OPEN ACCESS
IOP Publishing

Classical and Quantum Gravity

Class. Quantum Grav. **41** (2024) 125003 (26pp)

https://doi.org/10.1088/1361-6382/ad40f0

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

A Trovato[1,2,*], **E Chassande-Mottin**[3], **M Bejger**[4,5], **R Flamary**[6] **and N Courty**[7]

[1] Dipartimento di Fisica, Università di Trieste, I-34127 Trieste, Italy
[2] INFN, Sezione di Trieste, I-34127 Trieste, Italy
[3] Université Paris Cité, CNRS, Astroparticule et Cosmologie, F-75013 Paris, France
[4] INFN, Sezione di Ferrara, I-44122 Ferrara, Italy
[5] Nicolaus Copernicus Astronomical Center, Polish Academy of Sciences, ul. Bartycka 18, 00-716 Warsaw, Poland
[6] Ecole polytechnique, IP Paris, CMAP, F-91120 Palaiseau, France
[7] Université Bretagne Sud, CNRS IRISA, F-35042 Rennes, France

E-mail: agata.trovato@units.it

Received 21 July 2023; revised 14 March 2024
Accepted for publication 19 April 2024
Published 14 May 2024

Abstract

The search for gravitational-wave (GW) 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 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. The most promising candidate from our search

[*] Author to whom any correspondence should be addressed.

is 4 January 2016 12:24:17 UTC. Although we are not able to constrain the significance of this event to the level conventionally followed in GW 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 $d_\mathrm{L} = 564^{+812}_{-338}\,\mathrm{Mpc}$.

Keywords: gravitational wave detection, machine learning, convolutional neural network, temporal convolutional network, inception time, single-detector analysis

1. Introduction

The breakthrough discovery of gravitational waves (GWs) on 14 September 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 (BBHs), they also include binary systems with neutron stars [4–7]. These detections are collected and characterised in the GW transient catalogs GWTCs [8–11]. On May 2023 the fourth observing run (O4) started with an increasing detector 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. [11] for a recent review. In particular, matched filtering [12] 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', see e.g. [13] and references therein) 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 occur sporadically in the data and that can mimic the GW signal [16, 17]. A very powerful tool to discriminate the signal from noise glitches is time coincidence across two or more separate detectors (see [18] 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 [19, 20]. During O3, thanks to a more stable and reliable operation and to the addition of a third detector to the network, this fraction was reduced to about 15% in O3a [21] and 11% in O3b [22] (the first and second six months parts of O3). In total, converting the percentages in time span, during O1, O2 and O3, for more than five months of observing time only one detector was taking data. 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 to high confidence levels. Several studies investigate ways to resolve these difficulties. Two methods [23, 24] that allow the identification of GW candidates in single-detector data have been employed in production in the context of low-latency GW searches [25], enabling the initial identification of GW170817 and GW190425. Similarly, [26] introduces a framework for assigning significance to single-detector GW events by leveraging the measured rate of BBH mergers. More recently, [27] studies the possibility to extend the multi-variate 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 [28] and one of the improvements is the addition of a tunable penalty in case of single detectors candidates to down weight their significance [29]. To extrapolate the significance measure of single-detector triggers produced by the PyCBC pipeline [30], a method proposed in [31] 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.

Despite those developments, single-detector periods have received less attention than the rest of the observations and are covered in a few studies. Following a 'multi-messenger' approach, several works looked for coincidences between data from a solitary GW detector with gamma-ray observations from the Fermi gamma-ray burst monitor [32–34]. Three searches for binary mergers in single-detector periods relied on GW data only. Magee *et al* [35] presents a search which specifically targets a narrow range of low masses motivated by the population of known double neutron-star binaries. Two contributions present the results of searches for binary mergers over the entire range from 1 to $100\,M_\odot$ for the component masses, for the observing runs O1 and O2 [36] and for O3 [31]. The former finds two candidate events observed in single detector periods: 25 December 2015 04:11:44 UTC with the LIGO Hanford detector and 4 January 2016 12:24:17 UTC with the LIGO Livingston detector. The first candidate event has a low significance with a probability of astrophysical origin [37] $p_{\mathrm{astro}} = 0.12$, while the second has a larger significance $p_{\mathrm{astro}} = 0.47$. However, for this event, an excess power observed in the residual after subtraction of the best-fit waveform from the data suggests this event may not be of astrophysical origin, and is thus discarded.

Glitches of different types 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 populations. Their complex and time-evolving nature makes glitch identification and rejection a good problem and a 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 to 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. [38–41] for recent reviews).

In the context of GW signal searches, convolutional neural networks (CNNs) [42] have been investigated to detect BBH signals for both single- and multi-detector cases [43–48]. The primary motivation put forward in those contributions is the computational gains expected from the use of CNNs compared to matched filtering techniques.

So far a large fraction of those investigations use simulated Gaussian noise [43, 45, 46, 48]. 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 [44, 47]. The classifiers 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 min, which is not sufficient in practice. A recent review [49] 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 artefacts and improve their sensitivity, with a particular focus on analysing data from a single detector. The goal is to achieve a false alarm rate similar to that of current online searches performed by the LIGO-Virgo-KAGRA collaboration (LVK). The current convention defines two selectivity levels [50]: two false alarms per day for marginally significant or 'subthreshold' events and one false alarm per month for public alerts. We explore various network architectures, particularly those designed for time series classification [51, 52].

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 signals were detected using the matched filtering based searches.

Section 2 provides details on how the training and testing sets are generated, while section 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 section 4. We applied these classifiers to the remaining three months of O1 data, including segments associated with the three GW events detected during O1. Section 5 summarises 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. Interestingly, we found that only one data segment was classified as 'signal' by all three classifiers we considered. This event coincides with the single-detector event found by [36] in the LIGO Livingston data, as mentioned above, and was downgraded by the same study as a noise artefact. Following the additional checks we conducted on this event, we arrived at a different conclusion as they confirmed its compatibility with an astrophysical origin. Finally, section 6 concludes on the applicability of the proposed methodology.

Note that the choice of using only O1 data collected by the LIGO Livingston detector is completely arbitrary and does not impact the method described. We expect to probe the data collected in O1 by the Hanford detector and all the other observation runs in future works.

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. [43–46, 48]. 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 labelled as follows:

- noise: the data are compatible with stationary background noise, i.e. are free of transient instrumental artefacts (glitches) or known GW events,
- glitch: the data include one or several transient instrumental artefacts (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 GW Open Science Center (GWOSC) [53]. Specifically, we utilise the data from the LIGO Livingston (L1) detector spanning one month between 25 November 2015 (GPS time 1132 444 817) and 25 December 2015 (GPS time 1135 036 817). Throughout this duration, no GW signals were detected by the standard search pipelines.

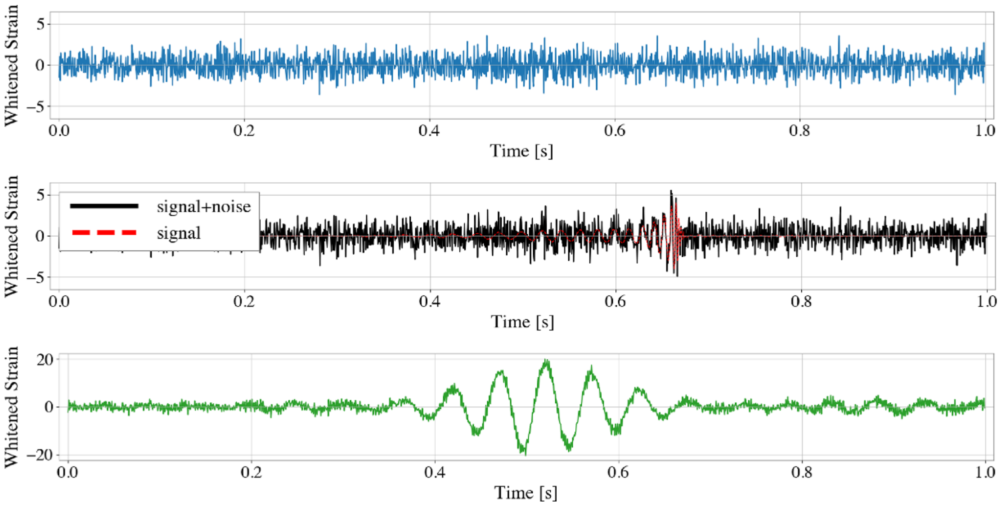

Figure 1. Instances of the classes noise (blue), signal (black) and glitch (green). Top (noise): one-second data segment recorded by LIGO Livingston at the GPS time 1132 550 972.487. Middle (signal): a simulated BBH waveform with SNR of 20 (dashed red line) is injected in the previous timeseries. Bottom (glitch): data recorded at the GPS time 1132 580 628.41 which contains a low-frequency transient instrumental artefact.

