Detection of Einstein Telescope gravitational wave signals from binary black holes using deep learning

Wathela Alhassan,1* T. Bulik,1,2 M. Suchenek1
1 Particle Astrophysics Science and Technology Centre, Nicolaus Copernicus Astronomical Center, Rektorska 4, 00-614 Warsaw, Poland
2 Astronomical Observatory, University of Warsaw, Aleje Ujazdowskie 4, 00-478 Warsaw, Poland

Received YYY; in original form ZZZ

ABSTRACT
The expected volume of data from the third-generation gravitational waves (GWs) Einstein Telescope (ET) detector would make traditional GWs search methods such as match filtering impractical. This is due to the large template bank required and the difficulties in waveforms modelling. In contrast, machine learning (ML) algorithms have shown a promising alternative for GWs data analysis, where ML can be used in developing semi-automatic and automatic tools for the detection and parameter estimation of GWs sources. Compared to second generation detectors, ET will have a wider accessible frequency band but also a lower noise. The ET will have a detection rate for Binary Black Holes (BBHs) and Binary Neutron Stars (BNSs) of order 105 – 106 year−1 and 7 × 104 year−1 respectively. We explored the efficiency of using convolutional neural networks (CNNs) for the detection of BBHs’ mergers in synthetic noisy data that was generated according to ET’s parameters. Without performing data whitening or applying bandpass filtering, we trained four CNN networks with the state-of-the-art performance in computer vision, namely VGG, ResNet and DenseNet. ResNet has significantly better performance, and was able to detect BBHs sources with SNR of 8 or higher with 98.5% accuracy, and with 92.5%, 85%, 60% and 62% accuracy for sources with SNR range of 7-8, 6-7, 5-6 and 4-5 respectively. ResNet, in qualitative evaluation, was able to detect a BBH’s merger at 60 Gpc with 4.3 SNR. It was also shown that CNN can be used efficiently for near-real time detection of BBHs.

Key words: gravitational waves - transients: black hole mergers - methods: data analysis

1 INTRODUCTION
The recent detection of Gravitational Waves (GWs) has opened a new era of multimessenger astrophysics, where significant amount of data is being delivered from the second generation detectors of GWs such as GEO600 (Willke 2007), the advanced Laser Interferometer Gravitational-Wave Observatory (LIGO) (Abbott et al. 2009), and the Virgo interferometer (Acernese et al. 2007).

The detection of GW signals is traditionally done through the analysis of a hidden signal in the raw data from the detector, so-called matched filtering. Matched filtering process compares modelled waveform templates with the raw data. The main disadvantage of match filtering is that it depends entirely on the availability of waveform templates. This process is computationally challenging, and depends also on the accuracy of the modelled waveforms. Once a signal is detected, the subsequent step is to estimate its parameters, for which as many as 107 waveform templates are required for an accurate estimation (Schmidt et al. 2020).

With the expected volume of data from the third generation GWs detectors such as the Einstein Telescope (ET) (Punturo et al. 2010; Abernathy et al. 2011; Maggiore et al. 2020), Cosmic Explorer (CE) (Evans et al. 2021) and The Laser Interferometer Space Antenna (LISA) (Shaddock 2008), match filtering can be extremely computational expensive and the large template bank required would make it impractical. This motivates the necessity for the development of semi-automatic and automatic machine learning algorithms for detection, denoising and parameter estimation of GWs sources. Deep Learning (DL) is a promising alternative for GW data analysis, both for detection and parameter estimation. This goes back to its flexibility in categorizing and modelling patterns read from data, with very little human input. In contrast to match filtering search method, DL models can learn to distinguish between only noise and injected noise (noise plus signal) without the need to model every possible GW waveform. This is done through learning the underlying features of pure noise and injected noise, which gives DL an advantage over the traditional methods, where no matching is involved. DL algorithms can open up a new door for unmodelled waveforms discovery, specially with the expected number and type of sources that are expected to be detectable from the third generation detectors.

The ET will revolutionise the field of GWs in every sense, including and most importantly the better sensitivity of one order of magnitude that will provide, which will allow the detection at lower frequency. Compared to second generation detectors, ET will have a wider accessible frequency band that can be used to investigate a huge number of key issues related to astrophysics, fundamental physics and cosmology. The ET will have a detection rate for BBHs and BNSs of order 105 − 106 year−1 and 7 × 104 year−1 respectively (Kalogera et al. 2021). A recent study showed that, the BBHs would have lower detection rate at ET, which expected to be in the range 3.1 × 104 to 8.0 × 104 year−1 (Iacovelli et al. 2022). Our work here...
will focus on BBHs sources, and BNSs will be included in future work.

