morphology, and the relative class imbalance with 3.5 times the glitch class being unrepresented by a factor of 3.5 compared to the noise class in the training set. The initial assumption that a three-class division would enhance classification performance turned out to be incorrect. Consequently, we proceed by combining the noise and glitch classes into a single class representing the absence of an astrophysical signal.

4.1. Noise rejection

We first assess the noise rejection capabilities of the classifiers. Fig. 2 compares the distributions of the decision statistic P_s (the membership probability assigned to the signal class) when the input segment belongs to each of the three classes. The P_s distributions obtained with samples from the noise or glitch classes have very similar shapes, reflecting the intrinsic similarity of those two classes (see above). The best classifier is the one that provides the greatest contrast between the distributions of the P_s statistic obtained in the presence of a signal (blue) versus noise or glitch (red dot-dashed and black).

The distributions obtained for the noise or glitch classes exhibit maxima at zero for the TCN and IT classifiers, while the maximum is shifted to around 0.1 for CNN. Moving from the peak to higher values, the distribution shows a monotonic decay for CNN and TCN. However, for IT, the distribution initially decreases and then slightly increases near $P_s = 1$. The TCN classifier appears to reach the lowest background $\lesssim 10^{-2}$ in normalized count units.

Since our objective is to achieve high-confidence classification, we are primarily interested in the immediate vicinity of $P_s = 1$. This motivates to reparameterize the P_s statistic as $\lambda := -\log_{10}(1 - P_s)$. While P_s ranges from 0 to 1, λ can theoretically take values across the entire real line. However, our main focus lies in the range $\lambda \gtrsim 7$. The most stringent criterion is to require $P_s = 1$ at machine precision, which corresponds to $\lambda = \infty$. The number of noise and glitch samples in the testing set that satisfy this selection criterion is 0, 1, and 2 for the CNN, TCN, and IT classifiers, respectively. Such rejection power (between 0 and 2 false alarms in 5.8×10^5 trials) is in agreement with the false-alarm rate targeted initially.

4.2. Signal extraction

We proceed to assess the classifiers' ability to extract signals. Fig. 2 illustrates the distributions of the decision statistic P_s when the input segment belongs to the signal class, represented in blue. As anticipated, all distributions exhibit a peak at $P_s = 1$. However, the peak appears narrower for the IT classifier. In order to focus on the region of interest near $P_s = 1$, we employ the λ reparametrization, as depicted in

Fig. 3. This figure also incorporates the dependencies on the signal-to-noise ratio (SNR) of the injected GW signal and the chirp mass $\mathcal{M}$ of the source binary. The distributions are computed separately for three ranges of chirp mass $\mathcal{M}$: low, mid, and high, corresponding to $\mathcal{M}$ values between 13 and 17 $M_\odot$, 17 and 21 $M_\odot$, and 21 and 26 $M_\odot$, respectively. The histograms on the right hand side are computed with the samples of the signal class that are classified with $P_s = 1$, showing their distribution in terms of SNR for the three chirp mass ranges.

It is worth noting that we have either $\lambda \lesssim 7.5$ or $\lambda = +\infty$ (i.e., $P_s = 1$). In a sense, the latter case seems to accumulate all samples with $\lambda \gtrsim 7.5$. From the left column of the figure, it is apparent that the IT classifier assigns larger λ (or P_s) values more uniformly over the full range of chirp mass and to lower SNR. In contrast, the CNN fails to do so for the lower chirp mass interval shown in blue. This is confirmed by the histograms in the right column, which indicate that the TCN and IT classifiers have a higher overall count (approximately 76%, compared to 65% for CNN) and extend to lower SNR values.

4.3. Global assessment with Receiver Operating Characteristics

To fully characterize the performance of the classifier, the noise rejection and signal extraction capabilities have to be evaluated jointly. This can be done by computing the Receiver Operating Characteristic (ROC) curves [51]. The classification efficiency S_{th}/S_{tot} and false alarm rate N_{th}/N_{tot} are evaluated from the testing set, with S_{th} and N_{th}, the number of signal samples and noise and glitch samples with a P_s value above some threshold, and S_{tot} and N_{tot} the total number of samples for each categories. Note that since each sample has a duration of 1 second, N_{tot} in intended here as the total duration in seconds of noise and glitch samples, so N_{th}/N_{tot} is measured in s^{-1}. By varying the threshold, one obtain the ROC curves in Fig. 4 which displays the classification efficiency versus the false alarm rate. The TCN and IT classifiers appear to have similar ROC curves, and show a clear improvement with respect to CNN. Fig. 4 also shows the ROC computed for two instances of the IT architecture, with and without the softmax activation during training (see Sec. 3.5 for a discussion).

Fig. 5 shows the classification efficiency for a given false alarm rate set to 10^{-5} s^{-1}, as a function of the injected SNR as defined in Eq. (1). The classifiers TCN and IT give similar efficiencies and surpass uniformly over CNN. Note that the efficiency shown in this figure is averaged over the full chirp mass range and thus do not show the differences evidenced in Fig. 3. Overall, signals with SNR=10 can be detected at the considered significance level with a good probability, larger than 50%.

5. Application to the remaining O1 single-detector data

This section presents the results of applying the different classifiers to the remaining O1 data from the Livingston detector. Our primary focus is on the IT classifier, while the