In this period the available L1 data amounts in total to about 13.3 days (1147 457 s), of which 3.6 days (312 284 s) were 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,[8] bandpass-filtered between 20 Hz and 1 kHz and whitened by applying the inverse amplitude spectral density (ASD) in the frequency domain[9]. 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 100 573 s). The data are divided into one-second non-overlapping segments.

The data are distributed into the three classes introduced above as explained in the next sections. Representative instances of the three classes are shown in figure 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[10]. All segments in the dataset passed the first criterion, as no GW signals were confidently detected by standard pipeline over the selected period[11]. Overall, a total of 1000 000 samples are obtained in this way from the one-month O1 dataset. Of these, 250 000 are only used to build the signal class for training as discussed in section 2.3, other

[8] The method `signal.decimate` of the software package `Scipy` [54] is used to downsample.

[9] For the preparation of the training and testing data, we acknowledge the use of the following software packages: `GWpy` [55], `PyCBC` [30] and `LALSuite` [56].

[10] 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 www.gw-openscience.org/o1_inj.

[11] This implies that the noise label is essentially determined by the sensitivity limit of the matched-filtering based searches.

250 000 make up the noise samples for training and 500 000 are used both to generate the noise and signal samples for testing.

2.2. The glitch class

A database of glitches is created using two different sources: the unmodelled transient search *coherent WaveBurst* (cWB) [57, 58] and the citizen science project Gravity Spy [59].

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 [57] that involves applying non-physical time shifts to the data before 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 an SNR higher than 5.8 were selected (258 480 glitches). This list was complemented with 8244 glitches from the Gravity Spy database. These glitches are identified by the Omicron trigger pipeline [60] with an SNR above 7.5 and can be downloaded from zenodo [61] selecting the GPS time between 1132 444 817 and 1135 036 817, i.e. in the one month we are considering in this step of the analysis. 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 s, the associated segment is labelled as a glitch. Note that the glitch has a random position within the one-second window. If the glitch duration is longer than 1 s, all segments that overlap with that glitch duration are labelled as glitch. Only the glitches whose time belongs to the data segments available on GWOSC (i.e. when the detector is on observing mode) 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.

The number of glitches over a given period is determined by the occurrence frequency of those noise artefacts, typically tenths of seconds to tenths of minutes depending on the period, and on the typical glitch amplitude. As glitches occur sporadically, this represents a small fraction of the total observation time, thus resulting in a smaller number of samples in the glitch class. 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 utilised 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 [62] 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 BBH 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 $m_1 > m_2 \geqslant 10M_\odot$ and the total mass $M = m_1 + m_2$ is uniformly distributed in $33M_\odot \leqslant M \leqslant 60M_\odot$. We consider non-spinning BH, so the dimensionless spin magnitudes χ_1 and χ_2 are set to 0. The phase at coalescence and the polarisation 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 [63], it is defined as

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

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 merger part of the signal is completely contained in the segment. More precisely, the whole signal is shifted in such a way that the merger time is in the interval [0.25 s, 0.8 s]. 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 characterising the classifier's performance. In total, the training and testing datasets comprise 1650 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 utilised 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 [43–48], the classifier is directly fed by the one-second segment of strain time series, so a vector size of 2048. We experiment[12] with three different network architectures, namely the CNN, as well as two other architectures specialised for time-series classification: temporal convolutional network (TCN) [51] and inception time (IT) [52]. The last two, to our knowledge, have never been tested with this type of problem, even if some ideas on which they are based have been used in GW astronomy for other kind of analyses, for example in the context of continuous GWs searches in [66] or for binary neutron star searches in [67].

The architectures are described in more detail in the following subsections. The model hyperparameters provided below have been tuned after a coarse exploration of the parameter space.

3.1. CNN

CNNs were first introduced for image classification [42]. They are now used for a wide variety of tasks, including the detection of GW signals [43–48]. In this study, we tested a range of CNNs similar to those considered in previous works.

[12] Implementations are based on the `TensorFlow` library [64] with the `Keras` API [65].

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 [42].

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	—

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 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.

3.2. TCN

TCN [51, 68] is a neural network architecture specifically developed for sequence modelling 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 where the filter is ‘dilated’ by inserting gaps between the filter coefficients. The size of the gaps is fixed to $d-1$ where d is referred to as the *dilation factor*.

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 six 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, which 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 $d_{\rm tot} = \sum d_k$ [51]. 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 the input sequence [51]. To satisfy this constraint, 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. IT

IT [52] is a deep network ensemble designed specifically for time series classification. It leverages the concept of residual networks and incorporates Inception modules [69]. 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 [52] 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 optimised using the training set described in section 2 to minimise the categorical cross-entropy loss function. The default implementations of the Adam optimiser are utilised, with a batch size of 24 [64]. The training procedure is repeated 10 times with different (random) initialisations of the model weights and dropouts, and the instance exhibiting the best receiver operating characteristic (ROC) curve on the testing dataset (as explained in section 4) is chosen. Note that this evaluation cannot be done with the validation dataset, as it does not provide enough statistics to compute the ROC in the relevant regime of low false alarm rates.

Throughout the training process, the model's area under the ROC curve [70] 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[13]. On the Tesla K40d GPU we used, the training times per epoch were 220 s for CNN, 1000 s for TCN, and 3320 s 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[14]. We aim to constrain false alarms to a rate of two per day (similar to the current online 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.

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

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

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

which concludes the classifiers. Because of the high-confidence level required, this threshold is very close to 1, thus requiring attention to the numerical precision for the evaluation of the membership probability. (This issue related to the precision of floating-point arithmetics was already noted in [48]). 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. The impact of this numerical issue is shown later in section 4.3.

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. The prevalent factor is the absence of a distinct boundary between the two classes. To produce a large sample of glitches, the criterion used to identify the training samples of the glitch class had to be relaxed, thus resulting in the selection of glitch instances with low amplitude, that confine with noise segments where the background noise present some level of non-stationarity. The relative class imbalance of the glitch class being under-represented by a factor of 3.5 compared to the noise class in the training set is also likely to play a role.

To support this interpretation we trained and tested the classifiers with much smaller datasets in which the three classes had the same number of samples. In those trials, the glitch samples only included distinct noise artefacts of large amplitude. The resulting classifiers did not have the confusion issue as we observe here.

The initial assumption that a three-class division would enhance classification performance thus turned out to be incorrect, at least with this dataset. 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. Figure 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 P_s higher values, the distributions show a monotonic decay until the last bin $P_\mathrm{s} = 1$ where distributions exhibit an count increase which is more prominent for TCN[15]. The TCN classifier appears to reach the lowest background $\lesssim 10^{-2}$ in normalised count units.

Since our objective is to achieve high-confidence classification, we are primarily interested in the immediate vicinity of $P_\mathrm{s} = 1$. This motivates us to reparameterise the P_s statistic as

[15] We relate this increase in the background count to an issue with the floating-point precision.

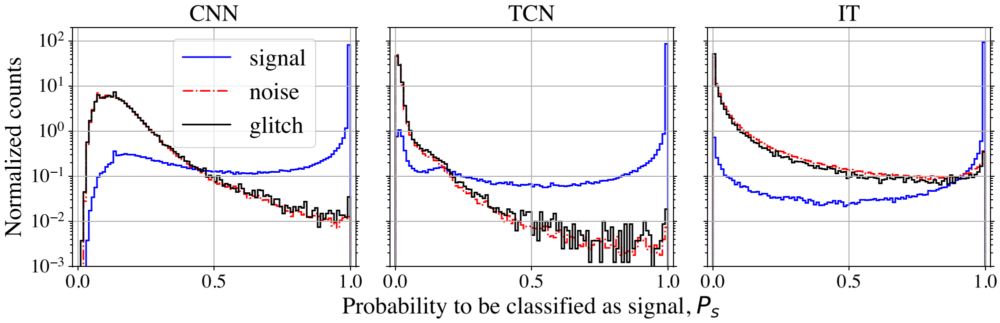

Figure 2. Distributions of P_s (the class membership probability assigned to the signal class), conditioned on the class of the input segment from the testing set: signal (blue), noise (dot-dashed red), or glitch (black). These distributions were computed for the CNN (left), TCN (middle), and IT (right) architectures. The histograms are normalised 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 section 4 for a discussion on this point).