Over the last few years, different Machine Learning (ML) and DL techniques have been successfully applied to astronomical applications, on both images and spectral data (Lukic & Brüggen 2016). For example distinguishing between normal and peculiar optical galaxies by training Artificial Neural Networks (ANNs) (Naim et al. 1997)

Estimation of the photometric redshift of galaxies using ANNs (Collister & Lahav 2004), improved methods of the estimation of the photometric redshift of galaxies with more accurate results (Zhang et al. 2009; Bonfield et al. 2010) and most recent improvement by D’Isanto & Polsterer (2018), Selim et al. (2016) and Selim & Abd El Aziz (2017) where a method based on non-negative matrix factorization algorithm were used to classify the galaxy morphology. Similar method has also been applied to the galaxy-star classification (Kim & Brunner 2017), and Radio sources classification using CNNs Alhassan et al. (2018).

In GW, ML and DL have been wildly introduced into the field of GW data analysis in recent years. For second generation detectors ML and DL techniques have been successfully applied on both, time and frequency domain GWs data. For example, ML was used for glitches identification (Colgan et al. 2020), Biswas et al. (2013) used artificial neural networks, support vector machines, and random forests on LIGO data for the identification and removal of glitches. Another successful application used ML to discover the correlation between auxiliary monitoring channels and the main GWs output channels, for improving the sensitivity of the current interferometric GWs detectors (Vajente 2022). The potential of time-domain noise subtraction for improving detectors sensitivity below 30 Hz was also discussed in the paper. Baltus et al. (2021) used CNNs for pre-merger alert of three classes of BNSs of LIGO. This work was improved in (Baltus et al. 2022) to be able to search for all types of BNSs pre-mergers in three different types of noise: Gaussian O3 noise, real O3 noise, and predicted O4 noise. Moreno et al. (2022) used deep recurrent autoencoders (AEs) -with an unsupervised strategy- for GWs signals detection in laser interferometers. The results were better compared to other AEs architectures, but less accurate compared to the traditional supervised techniques. Álvares et al. (2021) Used CNNs for BBHs merger detection from simulated data of a single detector (Advanced LIGO detector) and multi-detectors combined. Their models were trained on sources with luminosity distance ($D_L$) ranging from 100 to 2000 Mpc, and masses between 5 to 100 $M_\odot$. Results from model trained on single detector data and multi-detector data achieved an accuracy of 72% and 82% respectively. Similar work was done by Jiang et al. (2022), where CNNs was used to search the LIGO O1, O2 and O3 data for BBHs sources that have been detected by both, Hanford and Livingston detectors, using time sliding method. Their model was able to detect the whole 41 BBHs sources exist in the data.

Detection of Extreme mass ratio inspirals (EMRIs) can be possible from third generation space-borne GW detectors such as LISA. A proof-of-principle approach for EMRI detection based on CNNs was presented in (Zhang et al. 2022), where their CNNs model was able to detect effectively EMRI systems with a signal-to-noise ratio (SNR) larger than 50. It has also been shown that, the CNNs’ performance is strongly associated with SNR.

ET is expected to have a detection horizon for BBHs up to redshift $z \approx 100$, but BBHs with total mass 20 - 100 $M_\odot$ will be visible up to $z \approx 20$ (Singh et al. 2021). In this work, we focus on the detection of BBHs merger from the ET with luminosity distance up to 120 Gpc and component masses ranging from 15 to 56 $M_\odot$. Different state-of-the-art CNNs models in image classification were used to find the most optimal performing model. Time-frequency domains of pure noise and injected noise have been used to train, validate and test our models. The structure of this work goes as follow: in Section 2 we describe the method and tools were used for the generation of our synthetic data and the preprocessing pipeline employed for spectrograms preparation. Section 3 discusses The Convolutional Neural Networks, the DL networks used, and training and evaluation processes. Evaluating the best performing model on five hours of ET synthetic data is presented in section 4. Finally, a summary of this work and a future work plans is presented in Section 5.
To generate GWs of a BBH sources, some parameters - of the source to be simulated - have to be provided, most importantly the masses of BBH and its $D_L$. These tow parameters - for all our sources - were taken from Singh et al. (2021), which contains a physical properties (masses and redshift $z$) of three populations of merging binaries (NSs and BHs) simulated specifically for the ET, using the ET-D design (Hild et al. 2011). To obtain $D_L$ from redshift $z$, we used Planck 2015 cosmology (Planck Collaboration et al. 2016) assuming flat LCDM, with Hubble constant, $H_0 = (67.8 +/- 0.9) km/s/Mpc$, a matter density parameter $\Omega_m = 0.308 +/- 0.012$, and a scalar spectral index with $n_s = 0.968 +/- 0.006$.