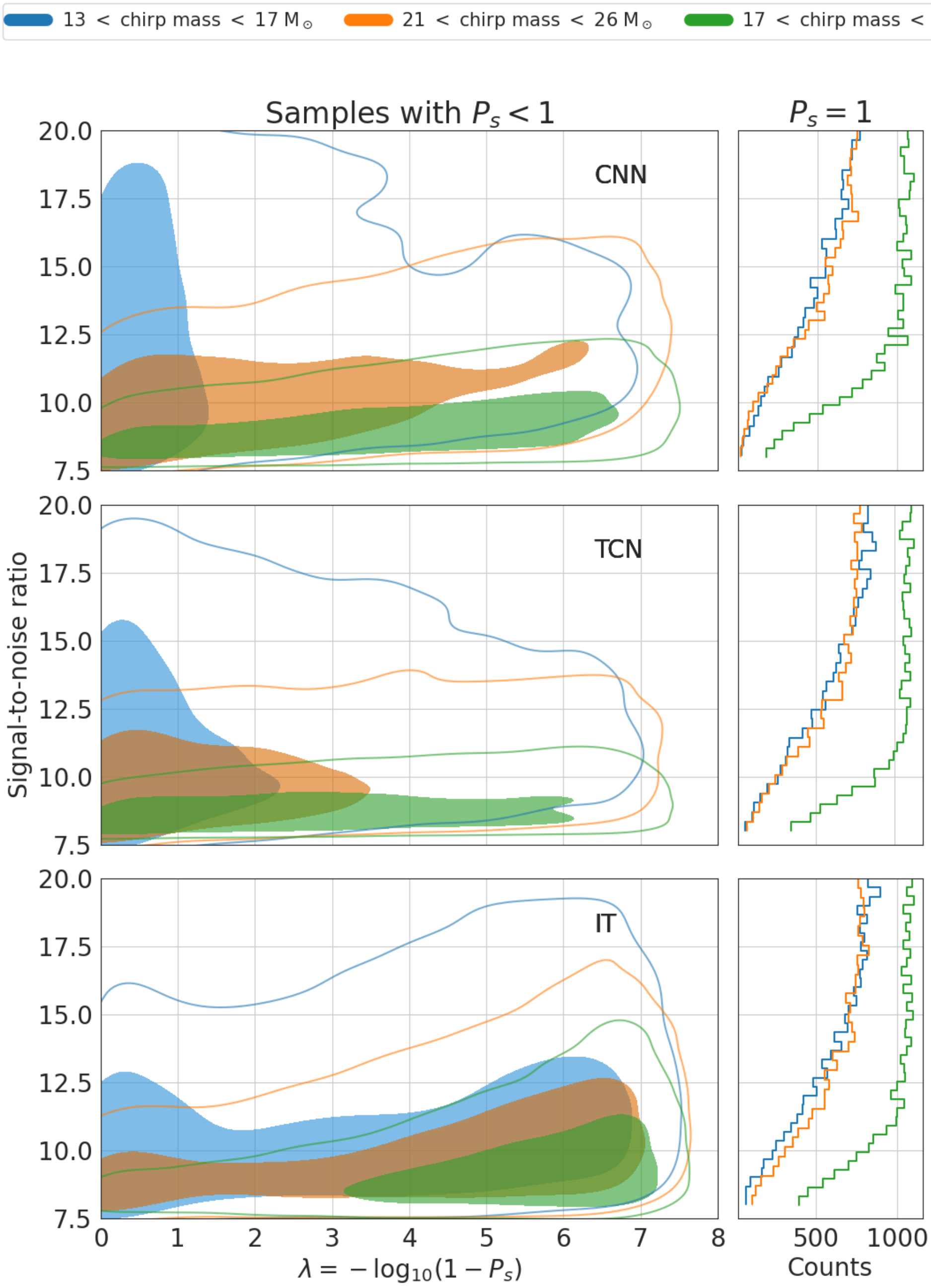


Figure 3. Distribution of the statistic $\lambda := -\log_{10}(1 - P_s)$ obtained with testing samples from the signal class and computed for the three considered classifiers: CNN (top), TCN (center) and IT (bottom). The column on the left shows a kernel density estimate of the λ distribution for the samples with $P_s < 1$, thus leading to a finite value for λ. The shaded area is the 50 % containment region, and the line is the 90 % containment region. Those distributions are shown versus the SNR of the injected GW signal and computed separately for three ranges of chirp mass. The column on the right shows an histogram for the samples with $P_s = 1$. The signal samples detectable with high confidence fall in the range of large $\lambda \gtrsim 7$ (i.e, P_s values very close or equal to 1).

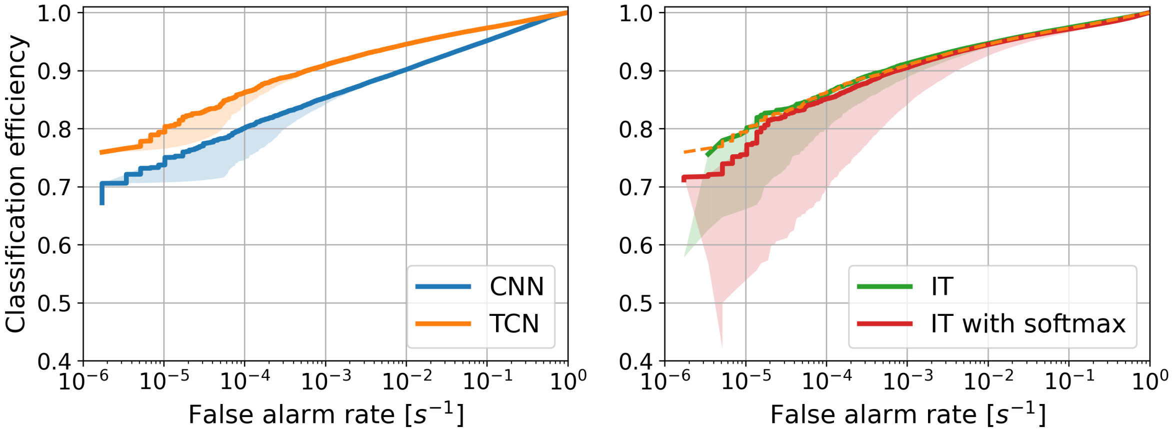


Figure 4. ROC curves for the three considered classifiers, CNN, IT, and TCN, illustrating the classification efficiency versus the false positive rate. Each classifier has been trained 10 times, and the continuous line represents the result obtained for the best model, while the shaded area covers the range from the best to the worst model. The left panel displays the TCN (orange) and CNN (blue) ROC curves. In the right panel, the ROC curves are shown for two instances of the IT architecture: one trained with softmax activation (red) and another without softmax activation (green) (refer to Sec. 3.5). The TCN ROC curve is reproduced in this panel as a dashed orange line to facilitate comparison.

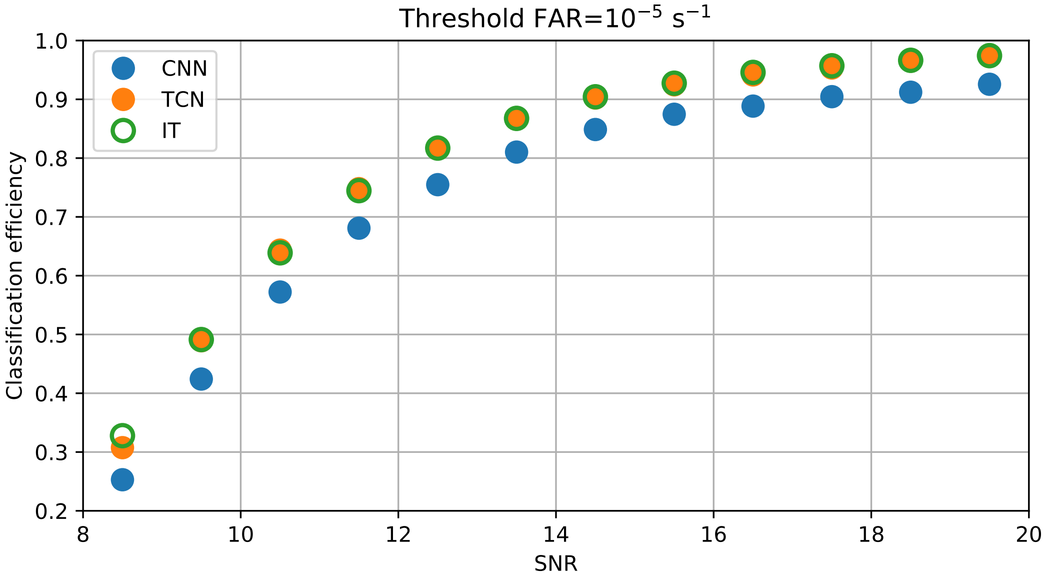


Figure 5. Classification efficiency versus SNR for a false alarm rate of 10^{-5} s^{-1}. The classifiers TCN and IT give similar efficiencies and surpass uniformly over CNN. Overall, signals with SNR=10 can be detected at the considered significance level with a good probability, larger than 50%.