$\lambda := -\log_{10}(1-P_\mathrm{s})$. While P_s ranges from 0 to 1, λ can theoretically take values across the entire real line. Our main focus lies in the range of large λ values, that is $\lambda \gtrsim 7$ because the computations are performed in single precision[16]. The most stringent criterion is to require $P_\mathrm{s} = 1$ 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. Figure 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 $P_\mathrm{s} = 1$. However, the peak appears narrower for the IT classifier. To focus on the region of interest near $P_\mathrm{s} = 1$, we employ the λ reparametrisation, as depicted in figure 3. This figure also incorporates the dependencies on the 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 $P_\mathrm{s} = 1$, 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. $P_\mathrm{s} = 1$).[17] 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,

[16] With single-precision floats, the closest P_s can get to 1, without being 1, is $P_\mathrm{s} = 1 - 2^{-24}$. For this value of P_s we have $\lambda = -\log_{10}(1-P_\mathrm{s}) \approx 7.22$.

[17] This comes from the use of single-precision floating point numbers (see discussion above).

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. In general, the histograms with $P_s = 1$ tend to be more populated at large SNR. For the samples classified with $P_s < 1$, the correlation between the SNR and λ is visible for IT and less clear for the other classifiers, for which the chirp mass seems to play a more important rôle.

4.3. Global assessment with ROC

To fully characterise the performance of the classifier, the noise rejection and signal extraction capabilities have to be evaluated jointly. This can be done by computing the ROC curves [70]. The classification efficiency $S_{\mathrm{th}}/S_{\mathrm{tot}}$ and false alarm rate $N_{\mathrm{th}}/N_{\mathrm{tot}}$ 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 category. Note that since each sample has a duration of 1 s, N_{tot} is intended here as the total duration in seconds of noise and glitch samples, so $N_{\mathrm{th}}/N_{\mathrm{tot}}$ is measured in s^{-1}. By varying the threshold, one obtains the ROC curves in figure 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. Figure 4 also shows the ROC computed for two instances of the IT architecture, when the categorical cross-entropy loss is calculated from the logits tensor (green) and when it is calculated from the class membership probability after the softmax transformation (red), see section 3.5 for an explanation and discussion. The shaded area represents the range between the best and worst models among the ten instances computed at training. When softmax is used, the uncertainty in the performance is larger (the shaded area is wider) and the classification accuracy reached at small false alarm rates is lower.

Figure 5 shows the classification efficiency for a given false alarm rate set to $10^{-5}\ \mathrm{s}^{-1}$ (that is 2 per day), as a function of the injected SNR as defined in equation (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 does not show the differences evidenced in figure 3. Overall, signals with SNR = 10 can be detected at the considered significance level with a good probability, larger than 50%.

Another standard metric to assess the performance of the classifier is the sensitive distance, see e.g. [71]. This metric can be computed by running the classifiers over a new test set for the signal class generated using the same configuration as above, but with sources distributed uniformly in volume instead of uniformly in SNR[18]. We obtain a sensitive distance of about 500 Mpc for the IT, TCN and CNN classifiers for a false alarm rate of $10^{-5}\ \mathrm{s}^{-1}$. The sensitive distances measured by [49] (dataset 4, using O3a data from LIGO Livingston and Hanford) range from 700 to 1700 Mpc approximately with the same requirement on the false alarm rate. In this range, we have ignored the method 'TPI FSU Jena' [48] whose sensitive distance drops close to 0 Mpc when applied to real data. This comparison is only indicative as the search sensitivities evaluated in [49] are for a two-detector network.

[18] The parameters other than the distance are distributed as described in section 2.3. The classifiers previously trained with the sources distributed uniformly in SNR are used for this evaluation.

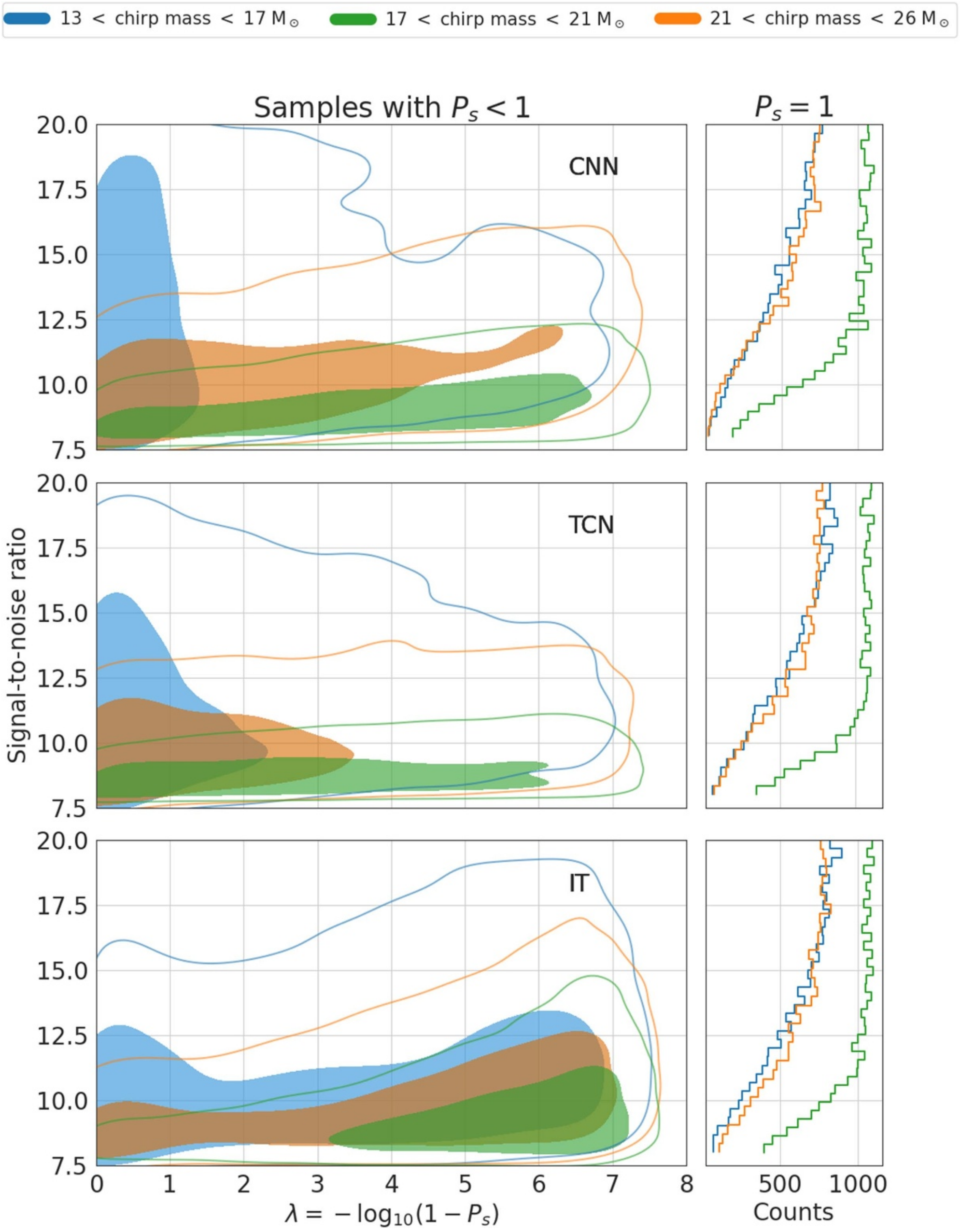

Figure 3. Distribution of the statistic $\lambda := -\log_{10}(1 - P_\mathrm{s})$ obtained with testing samples from the signal class and computed for the three considered classifiers: CNN (top), TCN (centre) and IT (bottom). The column on the left shows a kernel density estimate of the λ distribution for the samples with $P_\mathrm{s} < 1$, 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 a histogram for the samples with $P_\mathrm{s} = 1$. 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).