pyCBC also requires specifying the approximant, which is the model to be used for the simulation. We used IMRPhenomD approximant (Husa et al. 2016) which describes time and frequency domain GWs from non-precessing BBHs. More parameters as in Table 1 was to be provided, such as Positions in the sky, time steps ($t_\Delta$), starting frequency ($f_{\text{low}}$), Inclination angle ($I_\theta$), Polarization ($P_\delta$) and Coalescence ($C_\delta$) phases. It must be noted here that, lower starting frequency will result in longer lasting waveforms that can last up to days on the detector. Since we are interested in detecting the BBHs’ mergers, starting frequency was set to 30 Hz to have shorter waveforms.

### 2.1 Data processing

The output of the pyCBC simulation is a one channel time-domain data ($d(t)$), where $d(t)$ is either a pure gaussian noise ($n$) or has a signal (h) been injected into it:

$$d(t) = \begin{cases} h(t) + n(t), & \text{if data contains a BBH’s signal} \\ n(t), & \text{if data contains noise only.} \end{cases}$$

(1)

Using the parameters in Table 1, 25,000 BBH signals were generated with $D_L$ up to 120 Gpc and five SNR ranges: 4-5, 5-6, 6-7, 7-8 and >8. In ML, it’s important to make sure the training data-set is well balanced, to make sure that classifiers won’t be biased, in our case, towards a certain SNR range. To make sure that each SNR range has the same number of sources, we generated 180,000 BBH’s signals and then divided them -based on their SNR values- into the five mentioned ranges. From each SNR range, 5000 signals were selected.

The data preparation and processing has four main steps as shown in Figure 1: (1) Signal and noise generation, generated signals has an observation time $T$ ranges between 0.29 to 0.93 seconds, and each generated noise is a 5 seconds long. (2) Each signal was injected -in a random position- into the 5 seconds long noise. (3) STFT was used to produce the phase and magnitude spectrograms of each injection (noise plus signal) and a noise only time series. The result of this step is a spectrograms of only noise data (here after only noise spectrograms) and spectrograms of data with injected signals (here after injected spectrograms). (4) Limit the frequency to 300 Hz, this is due to the fact that, our signals are always at the bottom of the spectrograms, lies within the 300 Hz limit. This step saves time and computational power needed for the training since we are removing a significant part of the spectrogram where it’s only noise. Figure 2 displays samples of only noise and injected spectrograms, where we show samples of 5 injected signals with SNR belong to the five ranges. We did not whiten the data and no bandpass filtering was applied. For the training of our CNN models, only the magnitude of the STFT was used and the complex part was ignored.

### 3 CONVOLUTIONAL NEURAL NETWORKS (CNN)

The Convolutional Neural Networks (ConvNet; Fukushima (1980)) is a type of feed-forward neural network deep learning model. A feed-forward model means the output from one layer will be used as an input to the subsequent layer. CNNs are primarily designed to take advantage of the 2D structure of an input image (or other 2D input such as a speech signal) to solve complex tasks such as pattern recognition and object detection (O’Shea & Nash 2015). CNNs have a multi-layer structure, which contains different layers with different rules. These layers namely are, convolutional layers, pooling layers and fully-connected layers, for details please see (Schmidhuber 2014). The main rule of a convolutional layer, is to extract features from the input (an image of a spectrogram in our case) through applying a user-defined number of filters, using a convolution operation (Goodfellow et al. 2016). Each convolutional layer is usually followed by an activation function -to introduce non-linearity in the model- such as Hyperbolic Tangent $\tanh$, sigmoid (Haykin 1998), $\text{softmax}$ (Briddle 1990), or Rectified Linear Activation function ($\text{ReLU}$) (Agarap 2018). A Pooling layer comes after the activation function and its main rule is to apply an scalar to vector transformation on a local regions of an image, to generate a representative value of the pixels in that region. Once the features have been pooled, it will be then flattened and fed to a fully connected layer. The rule of the fully-connected layer is same as Artificial Neural Networks (ANNs) which acts as a features classifier (Agarap 2017). CNNs are essentially made up of neurons that receive inputs. When training an CNNs model, the model will learn the associated features of each class, through minimizing the difference (loss) between output prediction and training true labels. For details please see (O’Shea...
3.1 CNNs models