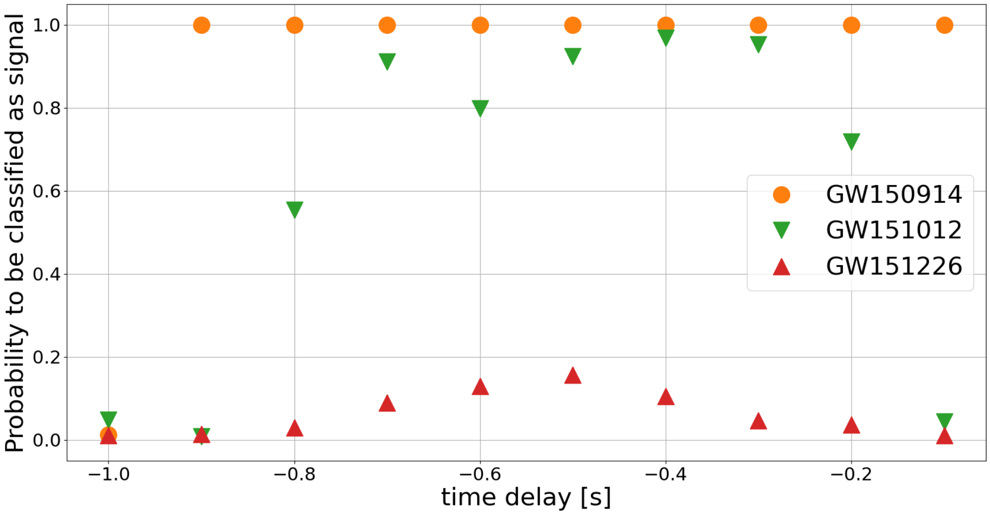


Figure 6. Evolution of the statistic P_s produced with IT classifier versus the relative delay Δt of the analysis window to the O1 event merger time (GW150914, GW151012 and GW151226). For $\Delta t = -1$ s, the analysis window only includes the initial part of the signal (inspiral), whereas, for $\Delta t = 0$ s, the analysis window starts at the merger time and thus only includes the final part (merger and ringdown).

results for the other models can be found in Appendix A.

5.1. *Analysis of known O1 GW events*

We first investigate how the three events detected in the O1 data [8] by matched filtering searches are classified by the considered models. The statistic P_s is evaluated for different positions of one-second window, that is for different time delays Δt between the start of analysis window and the merger time. This definition implies that, for $\Delta t = -1$ s, the analysis window only includes the initial part of the signal (inspiral), whereas, for $\Delta t = 0$ s, the analysis window starts at the merger time and thus only includes the final part (merger and ringdown). Fig. 6 shows the evaluation of P_s between those two extreme cases for the IT model for GW150914, GW151012 and GW151226 (see also Appendix A).

As expected, when the chirp signal is not included in the analysis window, the classifier is not able to detect the presence of the signal. GW150914 appears to be loud enough to be always identified, regardless of its position in the time window, even if it is partially visible. GW151012 is only detected when the chirp is at the center of the analysis window. GW151226 is not detected. This is expected as the binary component masses are outside the range used to generate the astrophysical signals in the signal class of the training data. It also has a low SNR in Livingston.

5.2. Analysis of the remaining O1 data

We analysed all the remaining L1 data in O1 excluding the month we used to train and test the classifiers (see Sec. 2). This corresponds to the period between GPS=1126051217 (2015-09-12 00:00:00 UTC) and GPS=1132444817 (2015-11-25 00:00:00 UTC) and between GPS=1135036817 (2015-12-25 00:00:00 UTC) and GPS=1137254417 (2016-01-19 16:00:00 UTC). In this period we excluded the intervals of ± 1 second around the chirp time of the 3 known events (see previous section). This amounts to a total of 4,216,489 s (about 49 days), of which 1,054,564 s (about 12 days) are single-detector times, corresponding to the 25% of the total. This data set is whitened following the same procedure used to produce the training set (the ASD was calculated from periods of non-interrupted data taking with 26 s minimum and 146,978 s maximum). The data are then divided into non-overlapping one-second segments that are processed through the three classifiers. For each, we used the best performing model on the testing data.

Fig. 7 shows the distribution of the $\lambda = -\log_{10}(1 - P_s)$ statistic obtained with the IT classifier (similar plots can be found in Appendix A for the other models). We apply the most restrictive selection cut, by requiring with $P_s = 1$ (at machine precision). We recall that this selection cut corresponds to a false-alarm rate of $\lesssim 4 \times 10^{-6}$ s^{-1} (that is one false alarm per 3 days) and a classification efficiency of 76% when estimated on the testing set, see Sec. 4.1 and 4.2. Based on these results, we estimate from basic counting statistics that the maximum number of false alarms expected for this analysis should be 29, 43 and 55 for CNN, TCN and IT respectively at 95% level for the full data set, and 9, 13 and 16 when restricting to the single-detector part.

For the IT classifier, a total of nine segments pass the selection cut, with two occurring in single detector time at GPS=1131289775 (2015-11-11 15:09:18 UTC) and GPS=1135945474 (2016-01-04 12:24:17 UTC). For the CNN and TCN classifier, we obtain 4 and 105 segments passing the cut, with 2 and 14 falling in single detector periods. The result are thus consistent with the expectations for CNN and IT, while there is a clear excess with TCN. We have observed that a significant fraction of the triggers comes from two time intervals around 2015-10-20. Our interpretation is that the data from those periods could differ in nature from those of the training set, and TCN may be sensitive to this difference.

Interestingly there is only one segment passing the selection cut for all the three classifiers: GPS=1135945474 (2016-01-04 12:24:17 UTC) which we investigate further in the next section.

5.3. Detailed analysis of the 2016-01-04 event

We have performed a number of detailed checks of the 2016-01-04 event. We have performed a "visual" inspection with the time-frequency Q-transform [53]. Fig. 8 provides a time-frequency representation of the entire segment with a Q-scan [53]. A transient is visible ∼ 0.35 seconds after the start of the segment, at a frequency of