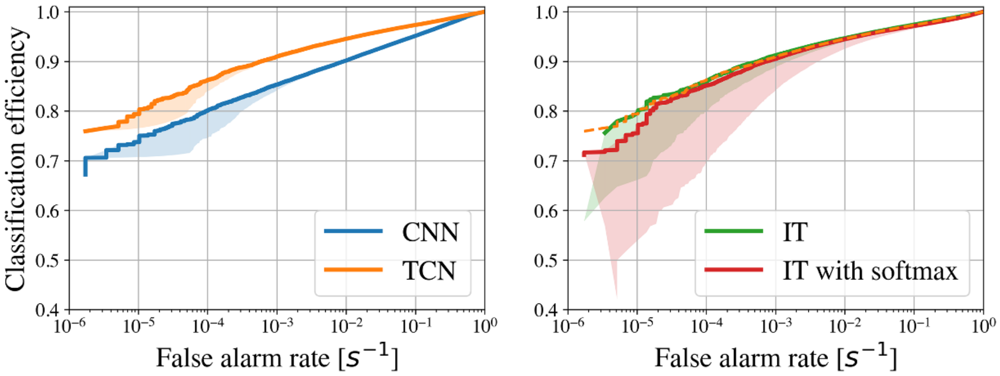

Figure 4. ROC curves for the three considered classifiers, CNN, IT, and TCN, illustrating the classification efficiency versus the false positive rate. Each classifier has been trained ten times, and the continuous line represents the result obtained for the best model, while the shaded area covers the 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 section 3.5). The TCN ROC curve is reproduced in this panel as a dashed orange line to facilitate comparison.

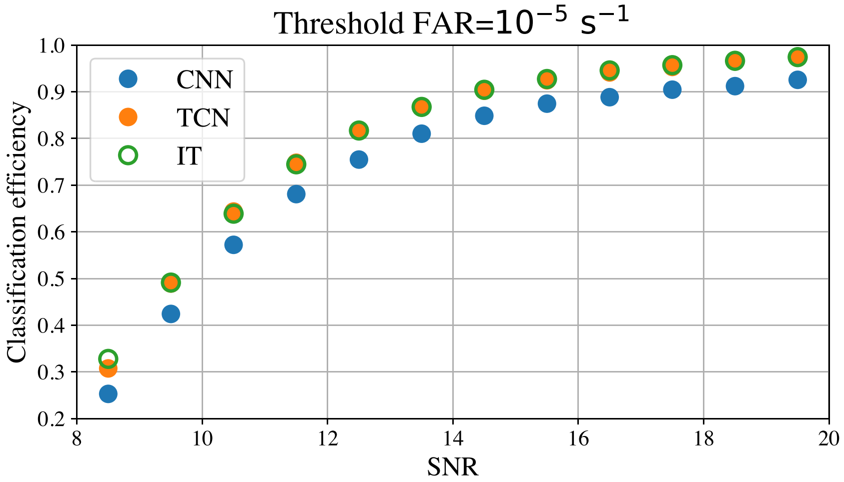

Figure 5. Classification efficiency versus SNR for a false alarm rate of 10^{-5} s^{-1}. The classifiers TCN and IT give similar efficiencies and surpass uniformly over CNN. Overall, signals with SNR = 10 can be detected at the considered significance level with a good probability, larger than 50%.

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 results for the other models can be found in appendix.

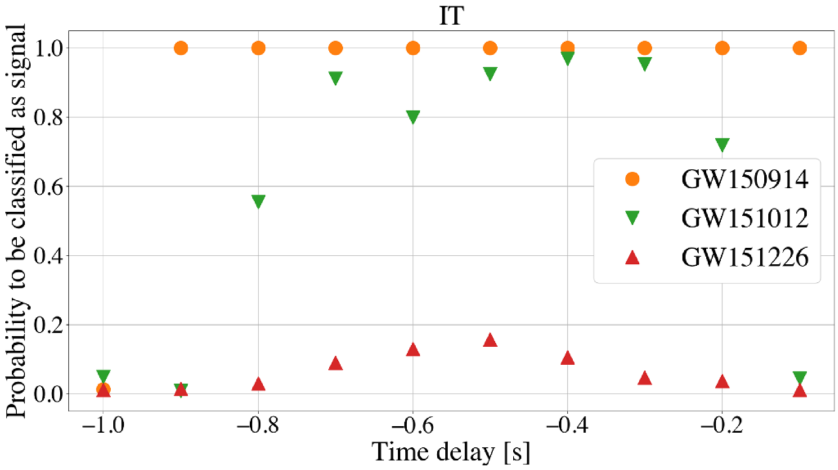

Figure 6. Evolution of the statistic P_s produced with IT classifier versus the relative delay Δt of the analysis window to the O1 event merger time (GW150914, GW151012 and GW151226). For $\Delta t = -1$ s, the analysis window only includes the initial part of the signal (inspiral), whereas, for $\Delta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

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 the one-second window, that is for different time delays Δt between the start of the 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). Figure 6 shows the evaluation of P_s between those two extreme cases for the IT model for GW150914, GW151012 and GW151226 (see also appendix).

As expected, when the merger 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 with $P_s = 1$, regardless of its position in the time window, even if it is partially visible. GW151012 is only detected with $P_s \sim 0.9$ when the merger is at the centre of the analysis window. GW151226 is not detected (i.e. P_s is always below 0.2). 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. Both GW151012 and GW151226 have single detector optimal SNRs for Livingston from parameter-estimation analyses lower than the minimum value of 8 we used to train the network (namely, $5.8^{+1.2}_{-1.2}$ for GW151012 and $6.9^{+1.2}_{-1.1}$ for GW151226 according to table V of [8]).

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 section 2). This corresponds to the period between GPS = 1126 051 217 (9 December 2015 00:00:00 UTC) and GPS = 1132 444 817 (25 November 2015 00:00:00 UTC) and between GPS = 1135 036 817 (25 December 2015 00:00:00 UTC) and GPS = 1137 254 417 (19 January 2016 16:00:00 UTC). In this period we excluded the

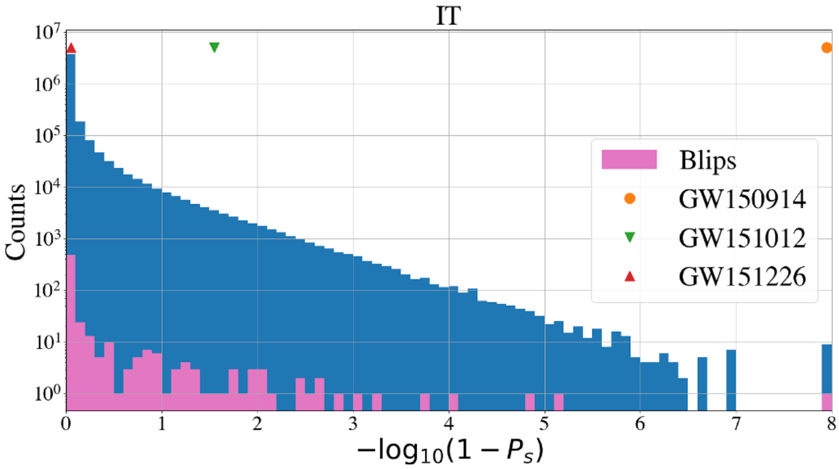

Figure 7. Distribution of the $\lambda = -\log_{10}(1-P_\mathrm{s})$ statistic (shown in blue) obtained using the IT classifier on the remaining O1 dataset (refer to section 5.2 for details). The segments with $P_\mathrm{s} = 1$ have been assigned a value of $\lambda = 8$ for plotting purposes. The pink histogram corresponds to a subset labelled as 'Blip' glitches by `Gravity Spy` [72]. The markers at the top indicate the highest values for the three O1 events displayed in figure 6. Please note that the vertical position of these markers is arbitrary.

intervals of ±1 s around the chirp time of the three known events (see previous section). This amounts to a total of 4216 489 s (about 49 days), of which 1054 564 s (about 12 days) are single-detector times, corresponding to 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. The processing time for the full data set is about 4 h per model on NVIDIA Tesla V100S GPUs, but most of this time is taken to load the data, the extraction of the model predictions takes about 8 min for CNN, 18 min for TCN and 52 min for IT. No data quality information was used, so this analysis is solely based on the GW strain data.

Figure 7 shows the distribution of the $\lambda = -\log_{10}(1-P_\mathrm{s})$ statistic obtained with the IT classifier (similar plots can be found in appendix for the other models). To find GW signals in these data we apply the most restrictive selection cut, by requiring $P_\mathrm{s} = 1$ (at machine precision). In other words, detected events correspond to segments classified with $P_\mathrm{s} = 1$. Given the result of the previous section, this implies that, with this selection cut, GW150914 would be the only detected event among the three known detections during O1. We recall that this selection cut corresponds to a false-alarm rate of $\lesssim 4\times10^{-6}\,\mathrm{s}^{-1}$ (that is one false alarm per 3 days) and a classification efficiency of 76% when estimated on the testing set, see sections 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 = 1131 289 775 (11 November 2015 15:09:18 UTC) and GPS = 1135 945 474 (4 January 2016 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 results 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 20 October 2015. 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 three classifiers: GPS = 1135 945 474 (4 January 2016 12:24:17 UTC) which we investigate further in the next section. As single-detector searches cannot employ statistical resampling techniques with time shifts [73], we can only provide an upper limit on the false alarm rate for this detection. The upper limit is estimated to be 1 event every 49 days, based on the available data from the three-month analysis period. This segment on 4 January 2016 corresponds to the event identified in the Livingston detector data during the O1 single-detector periods using a standard matched-filtering-based search, as reported in [36]. However, this candidate is subsequently eliminated by the authors of [36] after examining the residual obtained by subtracting the best-fit waveform from the data, since excess power is observed in the residual at frequencies below 80 Hz.