Our main task is to be able to distinguish whether a spectrogram has an injected signal or not, hence it’s a classification problem. In this work, we made use of the most popular networks in computer vision with the state-of-the-art performance in image classification and localization, to look for the most optimal model with the highest performance regarding the problem in hand. These models are namely VGG, ResNet and DenseNet networks.

3.1.1 VGG network

First introduced in 2014 by (Simonyan & Zisserman 2014) in the ImageNet challenge, where their submission won the first and the second places in the localization and classification tracks respectively. They built 2 models with 16 and 19 convolutional layers with 3 by 3 kernel size. We used both architectures, VGG16 and VGG19, in this work.

3.1.2 ResNet

Presented by He et al. (2015) on the ILSVRC 2015 classification challenge, where ResNet architecture won first place. ResNet uses batch normalization, which adjusts the input layer, and identity connection between the layers to protect the network from the well-known vanishing gradient problem. ResNet has different architectures which follow the same concept but with different number of layers, for example ResNet-18, ResNet-34, ResNet-50, ResNet-101, ResNet-110, ResNet-152 etc, we made use of ResNet-101 (ResNet hereafter).

3.1.3 DenseNet

In DenseNet (Huang et al. 2016), the input does not only come from only the previous layer, but from all the preceding layers, which means all the preceding layers share their feature-maps to all subsequent layers. All layers are concatenated in DenseNet, which gives each layer an access to a collective knowledge from all the previous layers. DenseNet achieved higher accuracy compared to ResNet on ImageNet dataset. In this work, we used DenseNet-121 (DenseNet hereafter), which has 120 Convolutional and 4 AvgPool layers.

3.2 Training

The performance of the ML models depend largely on the quality of the training data not on the algorithms (Zhao et al. 2015), which means different algorithms should provide generally similar trend if trained on a robust and large enough data. Hence we devoted a considerable effort to make sure our synthetic data is well representative in terms of $D_L$, mass distribution and balanced SNR ranges of all our BBH sources. We trained VGG16, VGG19, ResNet and DenseNet models with a batch size of 256, learning rate of 0.0001, 200 epochs using RSM optimizer. These parameters were chosen through running different experiments. The input layer size for all models were adjusted to $42 \times 365 \times 1$ shape size which describe the width, height and number of bands of the input image respectively (1 band denotes gray scale images). As a tradition in training ML models, our data were split into training, validation and testing sets, Table 2 shows the total number of spectrograms used in each.

3.3 Networks Evaluation

To evaluate our trained models, a classification report was produced for each model for each SNR range where the precision, recall and F1-scores (Fawcett 2006) were calculated as shown in Table 3. In terms of accuracy, DenseNet, ResNet and VGG19 showed significantly better performance over VGG16, on sources with higher SNR ranges (6-7, 7-8 and >8). DenseNet was outperformed by ResNet in
Table 3. Evaluation metrics for all models.

| Model | SNR range | class | accuracy | precision | recall | f1-score |
|-------|-----------|-------|----------|-----------|--------|----------|
| VGG16 | 4-5       | injected | 0.575   | 0.913     | 0.212  | 0.344   |
|       |           | noise   | 0.554   | 0.980     | 0.708  |          |
| VGG19 | 4-5       | injected | 0.605   | 0.913     | 0.212  | 0.344   |
|       |           | noise   | 0.554   | 0.980     | 0.708  |          |
| RESNET| 4-5       | injected | 0.620   | 0.900     | 0.273  | 0.419   |
|       |           | noise   | 0.571   | 0.970     | 0.719  |          |
| DENSE | 4-5       | injected | 0.600   | 0.919     | 0.343  | 0.500   |
|       |           | noise   | 0.586   | 0.970     | 0.738  |          |
|       | 5-6       | injected | 0.685   | 0.922     | 0.455  | 0.596   |
|       |           | noise   | 0.630   | 0.929     | 0.751  |          |