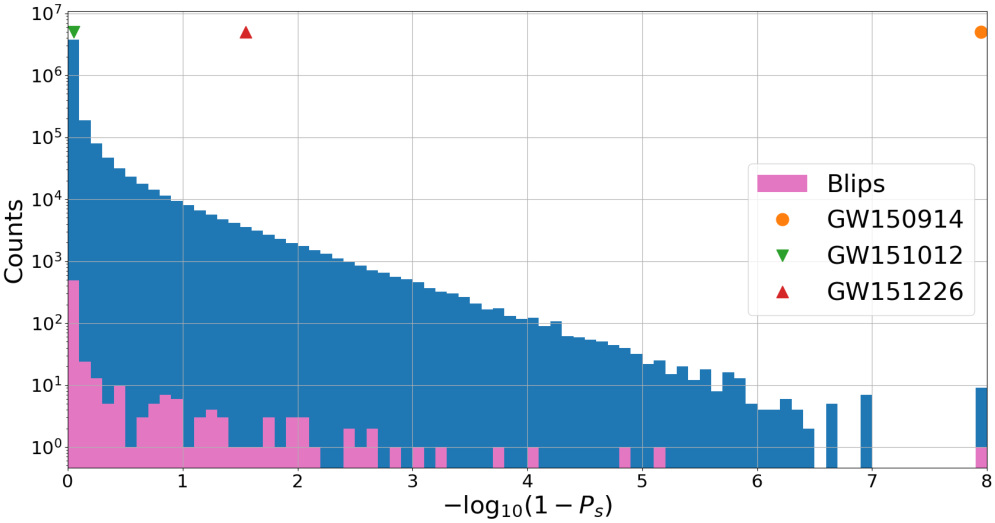


Figure 7. Distribution of the $\lambda = -\log_{10}(1 - P_s)$ statistic (shown in blue) obtained using the IT classifier on the remaining O1 dataset (refer to Sec. 5.2 for details). The segments with $P_s = 1$ have been assigned a value of $\lambda = 8$ for plotting purposes. The pink histogram corresponds to a subset labeled as "blip" glitch by `Gravity Spy` [52]. The markers at the top indicate the highest values for the three O1 events displayed in Fig. 6. Please note that the vertical position of these markers is arbitrary.

about 150 Hz. In the magnified view, the shape of the transient is clearly indicative a frequency modulated chirp-like transient.

The `Gravity Spy` database [52] has marked this specific GPS time classified as being an instrumental artefact of the "Blip" type. The term Blip refers to a well identified family of instrument glitches whose origin is still unknown. Generally, "blip" glitches do not exhibit a chirping frequency (see Fig. 1 of [54] for a typical example). To complement this initial inspection, Fig. 7 gives in pink the statistic λ (or equivalently P_s) of the 600 blip glitches listed in `Gravity Spy` overlapping with the part of the O1 dataset being analyzed. The resulting distribution is compatible with the overall background distribution. The Jan 6 segment appears to be an outlier with respect of the blip glitches identified in the data.

Further, we checked if the transient signal can be fitted by a GW waveform model associated to a compact binary merger. To do so, we ran the Bayesian inference library `Bilby` [55] and used the `IMRPhenomXPHM` waveform model [56]. It is assumed that the component spins are co-aligned with the orbital momentum. For the rest of the source parameters, generic and agnostic priors are assumed, along with a standard Λ-CDM cosmology model with $H_0 = 67.9\,\mathrm{km\,s^{-1}\,Mpc^{-1}}$ [57]. The analysis did not include a marginalization over calibration uncertainties. The analysis results in a signal-versus-noise log Bayes factor of 47. The estimated time of arrival of the merger at the detector is GPS=$1135945474.373^{+0.076}_{-0.07}$ and the measured optimal SNR is $11.34^{+1.8}_{-1.6}$.

Fig. 9 shows the result of the fit in the time domain, by comparing the whitened

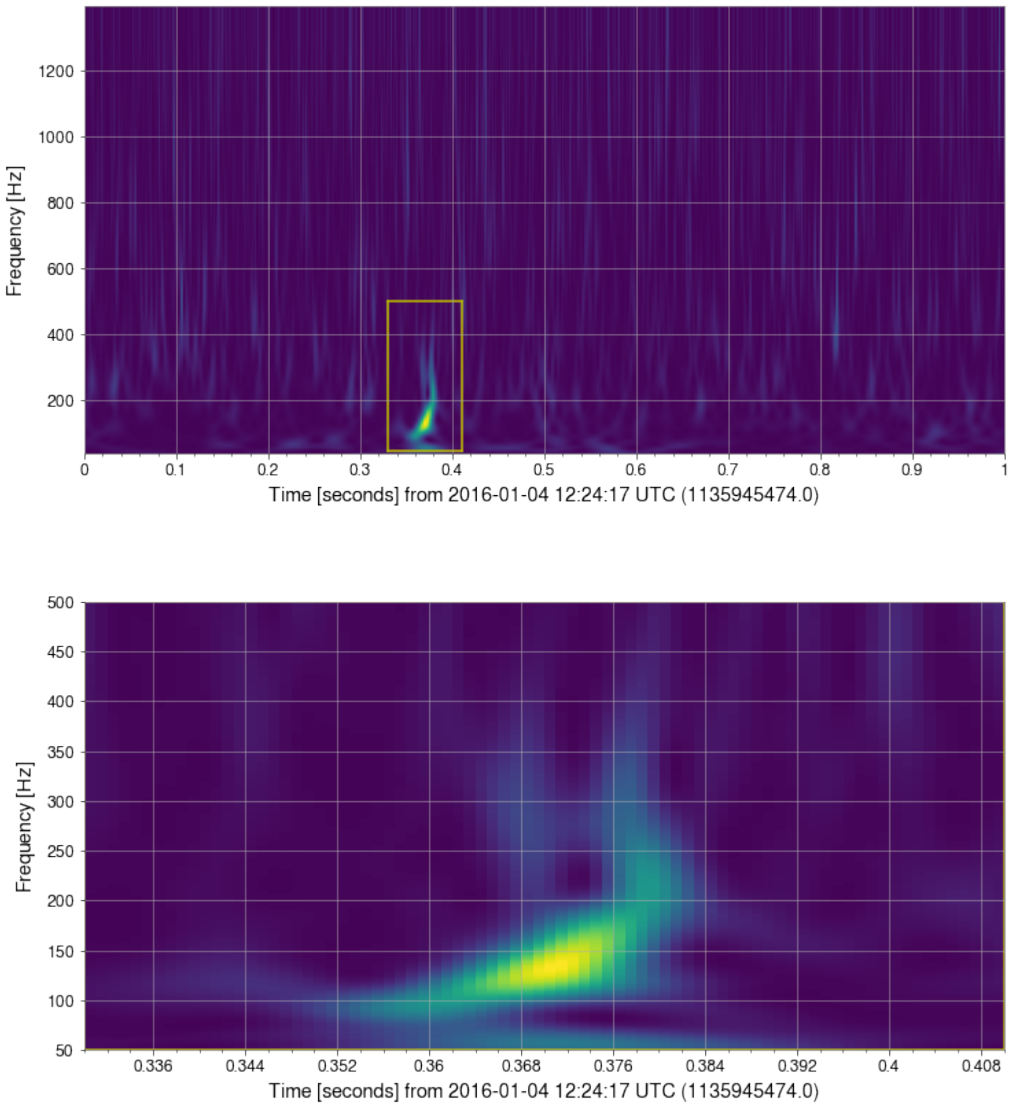


Figure 8. Time-frequency representation of the segment at 2016-01-04 12:24:17 UTC (GPS=1135945474 s) recorded by the LIGO Livingston detector. The top panel shows the entire segment. The bottom panel is a detailed view that focuses on the transient signal at $t \sim 0.35$ and $f \sim 150$ Hz. The frequency of the signal is distinctly increasing in a chirping pattern.

data in orange to the inferred waveform (blue) with a 90% credible belt. We report that there is no significant residual after substraction of the inferred waveform. As an independent check of the nature of the signal, the figure also includes the waveform estimate produced by the denoising convolutional autoencoder described in [58] (dashed red). The two reconstructed waveforms are in good agreement, following a similar phase evolution, except for the initial and final parts of the signal, where the denoiser's reconstruction is not optimal because of its low-frequency cut-off, and the rather low