5.3. Detailed analysis of the 4 January 2016 event

We have performed a number of detailed checks of the 4 January 2016 event. We have performed a 'visual' inspection with the time-frequency constant-Q transform [55, 74]. Figure 8 provides a time–frequency representation of the entire segment. A transient is visible ~0.37 s after the start of the segment, at a frequency of about 150 Hz. In the magnified view, the shape of the transient is clearly indicative of a frequency modulated chirp-like transient.

The `Gravity Spy` database [72] has marked this specific GPS time classified as being an instrumental artefact of the 'Blip' type. The term refers to a well identified family of instrument glitches whose origin is still largely unknown (see, e.g. [14, 75] for more details). Generally, 'Blip' glitches do not exhibit a chirping frequency (see figure 1 of [76] for a typical example). To complement this initial inspection, figure 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 analysed. The resulting distribution is compatible with the overall background distribution. The 4 January segment appears to be an outlier with respect to 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` [77] and used the `IMRPhenomXPHM` waveform model [78]. 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 $H_0 = 67.9\,\mathrm{km\,s^{-1}\,Mpc^{-1}}$ [79]. The analysis did not include a marginalisation 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 = $1135\,945\,474.373^{+0.076}_{-0.07}$ and the measured optimal SNR is $11.34^{+1.8}_{-1.6}$.

Figure 9 shows the result of the fit in the time domain, by comparing the whitened data in orange to the reconstructed waveform corresponding to the maximum likelihood fit (green), and from the posterior mean (blue), shown with the 90% credible belt. We do not visually identify any notable residual after subtraction of the best fitting waveform as shown in figure 10. As an independent check of the nature of the signal, figure 9 also includes the waveform estimate produced by the denoising convolutional autoencoder described in [80] (dashed red). The reconstructed waveforms are in good agreement, following a similar phase evolution,

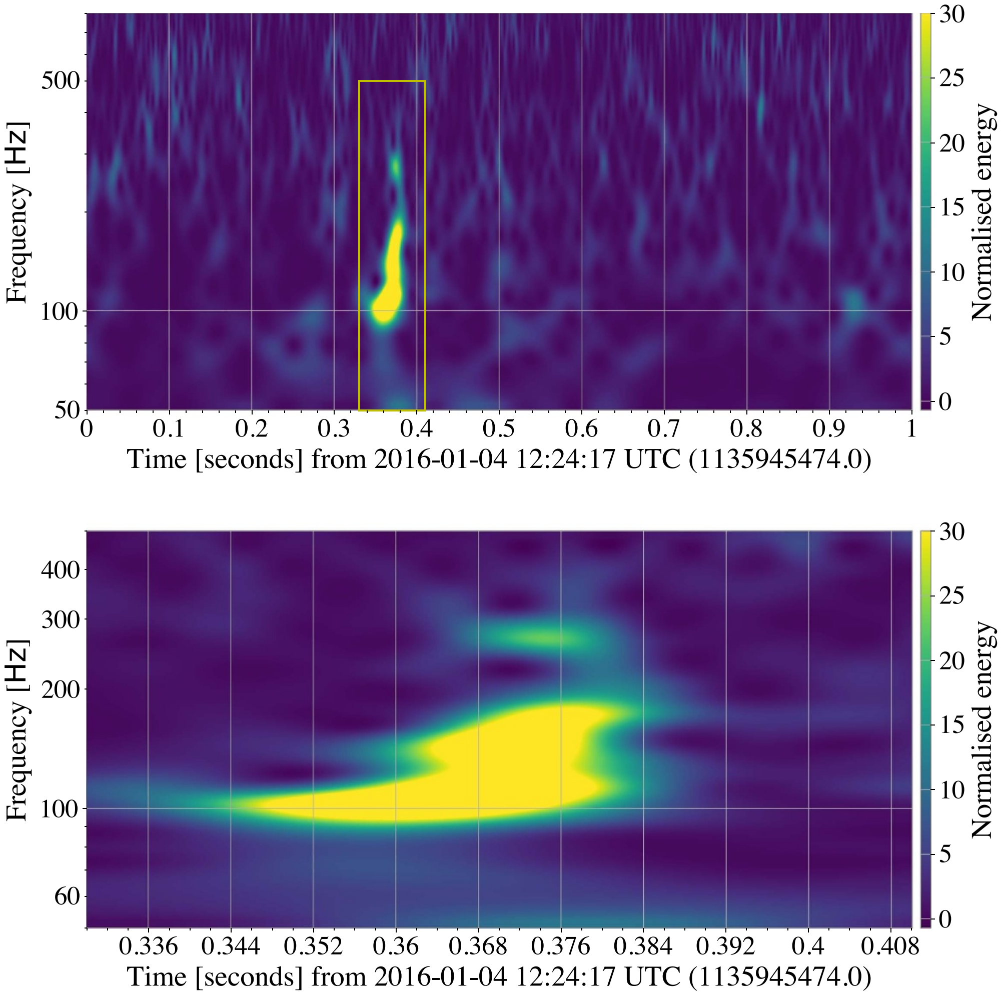

Figure 8. Time–frequency representation of the segment at 4 January 2016 12:24:17 UTC (GPS = 1135 945 474 (s) recorded by the LIGO Livingston detector. The top panel shows the entire segment over a frequency range between 50 and 800 Hz. The bottom panel is a detailed view that focuses on the transient signal at $t \sim 0.37$ s with a zoom in the frequency range between 50 and 500 Hz. This representation is obtained through a constant-Q transform [55] with quality factor $Q = 12$. To facilitate comparison, the dynamic range is fixed, following a similar approach as described in [36], and the colormap is saturated at a maximum value of 30 for the normalised energy.

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 (see [80]), and lower 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.

Following [80] we calculate the SNR of the denoised waveform using equation (1) and obtain a value of 9.7. Bacon *et al* [80] investigates the ability of the denoising algorithm to denoise data in the presence of glitches depending on the denoised SNR, see figure 5 of [80]. A value of 9.7 is shown to be sufficiently high to hint an astrophysical origin of the signal.

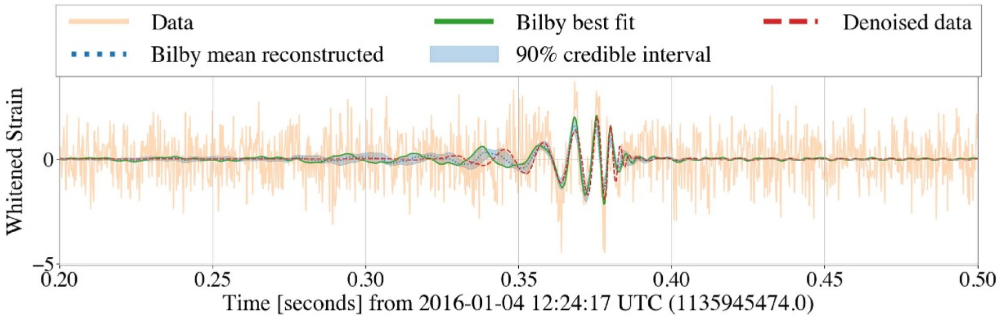

Figure 9. Comparison of the whitened L1 data (orange line) with the reconstructed waveform obtained from the posterior mean (dotted blue line), and from the maximum likelihood fit (solid green line) both computed using `Bilby`, and the 90% credible interval (blue) along with the ML denoising convolutional autoencoder neural network described in [80] (dashed red line).