SNR range 4-5 with 0.02, and VGG19 with 0.005 better accuracy score respectively. And in SNR range 6-7 by ResNet, with 0.01. In SNR range 7-8, DenseNet has better accuracy score with 0.02 over ResNet, and both has identical performance in SNR >8 of 0.985. Same conclusion will be drawn if we compare the values of F1-score of both, ResNet and DenseNet. Based on this, ResNet model was used for producing confusion matrix to further evaluate its performance. Better visualization of the accuracy for all models is shown in Figure 3 where the detection accuracy is compared to all SNR ranges for each models. The general trend is exactly as expected, where higher SNR sources have higher detection accuracy. It can clearly be seen that all models can detect completely hidden signals, with different accuracy, from sources with very low SNR down to 4.

Confusion matrix (Stehman 1997) illustrates how accurate the model is classifying each class, injected and only noise for each SNR range in our case, through visualizing the true positive, false positive, true negative and false negative values for each class. For producing confusion matrices, we have generated 200 spectrograms (of injected signals and only noise) for each SNR range. Figure 4 displays the confusion matrices for each SNR range. For SNR >8, no only-noise spectrogram was classified as an injection, while only one only-noise in SNR range 7-8 was miss-classified as an injection. 8 injected spectrograms were mis-classified as only-noise and 5 only-noise got classified as injected in SNR range 6-7. For SNR range 5-6, 54 injected spectrograms were classified as only-noise, while only 7 only-noise spectrograms were recognised as injected. The least performance was for SNR 4-5 sources, where 66 injected spectrograms were classified as only-noise, but only three injected spectrograms were classified as only-noise. This indicates a very good performance and means that type II error is unlikely to occur, therefore the model won’t easily recognize noise as an injection.

4 EVALUATION ON ET’S SYNTHETIC DATA

It’s important to evaluate our best performing model on a relatively long time series data, to check for the efficiency and practicality of
a near-real time detection. To do so, we generated a mock data of five hours and 75 seconds of time series data with a signal injected randomly in every five seconds, and one hour and 15 seconds data of only noise. False Negative Rate (FNR) (also known as Type II Error rate) and False Alarm Rate (FAR) (also called false positive ratio or Type I Error Rate) are an effective metrics for measuring the performance of machine learning models applied to classification problems (Burke et al. 1988). FNR is the ratio of the total number of false negative (FN) (here the total number of undetected sources that’s actually been injected) by the total number of FN and true positive (TP) (here the correct classification of injected sources that’s actually injected). FNR shows how often an injected source would be miss-classified as pure noise, and mathematically is calculated as FN/FN+TP. FAR is the ratio of the total number of false positive (FP) (here the incorrect identification of only noise, i.e. classifying only noise as injected) by the total number of FP and true negative (TN) (here the total number only noise). FAR shows how often the model incorrectly classifies an only noise spectrogram as an injected, and FAR mathematically is calculated as: FP/(FP + TN).

Each segment (hour and 15 seconds) in the injected mock data contains 723 signals, which belong to specific SNR range. For the detection and calculation of FNR on this data, we built a pipeline where an sliding window of 5 seconds size will be moved through the data to capture a 5 seconds chunk each time. Then a spectrogram will be produced for each chunk and fed into the model for inference. Figure 5 displays the flow chart of the ET’s mock data inference process. Table 4 shows the report for each hour, where the number of detected sources is compared to the total number of injected signals, and FNR is calculated for each hour. 722 sources with SNR >8 out of 723 total sources were detected, with only 0.001 FNR. Sources with 7-8 SNR range, 713 sources were detected with an FNR of 0.014, while sources with 6-7 SNR range has an FNR of 0.05 with 687 sources detected. Sources with 5-6 and 4-5 SNR range have lower FNR as expected, with 336 and 208 sources were detected respectively. FAR was calculated on the only noise 1 hour and 15 seconds data, where 24 nonexistent sources were detected with 0.033 FAR. This confusion is due to the training on signals with lower SNR.