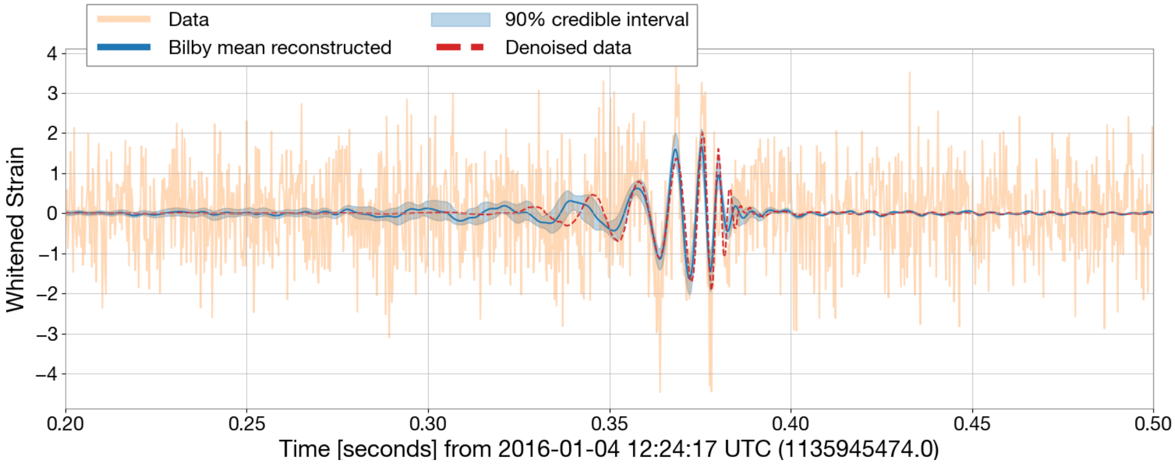


Figure 9. Comparison of the whitened L1 data (orange line) with the reconstructed waveform from the `Bilby` posteriors (blue) and from the ML denoising convolutional autoencoder neural network described in [58] (dashed red line).

SNR of the signal. In addition, we note that the denoising autoencoder was trained on the waveform family `SEOBNRv4` which is different than the one used for `Bilby` (`IMRPhenomXPHM`), which may contribute to differences.

Above checks are all compatible with the event being of astrophysical origin. The corner plot in Fig. 10 displays the posterior distribution of the source parameters including the binary component masses, spins and source distance. Since only one detector is available, the source direction is not localized in the sky. The 90% credible intervals for those parameters are: the measured (redshifted) chirp mass $\mathcal{M} = 30.18^{+12.3}_{-7.3} M_\odot$, the (redshifted) component masses $m_1 = 50.7^{+10.4}_{-8.9}\, M_\odot$ and $m_2 = 24.4^{+20.2}_{-9.3}\, M_\odot$, the binary effective spin $\chi_{\rm eff} = 0.06^{+0.4}_{-0.5}$ and the luminosity distance $d_L = 564^{+812}_{-338}$ Mpc; see [59] for a definition of those physical parameters. Overall, these values are consistent with the observed population of BBH to date.

6. Conclusions

This contribution demonstrates the viability of training neural network classifiers on real (non-simulated) data for analyzing single-detector observing periods of ground-based GW detectors. We have shown that architectures specifically designed for time-series classification, such as IT or TCN, outperform the standard CNN typically used so far. These classifiers, when applied to data from the Livingston detector during the three-month period of the first observing run (O1), jointly detect a plausible signal of astrophysical origin on January 6, 2016, which has not been previously recorded to our knowledge.

Operationally, we propose an approach where the multiple detector data from the first month of an observing run, labeled by standard matched filtering-based pipelines, are used to train the neural network models. The resulting classifiers can then be

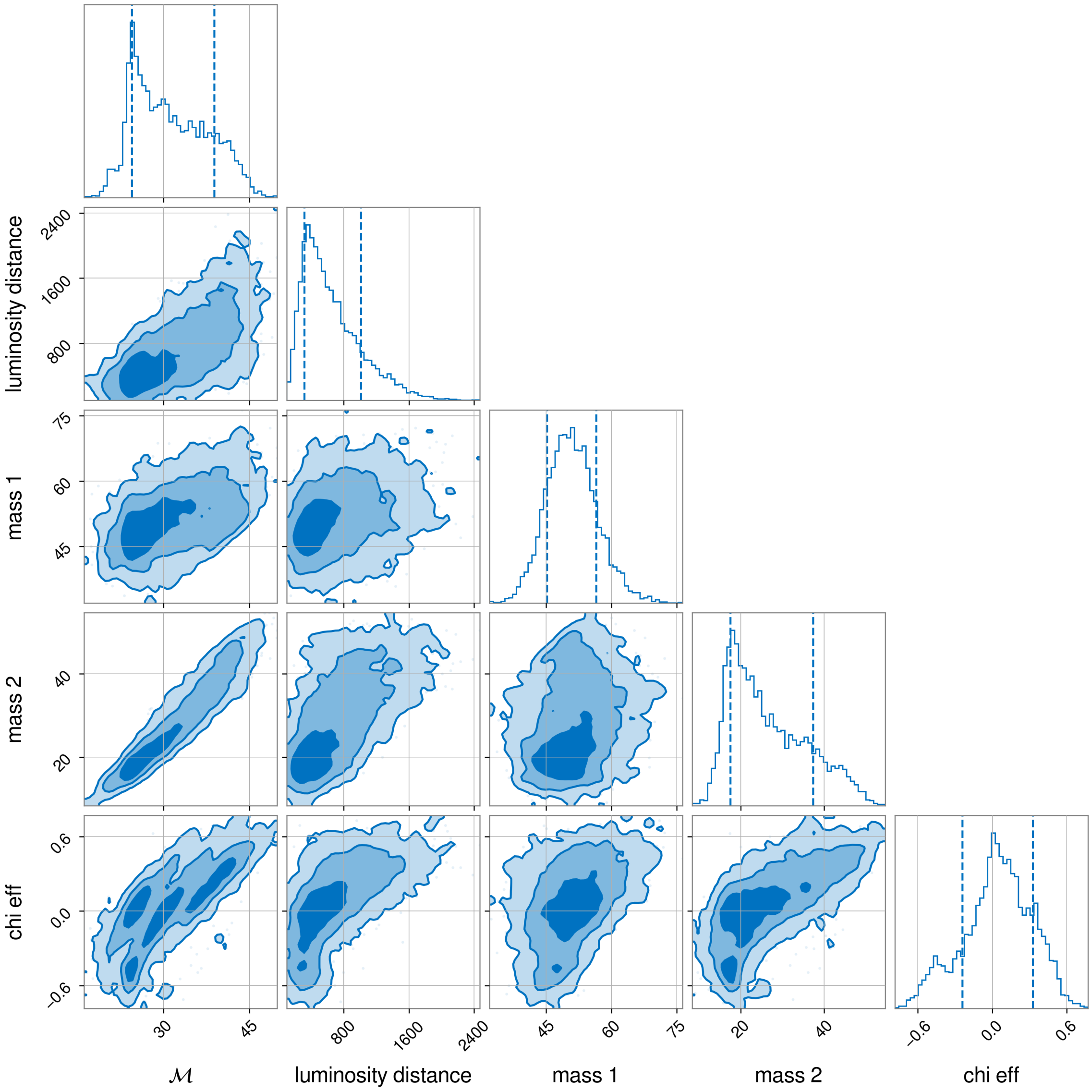


Figure 10. Posterior distribution of the chirp mass $\mathcal{M}$, luminosity distance, component masses m_1 and m_2, and effective spin $\chi_{\rm eff}$ for the 2016-01-04 event (see Sect. 5.3 for details).

applied to the remaining data collected during single-detector periods. Once trained, the computational cost is such that the classifiers can produce low-latency triggers. However, the poor sky localization obtained with only one detector limits the relevance of this approach.