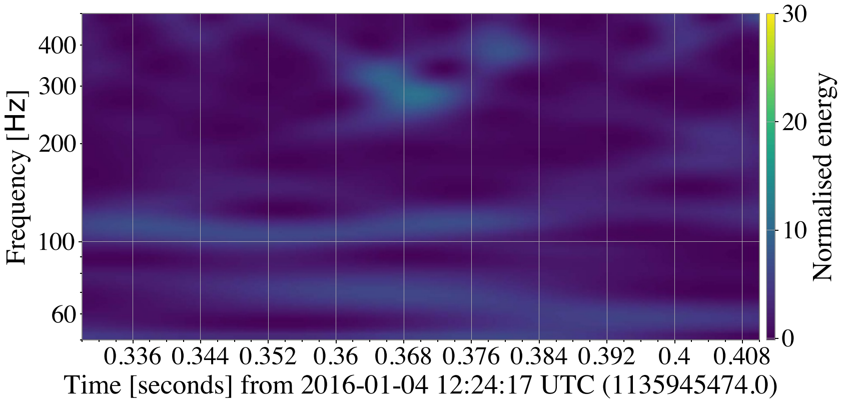

Figure 10. Time–frequency representation of the residual after the subtraction of the maximum likelihood fit waveform obtained with `Bilby` (green line in figure 9) from the data segment at 4 January 2016 12:24:17 UTC (GPS = 1135 945 474 (s). This representation adheres to the same settings as in figure 8, utilising a frequency range between 50 and 500 Hz, a constant-Q transform with a quality factor $Q = 12$ and the dynamic range is capped at a maximum of 30 for normalised energy, aligning with [36] to facilitate comparison. No excess power is visible in this plot.

This candidate signal was also identified by [36] using standard matched filtering techniques, but the event was subsequently downgraded to a noise artefact due to an excess observed in the residual. Figures 8 and 10 are produced with similar dynamic scale, colour range and time-frequency resolution as in [36]. Therefore, the mismatch either comes from a difference in the parameters estimated for the best fitting waveform (unfortunately, not included in [36]), from the difference in the waveform model (IMRPhenomPv2 was used by [36]) or both.

Above checks are all compatible with the event being of astrophysical origin. The corner plot in figure 11 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 localised in the sky. The 90% credible intervals for those parameters are:

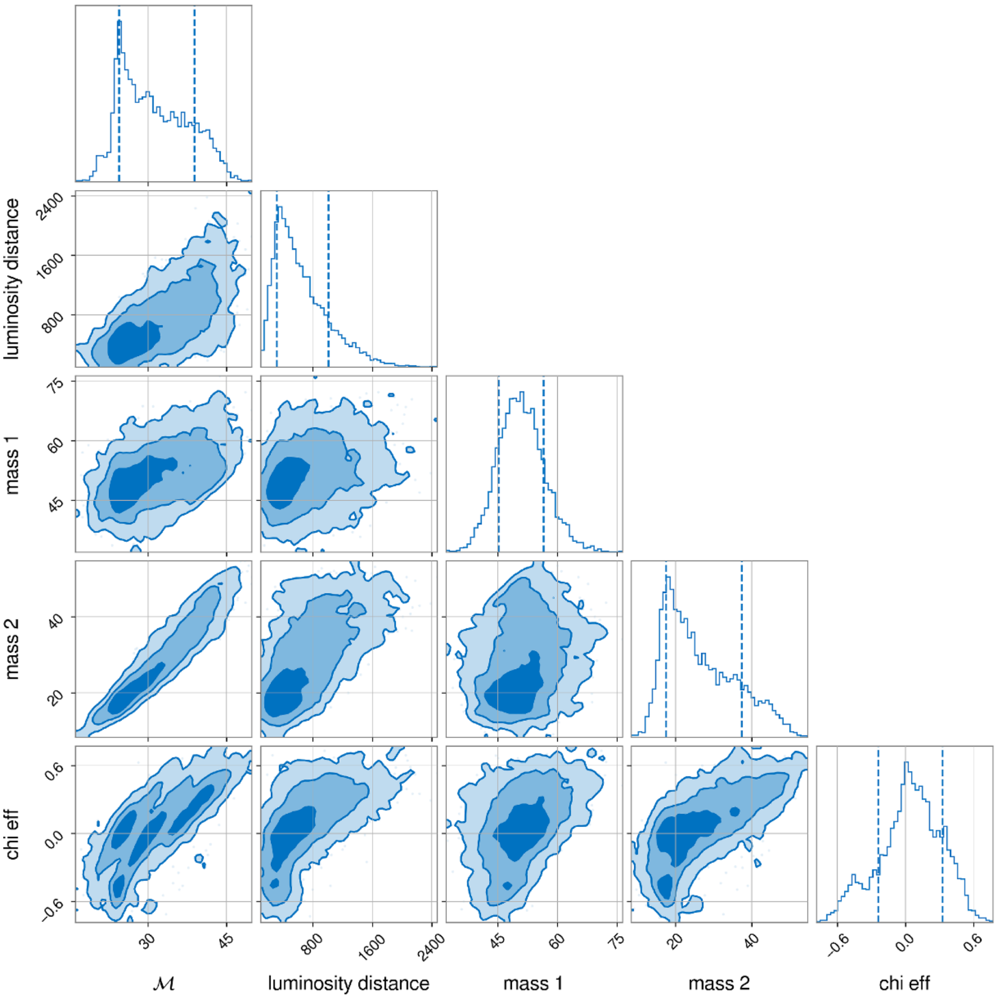

Figure 11. Posterior distribution of the chirp mass $\mathcal{M}$, luminosity distance, component masses m_1 and m_2, and effective spin χ_{eff} for the 4 January 2016 event (see section 5.3 for details).

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 $\chi_{\mathrm{eff}} = 0.06^{+0.4}_{-0.5}$ and the luminosity distance $d_{\mathrm{L}} = 564^{+812}_{-338}$ Mpc; see [81] 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 detectors' data for analysing single-detector observing periods of ground-based GW detectors. We show that architectures specifically designed for time-series classification, such as IT or TCN, outperform the standard CNN typically used so far. The SNR required to reach 90% classification efficiency with IT and TCN is lowered by 15% compared to CNN. The models were trained with one month of the observing run O1 data from the LIGO Livingston detector. When applied to the remaining three months of O1 data, the classifiers independently detect a plausible GW signal of astrophysical origin on 4 January 2016. The various diagnostics we performed substantiates the possibility of its astrophysical origin.

Operationally, we propose an approach where the multiple detector data from the first month of an observing run, labelled by standard matched filtering-based pipelines, are used to train the neural network models. The resulting classifiers can then be 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 localisation 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 characterisation of these algorithms and their noise rejection capabilities, as already highlighted in [49] 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. Those issues can be tackled in the future by resampling techniques for (i), and using double precision floating point numbers for (ii). More generally, due to the absence of a mathematical theory for neural networks, their precise statistical characterisation 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 analysing the fairly large single-detector data set. Applying the proposed classifiers to other LIGO-Virgo observing runs 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.

Data availability statement

The dataset used for training and testing have been published on zenodo at this DOI: https://doi.org/10.5281/zenodo.11093596 [82].

Acknowledgments

This work was partially supported by European Union’s Horizon 2020 research and innovation programme under Grant Agreement No. 653477, by diiP (data intelligence institute of Paris), by IdEx Université de Paris, by ANR-18-IDEX-0001, by the COST action G2net (CA 17137) and by ICSC—Centro Nazionale di Ricerca in High Performance Computing, Big Data and Quantum Computing, funded by European Union—NextGenerationEU. MB gratefully acknowledges partial support from the Polish National Science Centre Grant Nos. 2016/22/E/ST9/00037 and 2021/43/B/ST9/01714, and Poland’s high-performance Infrastructure PLGrid (ACK Cyfronet AGH). The authors are grateful for computational resources provided by the LIGO Laboratory and supported by National Science Foundation Grants PHY-0757058 and PHY-0823459. This work was granted access to the HPC resources of IDRIS under the allocation 2021-A0111012956 and 2021-AD011012279 made by GENCI. Some of the numerical computations were performed on the DANTE platform, APC, France.

The authors would like to thank Charlotte Pelletier for suggesting the investigation of the TCN and Inception Time classifiers. The authors also 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 advice with the usage of

Bilby, and Konstantin Leyde for stimulating discussions about this work. We thank Sophie Bini and Thomas Dent for their comments during the LVK internal review.

This research has made use of data or software obtained from the GWOSC (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 of the United Kingdom, the Max-Planck-Society, 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, by the French Centre National de Recherche Scientifique, the Italian Istituto Nazionale di Fisica Nucleare) 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, Japan Society for the Promotion of Science in Japan; National Research Foundation and Ministry of Science and ICT in Korea; Academia Sinica and National Science and Technology Council in Taiwan.

Appendix. 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 section 5.1. Comparing figures 6 and A2, we observe that GW150914 is the only event classified with $P_\mathrm{s} = 1$ 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 figure 7 in section 5.2. Figure A1 shows the same distribution for CNN and TCN. The distribution obtained with CNN decays faster than the other two models but exhibits a tail that reaches the extreme point, $P_\mathrm{s} = 1$. 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.