### Table 4. Detection evaluation on ET’s synthetic data.

| Hours | SNR range | Total Detected | FNR/FAR |
|-------|-----------|----------------|---------|
| 1st   | >8        | 723            | 722     | 0.001 |
| 2nd   | 7-8       | 723            | 713     | 0.014 |
| 3rd   | 6-7       | 723            | 687     | 0.050 |
| 4th   | 5-6       | 723            | 336     | 0.535 |
| 5th   | 4-5       | 723            | 208     | 0.712 |
| 6th   | Noise only| 0              | 24      | 0.033 |

Figure 6 and Figure 7 displays the SNR, M, and D_L associated with the detected and undetected BBH sources respectively. Detected sources range in D_L from 125 - 60 Gpc, SNR from 4 - 168, and chirp mass from 15 - 56 M_☉. Undetected sources have a left skewed distribution for chirp mass and maximum D_L of 117.5 Mpc with SNR range between 4 to 8.8. Most undetected sources have lower SNR and mass with larger D_L. Minimum SNR detected source has an SNR of 4 with D_L of 17.9 Mpc and 23.6 M, while the maximum D_L of a detected source has SNR of 4.3 with a D_L of 60 Gpc and M of 35 M_☉.

### 5 CONCLUSIONS AND FUTURE WORK

VGG16, VGG19, ResNet and DenseNet were trained and tested for the detection of BBHs merger from ET’s synthetic data with five SNR ranges. Without performing data whitening or applying bandpass filtering, ResNet has significantly better performance with an average accuracy of 85%. On testing data, ResNet model was able to detect BBH sources with SNR of 8 or higher with 98.5% accuracy, and with 92.5%, 85%, 60% and 62% accuracy for sources with SNR range of 7-8, 6-7, 5-6 and 4-5 respectively. For 4-5 SNR range. ResNet model was able to detect a BBH’s merger at 60Gpc with 4.3 SNR and a chirp mass of 35 M_☉. It was also shown that, the use of CNN for merger detection is computationally efficient, in our case, scanning a one hour of data only took 4 minutes and 57 seconds. This can be significantly improved, which will focus on spinning BBHs and BNSs sources, injecting glitches and consider the overlapped sources scenario.

### ACKNOWLEDGEMENTS

The authors thank Prof. Piotr Gawron and Dr. Josiah Ensing for their helpful discussions and reviews. This work has been supported by...
the European Union’s Horizon 2020 research and innovation programme under grant agreement No 952480 (DarkWave project), and from the International Research Agenda Programme AstroCeNT (MAB/2018/7) funded by the Foundation for Polish Science (FNP) from the European Regional Development Fund. Additionally it made use of the computational cluster available at Nicolaus Copernicus Astronomical Center.

DATA AVAILABILITY

Data used in this work -including time-series and spectrograms- can be shared on a reasonable request, please contact the author to request the data.