The current approach faces two limitations: (i) using real data for training and testing inherently limits the statistical characterization of these algorithms and their noise rejection capabilities, as already highlighted in [35] and observed with the excess of triggers produced by the TCN classifier; (ii) there is a technical issue arising from the use of bounded selection statistics (i.e., class membership probabilities in our case) that leads to numerical intricacies. More generally, due to the absence of a mathematical theory for neural networks, their precise statistical characterization on noisy data remains an open question. Consequently, research in this field is limited to a trial and error heuristic approach.

This contribution opens up new possibilities for analyzing the fairly large single-detector data set. Applying the proposed classifiers to other LIGO-Virgo observations and broadening the parameter space to include lower masses and effects such as higher-order modes or precession would be interesting directions for future work.

7. Acknowledgements

The authors would like to thank Liliia Sinitsyna, Anirudh Kalla, Hugo Marchand and Félix Bretaudeau that have contributed to this project during their internship. É CM would like to thank Cecilio Garcia-Quiros for useful advices with the usage of `Bilby`, and Konstantin Leyde for stimulating discussions about this work. MB gratefully acknowledges partial support from the Polish National Science Centre grants no. 2016/22/E/ST9/00037 and 2021/43/B/ST9/01714, and Poland's high-performance Infrastructure PLGrid (ACK Cyfronet AGH). This work was partially supported by European Union's Horizon 2020 research and innovation programme under grant agreement No 653477, diiP (data intelligence institute of Paris), IdEx Université de Paris, ANR-18-IDEX-0001, and the COST action G2net (CA 17137).

This research has made use of data or software obtained from the Gravitational Wave Open Science Center (gwosc.org), a service of the LIGO Scientific Collaboration, the Virgo Collaboration, and KAGRA. This material is based upon work supported by NSF's LIGO Laboratory which is a major facility fully funded by the National Science Foundation, 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. KAGRA is supported by Ministry of Education, Culture, Sports, Science and Technology (MEXT), Japan Society for the Promotion of Science (JSPS) in Japan; National Research Foundation (NRF) and Ministry of Science and ICT (MSIT) in Korea; Academia Sinica (AS) and National Science and Technology Council (NSTC) in Taiwan.

The authors are grateful for computational resources provided by the LIGO Laboratory and supported by National Science Foundation Grants PHY-0757058 and PHY-0823459.

References

[1] Abbott, B P *et al* (LIGO Scientific Collaboration, Virgo Collaboration) 2016 *Phys. Rev. Lett.* **116** 061102

[2] Aasi, J *et al* (LIGO Scientific Collaboration) 2015 *Class. Quantum Grav.* **32** 074001

[3] Acernese, F *et al* (Virgo Collaboration) 2015 *Class. Quantum Grav.* **32** 024001

[4] Abbott, B P *et al* (LIGO Scientific Collaboration, Virgo Collaboration) 2017 *Phys. Rev. Lett.* **119** 161101 (*Preprint* 1710.05832)

[5] Abbott, B P *et al* (LIGO Scientific Collaboration, Virgo Collaboration) 2020 *ApJ* **892** L3 (*Preprint* 2001.01761) URL https://doi.org/10.3847/2041-8213/ab75f5

[6] Abbott, R *et al* (LIGO Scientific Collaboration, Virgo Collaboration) 2020 *The Astrophysical Journal* **896** L44 URL https://doi.org/10.3847/2041-8213/ab960f

[7] Abbott, R *et al* (LIGO Scientific Collaboration, Virgo Collaboration) 2021 *ApJ Lett.* **915** L5 (*Preprint* 2106.15163)

[8] Abbott, B P *et al* (LIGO Scientific Collaboration, Virgo Collaboration) 2019 *Phys. Rev. X* **9** 031040 (*Preprint* 1811.12907)

[9] Abbott, R *et al* (LIGO Scientific Collaboration, Virgo Collaboration) 2021 *Phys. Rev. X* **11** 021053 (*Preprint* 2010.14527)

[10] Abbott, R *et al* (LIGO Scientific Collaboration, Virgo Collaboration) 2021 *arXiv e-prints* arXiv:2108.01045 (*Preprint* 2108.01045)

[11] Abbott, R *et al* (LIGO Scientific Collaboration, Virgo Collaboration, KAGRA Collaboration) 2021 *arXiv e-prints* arXiv:2111.03606

[12] Abbott R *et al.* 2021 *arXiv e-prints* arXiv:2111.03606 (*Preprint* 2111.03606)

[13] Creighton J and Anderson W 2011 *Gravitational-Wave Physics and Astronomy: An Introduction to Theory, Experiment and Data Analysis* (Hoboken, US: Wiley)

[14] Davis D *et al.* (LIGO) 2021 *Class. Quant. Grav.* **38** 135014 (*Preprint* 2101.11673)

[15] Acernese F *et al.* 2022 *arXiv e-prints* arXiv:2205.01555 (*Preprint* 2205.01555)

[16] Dhurandhar S, Mukhopadhyay H, Tagoshi H and Kanda N 2011 *Int. J. Mod. Phys. D* **20** 2051–2056 (*Preprint* 1003.5490)

[17] https://www.gw-openscience.org/detector_status/O1/

[18] https://www.gw-openscience.org/detector_status/O2/

[19] https://www.gw-openscience.org/detector_status/O3a/

[20] https://www.gw-openscience.org/detector_status/O3b/

[21] Sachdev S *et al.* 2019 *arXiv e-prints* arXiv:1901.08580 (*Preprint* 1901.08580)

[22] Tsukada L *et al.* 2023 *arXiv e-prints* arXiv:2305.06286 (*Preprint* 2305.06286)

[23] Ewing B *et al.* 2023 *arXiv e-prints* arXiv:2305.05625 (*Preprint* 2305.05625)

[24] Nitz A *et al.* 2022 gwastro/pycbc: v2.0.4 release of PyCBC Zenodo URL https://doi.org/10.5281/zenodo.6646669

[25] Davies G S C and Harry I W 2022 *Class. Quant. Grav.* **39** 215012 (*Preprint* 2203.08545)

[26] Cuoco E *et al.* 2020 *Machine Learning: Science and Technology* **2** 011002 URL https://doi.org/10.1088/2632-2153/abb93a