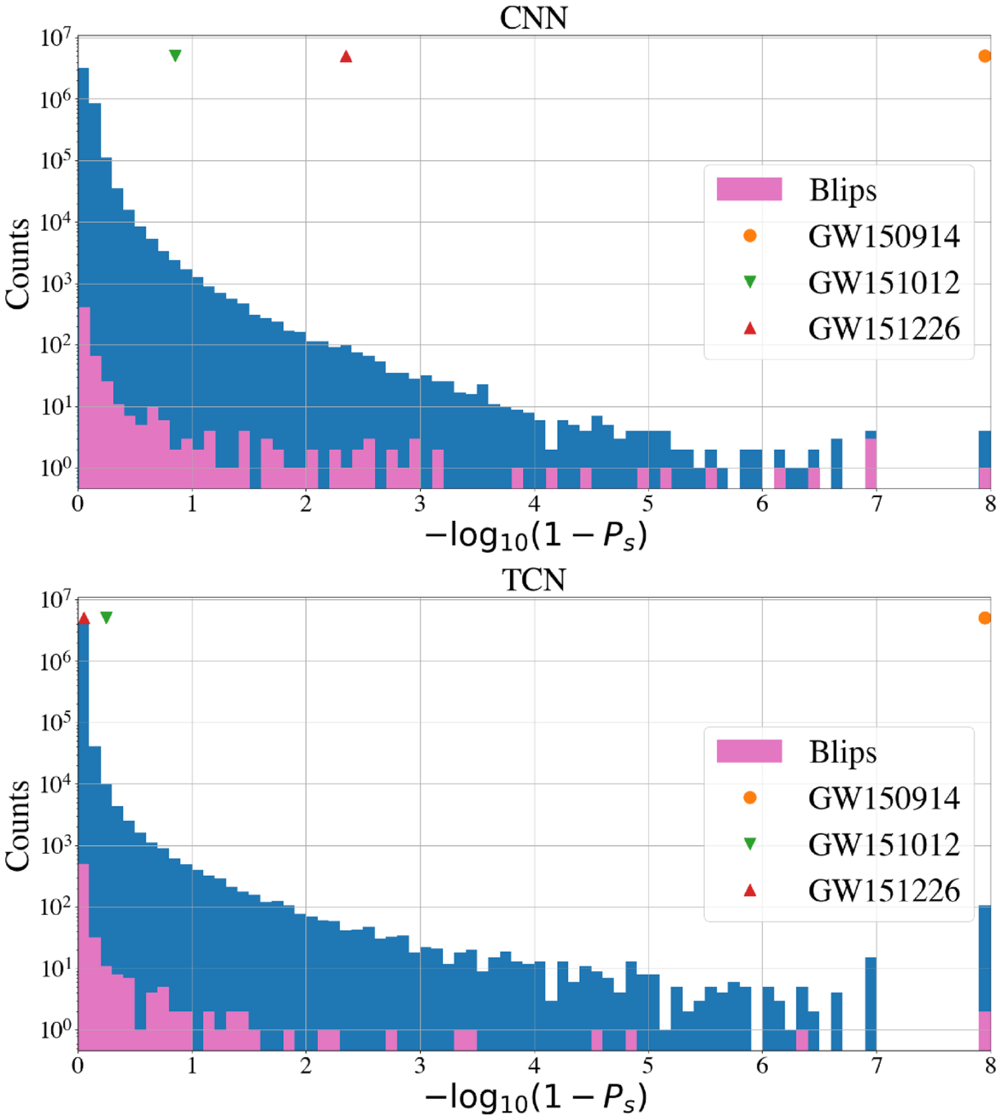

Figure A1. Distribution of the $\lambda = -\log_{10}(1-P_\mathrm{s})$ statistic (shown in blue) obtained using the CNN (top panel) and TCN (bottom panel) classifiers on the remaining O1 dataset (refer to section 5.2 for details). The segments with $P_\mathrm{s} = 1$ have been assigned a value of $\lambda = 8$ for plotting purposes. The pink histogram corresponds to a subset labelled as 'Blip' glitches by `Gravity Spy` [72]. The markers at the top indicate the highest values for the three O1 events displayed in figure 6. Please note that the vertical position of these markers is arbitrary.

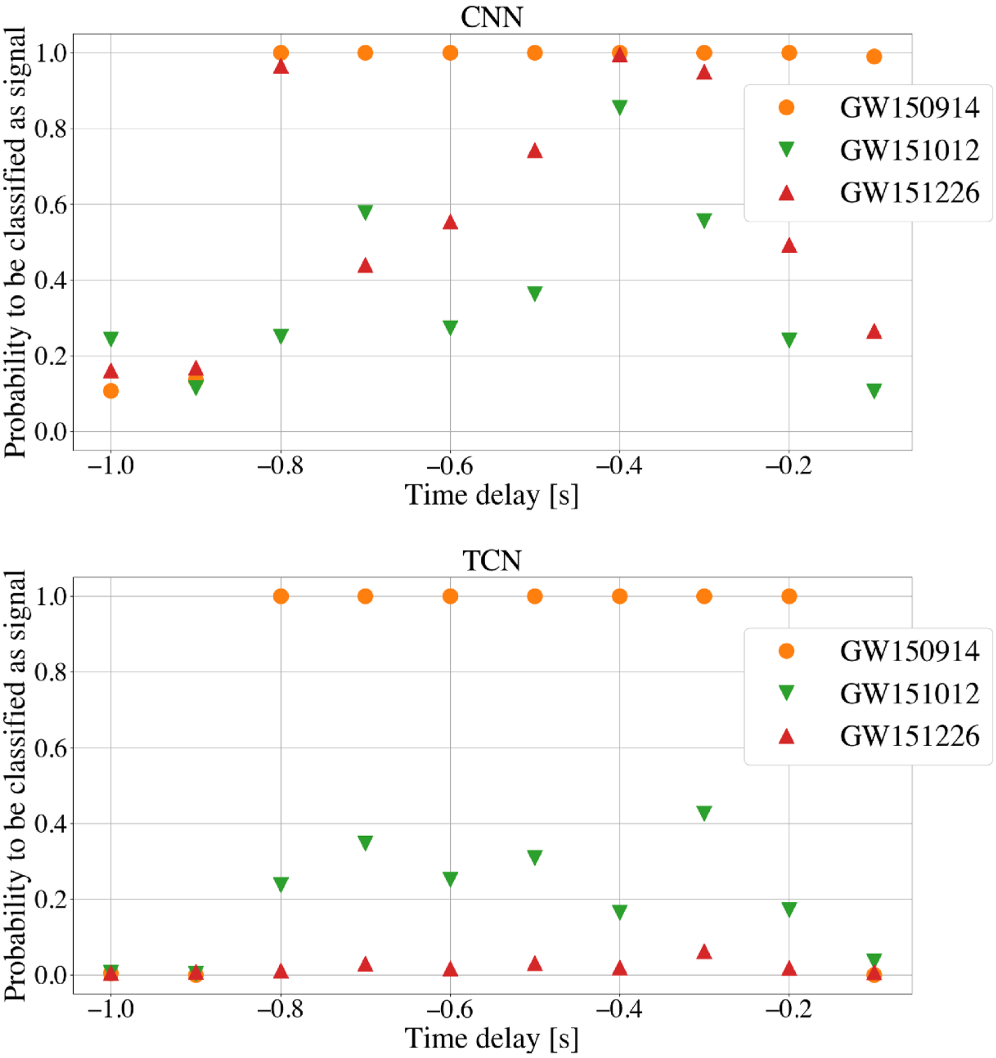

Figure A2. Top panel: evolution of the statistic P_s produced with CNN classifier versus the relative delay Δt of the analysis window to the O1 event merger time (GW150914, GW151012 and GW151226). For $\Delta t = -1$ s, the analysis window only includes the initial part of the signal (inspiral), whereas,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

ORCID iDs

A Trovato https://orcid.org/0000-0002-9714-1904
M Bejger https://orcid.org/0000-0002-4991-8213
R Flamary https://orcid.org/0000-0002-4212-6627

References

[1] Abbott B P *et al* (LIGO Scientific Collaboration and 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 and Virgo Collaboration) 2017 *Phys. Rev. Lett.* **119** 161101

[5] Abbott B P *et al* (LIGO Scientific Collaboration and Virgo Collaboration) 2020 *Astrophys. J.* **892** L3

[6] Abbott R *et al* (LIGO Scientific Collaboration and Virgo Collaboration) 2020 *Astrophys. J.* **896** L44

[7] Abbott R *et al* (LIGO Scientific Collaboration and Virgo Collaboration) 2021 *Astrophys. J. Lett.* **915** L5