REFERENCES

Abbott B. P., et al., 2009, Reports on Progress in Physics, 72, 076901
Abbott B. P., et al., 2016, Phys. Rev. Lett., 116, 061102
Abernathy M., et al., 2011, Universität śląski, pp 343–354
Acernese F., et al., 2007, AIP Conference Proceedings, 924, 187
Agarap A. F., 2017, arXiv e-prints, p. arXiv:1712.03541
Agarap A. F., 2018, arXiv e-prints, p. arXiv:1803.08375
Alhassan W., Taylor A. R., Vaccari M., 2018, Monthly Notices of the Royal Astronomical Society
Álvares J. D., Font J. A., Freitas F. F., Freitas O. G., Morais A. P., Nunes S., Onofre A., Torres-Forné A., 2021, Classical and Quantum Gravity, 38, 155010
Baltus G., Janquart J., Lopez M., Reza A., Caudill S., Cudell J.-R., 2021, Phys. Rev. D, 103, 102003
Baltus G., Janquart J., Lopez M., Narola H., Cudell J.-R., 2022, arXiv e-prints, p. arXiv:2205.04750
Biswas R., et al., 2013, Phys. Rev. D, 88, 062003
Blackman R. B., Tukey J. W., 1958, The Bell System Technical Journal, 37, 185
Bonfield J. S., 1990, in Souilé F. F., Hérault J., eds, Neurocomputing. Springer Berlin Heidelberg, Berlin, Heidelberg, pp 227–236
Bulik T., Belczyski K., 2003, The Astrophysical Journal, 589, L37
Burke D. S., Brundage J. F., Redfield R. R., Damato J. J., Schable C. A., Putman P., Visintine R., Kim H. I., 1988, New England Journal of Medicine, 319, 961
Colgan R. E., Corley K. R., Lau Y., Bartos I., Wright J. N., Márka Z., Márka S., 2020, Phys. Rev. D, 101, 102003
Collister A. A., Lahav O., 2004, Publications of the Astronomical Society of the Pacific, 116, 345
D’Isanto A., Polsterer K. L., 2018, Astronomy & Astrophysics, 609, A111
Evans M., et al., 2021, arXiv e-prints, p. arXiv:2109.09882
Fawcett T., 2006, Pattern Recognition Letters, 27, 861
Fukushima K., 1980, Biological Cybernetics, 36, 193
Goodfellow I., Bengio Y., Courville A., 2016, Deep Learning. MIT Press
Haykin S., 1998, Neural Networks: A Comprehensive Foundation, 2nd edn. Prentice Hall PTR, USA
He K., Zhang X., Ren S., Sun J., 2015, Deep Residual Learning for Image Recognition, doi:10.48550/ARXIV.1512.03385, https://arxiv.org/abs/1512.03385
Hild S., et al., 2011, Class. Quant. Grav., 28, 094013
Huang G., Liu Z., van der Maaten L., Weinberger K. Q., 2016, Densely Connected Convolutional Networks, doi:10.48550/ARXIV.1608.06993, https://arxiv.org/abs/1608.06993
Husa S., Khan S., Hannam M., Pürrer M., Ohme F., Forteza X. J., Bohé A., 2016, Phys. Rev. D, 93, 044006
Iaconelli F., Mancarella M., Foffa S., Maggiore M., 2022, arXiv e-prints, p. arXiv:2207.02771
Jiang M.-Q., Yang N., Li J., 2022, Frontiers of Physics, 17, 54501
Kalogera V., et al., 2021, arXiv e-prints, p. arXiv:2111.06990
Kim E. J., Brunner R. J., 2017, Monthly Notices of the Royal Astronomical Society, 464, 4463
Klimenko S., et al., 2016, Phys. Rev. D, 93, 042004
Lukic V., Brüggen M., 2016, Proceedings of the International Astronomical Union, 12, 217
Maggiore M., et al., 2020, J. Cosmology Astropart. Phys., 2020, 050
Moreno E. A., Borzyszkowski B., Pierini M., Vlimant J.-R., Spiropulu M., 2022, Machine Learning: Science and Technology, 3, 025001
Naim A., Ratnatunga K. U., Griffiths R. E., 1997, The Astrophysical Journal, 476, 510
Nitz A., et al., 2022, gwastro/pycbc: Release v2.0.1 of PyCBC, doi:10.5281/zenodo.5825666, https://doi.org/10.5281/zenodo.5825666
O’Shea K., Nash R., 2015, arXiv e-prints, p. arXiv:1511.08458
Peters P. C., Mathews J., 1963, Phys. Rev., 131, 435
Planck Collaboration et al., 2016, A&A, 594, A13
Punturo M., Abernathy M., Acernese F., other 2010, Classical and Quantum Gravity, 27, 194002
Schmidhuber J., 2014, arXiv e-prints, p. arXiv:1404.7828
Schmidt S., et al., 2020, arXiv: General Relativity and Quantum Cosmology
Selim I. M., Abd El Aziz M., 2017, Experimental Astronomy, 43, 131
Selim I., E. A., M.El B., 2016, International Journal of Computer Applications, 137, 4
Shaddock D. A., 2008, Classical and Quantum Gravity, 25, 114012
Simonyan K., Zisserman A., 2014, Very Deep Convolutional Networks for Large-Scale Image Recognition, doi:10.48550/ARXIV.1409.1556, https://arxiv.org/abs/1409.1556
Singh N., Balik T., Belczynski K., Askar A., 2021, arXiv e-prints, p. arXiv:2112.04058
Stehman S. V., 1997, Remote Sensing of Environment, 62, 77
Vajente G., 2022, Phys. Rev. D, 105, 102005
Willike B., 2007, Classical and Quantum Gravity, 24, S389
Yin Q., Shen L., Lu M., Wang X., Liu Z., 2013, Journal of Systems Engineering and Electronics, 24, 26
Zhang Y., Li L., Zhao Y., 2009, Monthly Notices of the Royal Astronomical Society, 392, 233
Zhang X.-T., Messenger C., Korsakova N., Chan M. L., Hu Y.-M., Zhang J.-d., 2022, Phys. Rev. D, 105, 123027
Zhao T., Jayaram V., Roy A., Marfurt K. J., 2015, Interpretation, 3, SAE29

This paper has been typeset from a \LaTeX file prepared by the author.