[27] Huerta E A and Zhao Z 2020 *Advances in Machine and Deep Learning for Modeling and Real-Time Detection of Multi-messenger Sources* (Singapore: Springer Singapore) pp 1–27

[28] Goodfellow I J, Bengio Y and Courville A 2016 *Deep Learning* (Cambridge, MA, USA: MIT Press) http://www.deeplearningbook.org

[29] George D and Huerta E A 2018 *Phys. Rev. D* **97**(4) 044039 URL https://link.aps.org/doi/10.1103/PhysRevD.97.044039

[30] George D and Huerta E 2018 *Physics Letters B* **778** 64–70 ISSN 0370-2693 URL https://www.sciencedirect.com/science/article/pii/S0370269317310390

[31] Gabbard H, Williams M, Hayes F and Messenger C 2018 *Phys. Rev. Lett.* **120**(14) 141103 URL https://link.aps.org/doi/10.1103/PhysRevLett.120.141103

[32] Krastev P G 2020 *Physics Letters B* **803** 135330 ISSN 0370-2693 URL https://www.sciencedirect.com/science/article/pii/S0370269320301349

[33] Gebhard T D, Kilbertus N, Harry I and Schölkopf B 2019 *Phys. Rev. D* **100**(6) 063015 URL https://link.aps.org/doi/10.1103/PhysRevD.100.063015

[34] Schäfer M B, Zelenka O, Nitz A H, Ohme F and Brügmann B 2022 *physical-review-d* **105** 043002 (*Preprint* 2106.03741)

[35] Schäfer M B *et al.* 2023 *Phys. Rev. D* **107** 023021 (*Preprint* 2209.11146)

[36] LIGO-Virgo-KAGRA Public Alerts User Guide, https://emfollow.docs.ligo.org/userguide

[37] Bai S, Zico Kolter J and Koltun V 2018 *arXiv e-prints* arXiv:1803.01271 (*Preprint* 1803.01271)

[38] Ismail Fawaz H and Lucas B and Forestier G et al 2020 *Data Min. Knowl. Disc.* **34** 1936–1962

[39] Abbott, R *et al* (LIGO Scientific Collaboration, Virgo Collaboration) 2021 *SoftwareX* **13** 100658

[40] Macleod D M, Areeda J S, Coughlin S B, Massinger T J and Urban A L 2021 *SoftwareX* **13** 100657 ISSN 2352-7110 URL https://www.sciencedirect.com/science/article/pii/S2352711021000029

[41] LIGO Scientific Collaboration 2018 LIGO Algorithm Library - LALSuite free software (GPL) URL https://doi.org/10.7935/GT1W-FZ16

[42] Drago M *et al.* 2021 *SoftwareX* **14** 100678

[43] Klimenko S *et al.* 2021 cwb pipeline library: 6.4.0 URL https://doi.org/10.5281/zenodo.4419902

[44] Bahaadini S, Noroozi V, Rohani N, Coughlin S, Zevin M, Smith J, Kalogera V and Katsaggelos A 2018 *Information Sciences* **444** 172–186 ISSN 0020-0255 URL https://www.sciencedirect.com/science/article/pii/S0020025518301634

[45] Bohé A *et al.* 2017 *Phys. Rev. D* **95** 044028 (*Preprint* 1611.03703)

[46] Maggiore M 2008 *Gravitational Waves, volume 1: theory and experiments* (Oxford University Press) ISBN 978-0-19-857074-5

[47] Abadi M *et al.* 2015 TensorFlow: Large-scale machine learning on heterogeneous systems software available from tensorflow.org URL https://www.tensorflow.org/

[48] Chollet F *et al.* 2015 Keras https://keras.io

[49] Remy P 2020 Temporal convolutional networks for keras https://github.com/philipperemy/keras-tcn

[50] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V and Rabinovich A 2014 *arXiv e-prints* arXiv:1409.4842 (*Preprint* 1409.4842)

[51] Fawcett T 2006 *Pattern Recognition Letters* **27** 861–874 ISSN 0167-8655 rOC Analysis in Pattern Recognition URL https://www.sciencedirect.com/science/article/pii/S016786550500303X

[52] Glanzer J *et al.* 2021 Gravity Spy Machine Learning Classifications of LIGO Glitches from Observing Runs O1, O2, O3a, and O3b URL https://doi.org/10.5281/zenodo.5649212

[53] Chatterji S, Blackburn L, Martin G and Katsavounidis E 2004 *Class. Quant. Grav.* **21** S1809–S1818 (*Preprint* gr-qc/0412119)

[54] Glanzer J *et al.* 2023 *Class. Quant. Grav.* **40** 065004 (*Preprint* 2208.12849)

[55] Ashton G *et al.* 2019 *The Astrophysical Journal Supplement Series* **241** 27 URL https://dx.doi.org/10.3847/1538-4365/ab06fc

[56] Pratten G, García-Quirós C, Colleoni M, Ramos-Buades A, Estellés H, Mateu-Lucena M, Jaume R, Haney M, Keitel D, Thompson J E and Husa S 2021 *Phys. Rev. D* **103**(10) 104056 URL https://link.aps.org/doi/10.1103/PhysRevD.103.104056

[57] Ade P A R *et al.* (Planck) 2016 *Astron. Astrophys.* **594** A13 (*Preprint* 1502.01589)

[58] Bacon P, Trovato A and Bejger M 2023 *Machine Learning: Science and Technology* URL http://iopscience.iop.org/article/10.1088/2632-2153/acd90f

[59] Christensen N and Meyer R 2022 *Rev. Mod. Phys.* **94** 025001 (*Preprint* 2204.04449)

Appendix A. Additional results

In addition to the figures presented with the IT classifier in Section 5, we provide here the corresponding figures for the CNN and TCN models.

This includes the analysis conducted with known O1 GW events in Sec. 5.1. Comparing Figs. 6 and A2, we observe that GW150914 is the only event classified with $P_s = 1$ by all classifiers, while GW151012 and GW151226 never satisfy this selection criterion. For TCN, the P_s statistic is particularly low for both of these events, whereas CNN yields the highest P_s value.

We also present background histograms obtained with the remaining O1 data, similar to Fig. 7 in Sec. 5.2. Fig. A1 shows the same distribution for CNN and TCN. Although it is lower than the other two models, the distribution obtained with CNN exhibits a tail that reaches the extreme point, $P_s = 1$. CNN appears to be more sensitive to the presence of blip glitches, as the total number of blip glitches with $\lambda > 3$ is twice as high as the number in the other two models.