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

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

[10] Abbott R *et al* (LIGO Scientific Collaboration and Virgo Collaboration) 2024 *Phys. Rev.* D **109** 022001

[11] Abbott R *et al* (LIGO Scientific Collaboration, Virgo Collaboration and KAGRA Collaboration) 2023 *Phys. Rev.* X **13** 041039

[12] Creighton J and Anderson W 2011 *Gravitational-Wave Physics and Astronomy: An Introduction to Theory, Experiment and Data Analysis* (Wiley) (https://doi.org/10.1002/9783527636037)

[13] Roy S, Sengupta A S and Ajith P 2019 *Phys. Rev.* D **99** 024048

[14] Davis D *et al* 2021 *Class. Quantum Grav.* **38** 135014

[15] Acernese F *et al* 2023 *Class. Quantum Grav.* **40** 185006

[16] Alvarez-Lopez S, Liyanage A, Ding J, Ng R and McIver J 2023 arXiv:2304.09977

[17] Choudhary S, More A, Suyamprakasam S and Bose S 2023 *Phys. Rev.* D **107** 024030

[18] Dhurandhar S, Mukhopadhyay H, Tagoshi H and Kanda N 2011 *Int. J. Mod. Phys.* D **20** 2051–6

[19] LIGO Scientific Collaboration and Virgo Collaboration 2015 *O1 Summary* (available at: https://gwosc.org/detector_status/O1/)

[20] LIGO Scientific Collaboration and Virgo Collaboration 2016 *O2 Summary* (available at: https://gwosc.org/detector_status/O2/)

[21] LIGO Scientific Collaboration and Virgo Collaboration 2019 *O3a Summary* (available at: https://gwosc.org/detector_status/O3a/)

[22] LIGO Scientific Collaboration and Virgo Collaboration 2019 *O3b Summary* (available at: https://gwosc.org/detector_status/O3b/)

[23] Messick C *et al* 2017 *Phys. Rev.* D **95** 042001

[24] Nitz A H, Dal Canton T, Davis D and Reyes S 2018 *Phys. Rev.* D **98** 024050

[25] Abbott B P *et al* (LIGO Scientific Collaboration and Virgo Collaboration) 2019 *Astrophys. J.* **875** 161

[26] Callister T A, Kanner J B, Massinger T J, Dhurandhar S and Weinstein A J 2017 *Class. Quantum Grav.* **34** 155007

[27] Sachdev S *et al* 2019 arXiv:1901.08580

[28] Tsukada L *et al* 2023 arXiv:2305.06286

[29] Ewing B *et al* 2023 arXiv:2305.05625

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

[31] Davies G S C and Harry I W 2022 *Class. Quantum Grav.* **39** 215012

[32] Abbott B P *et al* (LIGO Scientific Collaboration and Virgo Collaboration) 2019 *Astrophys. J.* **886** 75

[33] Nitz A H, Nielsen A B and Capano C D 2019 *Astrophys. J. Lett.* **876** L4

[34] Stachie C *et al* 2020 *Class. Quantum Grav.* **37** 175001

[35] Magee R *et al* 2019 *Astrophys. J. Lett.* **878** L17

[36] Nitz A H, Dent T, Davies G S and Harry I 2020 *Astrophys. J.* **897** 169

[37] Abbott B P *et al* (LIGO Scientific and Virgo) 2016 *Astrophys. J. Lett.* **833** L1

[38] Cuoco E *et al* 2020 *Mach. Learn.: Sci. Technol.* **2** 011002

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

[40] Soni S *et al* 2021 *Class. Quantum Grav.* **38** 195016

[41] Zevin M *et al* 2017 *Class. Quantum Grav.* **34** 064003

[42] Goodfellow I J, Bengio Y and Courville A 2016 *Deep Learning* (MIT Press) (available at: http://www.deeplearningbook.org)

[43] George D and Huerta E A 2018 *Phys. Rev.* D **97** 044039

[44] George D and Huerta E 2018 *Phys. Lett.* B **778** 64–70

[45] Gabbard H, Williams M, Hayes F and Messenger C 2018 *Phys. Rev. Lett.* **120** 141103

[46] Krastev P G 2020 *Phys. Lett.* B **803** 135330

[47] Gebhard T D, Kilbertus N, Harry I and Schölkopf B 2019 *Phys. Rev.* D **100** 063015

[48] Schäfer M B, Zelenka O, Nitz A H, Ohme F and Brügmann B 2022 *Phys. Rev.* D **105** 043002

[49] Schäfer M B *et al* 2023 *Phys. Rev.* D **107** 023021

[50] LIGO-Virgo-KAGRA Public Alerts User Guide (available at: https://emfollow.docs.ligo.org/userguide)

[51] Bai S, Zico Kolter J and Koltun V 2018 arXiv:1803.01271

[52] Ismail Fawaz H, Lucas B, Forestier G, Pelletier C, Schmidt D F, Weber J, Webb G I, Idoumghar L, Muller P-A and Petitjean F 2020 *Data Min. Knowl. Disc.* **34** 1936–62

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

[54] Virtanen P *et al* 2020 *Nat. Methods* **17** 261–72

[55] Macleod D M, Areeda J S, Coughlin S B, Massinger T J and Urban A L 2021 *SoftwareX* **13** 100657

[56] LIGO Scientific Collaboration 2018 LIGO algorithm library - LALSuite free software (GPL) (available at: https://doi.org/10.7935/GT1W-FZ16)

[57] Drago M *et al* 2021 *SoftwareX* **14** 100678

[58] Klimenko S *et al* 2021 cwb pipeline library: 6.4.0 (available at: https://doi.org/10.5281/zenodo.4419902)

[59] Bahaadini S, Noroozi V, Rohani N, Coughlin S, Zevin M, Smith J, Kalogera V and Katsaggelos A 2018 *Inf. Sci.* **444** 172–86

[60] Robinet F, Arnaud N, Leroy N, Lundgren A, Macleod D and McIver J 2020 *SoftwareX* **12** 100620

[61] Glanzer J *et al* 2021 Gravity spy machine learning classifications of LIGO glitches from observing runs O1, O2, O3a, and O3b (available at: https://zenodo.org/records/5649212)

[62] Bohé A *et al* 2017 *Phys. Rev.* D **95** 044028

[63] Maggiore M 2008 *Gravitational Waves* (*Theory and Experiments* vol 1) (Oxford University Press) (https://doi.org/10.1093/acprof:oso/9780198570745.001.0001)

[64] Abadi M *et al* 2015 TensorFlow: large-scale machine learning on heterogeneous systems software available from tensorflow.org (available at: www.tensorflow.org/)

[65] Chollet F *et al* 2015 Keras (available at: https://keras.io)

[66] Dreissigacker C and Prix R 2020 *Phys. Rev.* D **102** 022005

[67] Schäfer M B, Ohme F and Nitz A H 2020 *Phys. Rev.* D **102** 063015

[68] Remy P 2020 Temporal convolutional networks for keras (available at: https://github.com/philipperemy/keras-tcn)

[69] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V and Rabinovich A 2014 arXiv:1409.4842

[70] Fawcett T 2006 *Pattern Recognit. Lett.* **27** 861–74

[71] Capano C, Harry I, Privitera S and Buonanno A 2016 *Phys. Rev.* D **93** 124007

[72] Glanzer J *et al* 2021 Gravity spy machine learning classifications of LIGO glitches from observing runs O1, O2, O3a, and O3b (https://doi.org/10.5281/zenodo.5649212)

[73] Was M, Bizouard M-A, Brisson V, Cavalier F, Davier M, Hello P, Leroy N, Robinet F and Vavoulidis M 2010 *Class. Quantum Grav.* **27** 015005

[74] Chatterji S, Blackburn L, Martin G and Katsavounidis E 2004 *Class. Quantum Grav.* **21** S1809–18

[75] Cabero M *et al* 2019 *Class. Quantum Grav.* **36** 155010

[76] Glanzer J *et al* 2023 *Class. Quantum Grav.* **40** 065004

[77] Ashton G *et al* 2019 *Astrophys. J.* **241** 27

[78] Pratten G *et al* 2021 *Phys. Rev.* D **103** 104056

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

[80] Bacon P, Trovato A and Bejger M 2023 *Mach. Learn.: Sci. Technol.* **4** 035024

[81] Christensen N and Meyer R 2022 *Rev. Mod. Phys.* **94** 025001

[82] Trovato A *et al* Train and test datasets used for the paper ‘Neural network timeseries classifiers for gravitational-wave searches in single-detector periods’ https://doi.org/10.5281/zenodo.11093596