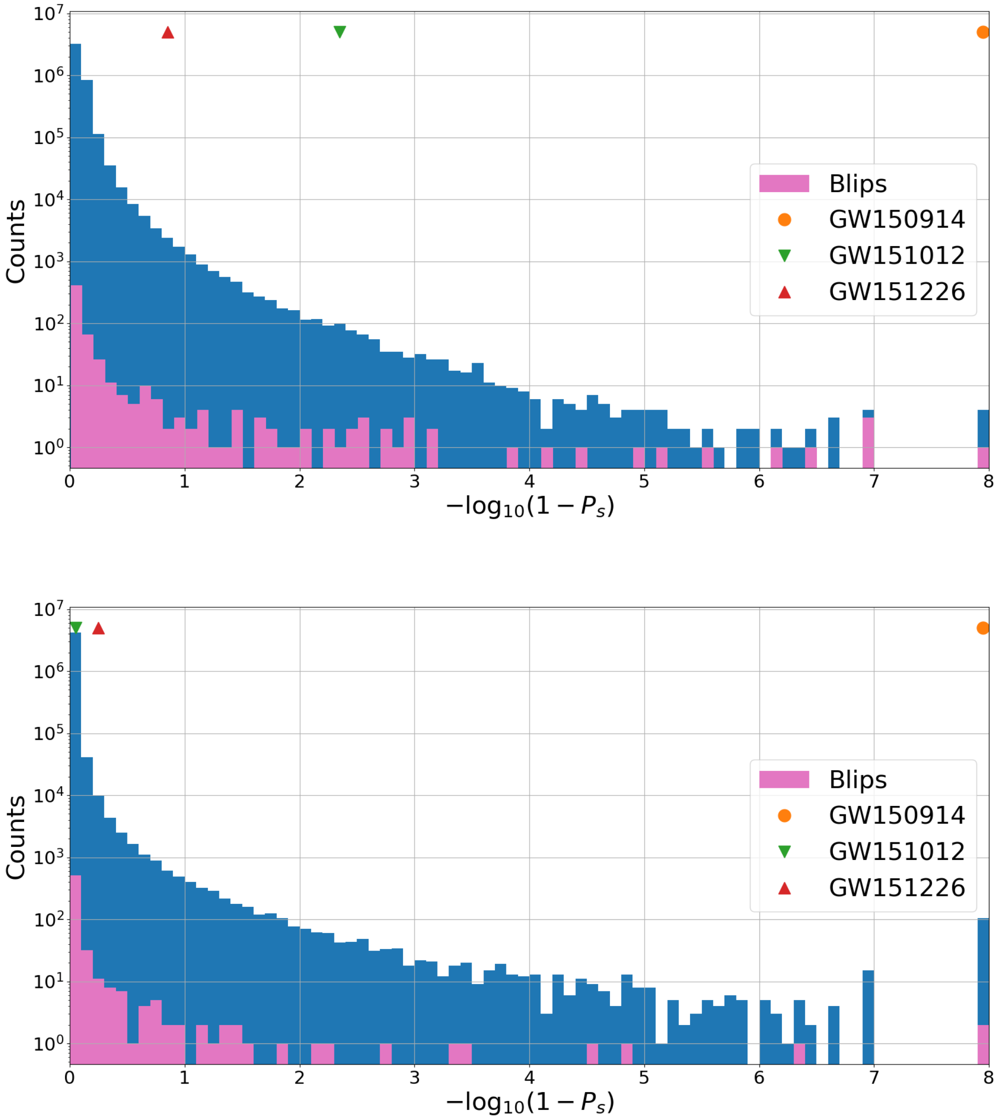
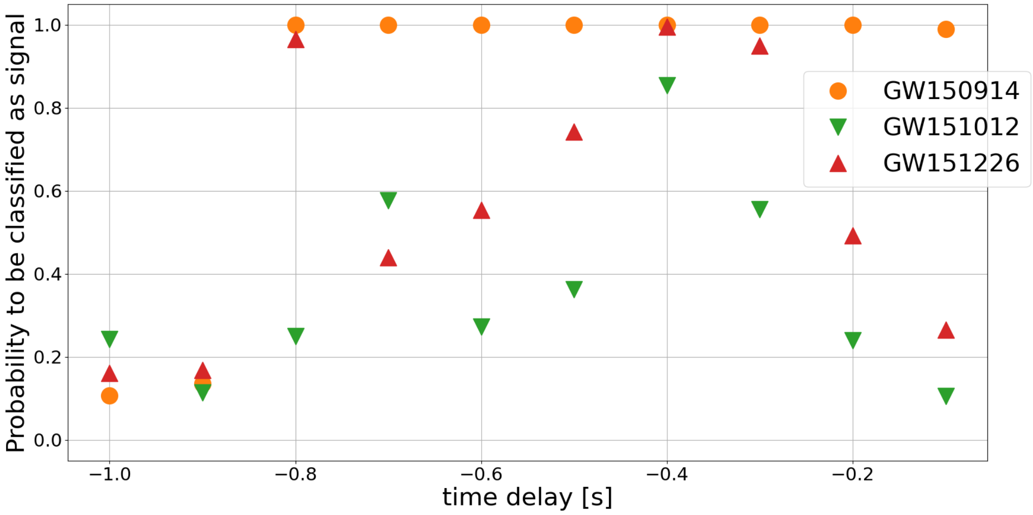


Figure A1. Evolution of the statistic P_s produced with CNN (top) and TCN (bottom) classifiers versus the relative delay Δt of the analysis window to the O1 event merger time (GW150914, GW151012 and GW151226). For $\Delta t = -1$ s, the analysis window only includes the initial part of the signal (inspiral), whereas, for $\Delta t = 0$ s, the analysis window starts at the merger time and thus only includes the final part (merger and ringdown).



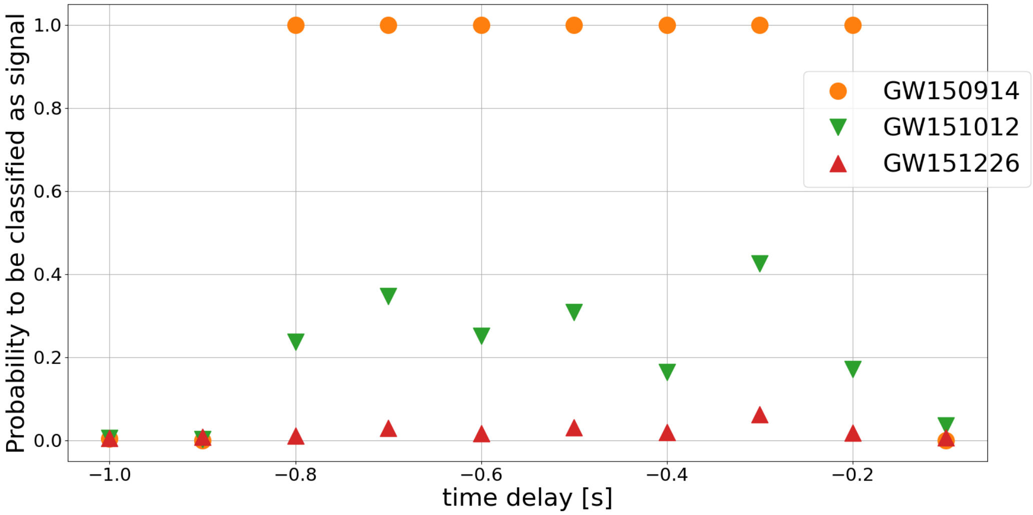


Figure A2. Evolution of the statistic P_s produced with CNN classifier versus the relative delay Δt of the analysis window to the O1 event merger time (GW150914, GW151012 and GW151226). For $\Delta t = -1$ s, the analysis window only includes the initial part of the signal (inspiral), whereas, for $\Delta t = 0$ s, the analysis window starts at the merger time and thus only includes the final part (merger and ringdown).